

Algoritmo heurístico de otimização sequencial para o problema de planejamento de lavra em minas a céu aberto

**Douglas Batista Mazzinghy^{1,2}, Fábio Oliveira de Paula², Luciano Perdigão Cota^{2,3},
Marcone Jamilson Freitas Souza²**

¹Departamento de Engenharia de Minas

Universidade Federal de Minas Gerais, CEP: 31270-901, Belo Horizonte, MG, Brasil

²Programa de Pós-Graduação em Ciência da Computação

Universidade Federal de Ouro Preto, CEP: 35402-136, Ouro Preto, MG, Brasil

³Instituto Tecnológico Vale

Campus Universitário, Morro do Cruzeiro, CEP: 35402-206, Ouro Preto, MG, Brasil

`dmazzinghy@demin.ufmg.br`, `fabio.olp@aluno.ufop.edu.br`,

`luciano.p.cota@itv.org`, `marcone@ufop.edu.br`

RESUMO

O problema de sequenciamento de blocos em minas a céu aberto é considerado NP-difícil devido às suas propriedades combinatórias. A solução dele através de programação matemática é inviável devido ao elevado tempo computacional requerido. Este trabalho apresenta um algoritmo heurístico de otimização sequencial, baseado em programação matemática, para resolver o problema período por período. Nesse algoritmo, cada subproblema referente a um período é resolvido na otimalidade por um resolvidor de programação matemática. O algoritmo foi testado em dois modelos de blocos que representam minas reais e disponibilizados de forma pública pela biblioteca MineLib. Os resultados obtidos pelo algoritmo heurístico de otimização sequencial são próximos aos melhores resultados apresentados pelo MineLib. Nossas contribuições incluem o compartilhamento do código desenvolvido, possibilitando que outros pesquisadores tenham um ponto de partida para desenvolver soluções ainda melhores que as apresentadas aqui.

PALAVRAS CHAVE. Mineração, Sequenciamento de lavra, Programação matemática, Pesquisa Operacional, Heurísticas.

Tópicos: POI – PO na Indústria, OA – Outras Aplicações em PO

ABSTRACT

The block sequencing problem in open-pit mines is considered NP-hard due to its combinatorial properties. Its solution through mathematical programming is infeasible due to the high computational time required. This work presents a heuristic sequential optimization algorithm, based on mathematical programming, to solve the problem period by period. In this algorithm, a mathematical programming-based solver determines the optimal solution for each period. The algorithm was tested on two block models that represent real mines and are publicly available through the MineLib library. The results obtained by the heuristic sequential optimization algorithm are close to the best results presented by MineLib. Our contributions include sharing the developed code, allowing other researchers to have a starting point to develop solutions even better than those presented here.

KEYWORDS. Mining, Mine scheduling, Mathematical programming, Operations research, Heuristics.

Topics: ORI - OR in Industry, OA – Others OR Applications.

1. Introdução

A mineração tem um papel fundamental na sociedade, fornecendo materiais essenciais para a fabricação de diversos bens de consumo. O planejamento de lavra em minas a céu aberto é um processo crítico para a viabilidade econômica das operações de mineração. Um componente fundamental desse planejamento é o sequenciamento de blocos, que define quais blocos serão extraídos, em qual momento e para quais destinos os blocos serão enviados (usina de processamento, pilha de estéril ou pilha de estoque). A otimização busca a maximização do valor do negócio respeitando os aspectos de segurança e sócio-ambientais. O problema de sequenciamento pode ser representado por um problema clássico de aplicação da Pesquisa Operacional através de Programação Linear Inteira Mista (PLIM), o Problema da Mochila com Restrições de Precedência Multiperíodo (*Multi-Period Precedence-Constrained Knapsack Problem - MPPCKP*). O problema aplicado ao sequenciamento de blocos consiste na seleção de blocos em vários períodos de tempo, onde alguns blocos precisam ser selecionados antes de outros (restrições de precedência) e as restrições de capacidade devem ser respeitadas em cada período.

Na década de 1960, dois algoritmos foram propostos para resolver o problema de planejamento de lavra, algoritmos de Lerchs and Grossmann [1965] e de Johnson [1968]. Recentemente, o algoritmo desenvolvido por Bienstock and Zuckerberg [2010] parece ser o mais robusto para a solução do problema de sequenciamento de blocos. As seções seguintes descrevem de forma simplificada cada um dos algoritmos citados.

1.1. Algoritmo de Lerchs and Grossmann [1965]

Lerchs and Grossmann [1965] desenvolveram um algoritmo para determinar a cava final, maximizando a receita. Esse algoritmo não permite incluir as restrições de capacidade. Para contornar essa limitação, o valor econômico dos blocos é multiplicado por um fator de receita (*Revenue Factor - RF*) para gerar várias cavas aninhadas. Ao diminuir o valor dos blocos, as cavas obtidas são menores e essas cavas aninhadas servem como guia para o sequenciamento dos blocos. No entanto, embora o algoritmo de LG seja exato para obter as cavas, não há garantia de que o sequenciamento derivado das cavas aninhadas seja ótimo. Como não é possível incluir as restrições de capacidades, essa variável não tem controle e cada cava gerada possui um valor de capacidade. Atualmente, o algoritmo *pseudoflow* [Hochbaum, 2008] é o mais utilizado pela indústria mineral, pois obtém a mesma solução do algoritmo de LG em menor tempo computacional (Fluxo Máximo e Corte Mínimo). Recentemente, Deutsch et al. [2022] disponibilizaram de forma pública (*open-source*) uma versão modificada e eficiente do algoritmo *pseudoflow*.

1.2. Algoritmo de Johnson [1968]

Johnson [1968] desenvolveu um algoritmo para o sequenciamento de blocos utilizando PLIM. Contudo, devido às limitações de hardware e software da época, o problema somente era resolvido para modelos de blocos de pequeno porte. Com os avanços de hardware e software, métodos exatos baseados em PLIM foram desenvolvidos [Caccetta and Hill, 2003; Caccetta, 2007; Chicoisne et al., 2012], mas ainda são limitados pelo tamanho do problema devido à complexidade computacional. O problema de sequenciamento de blocos é classificado como NP-difícil, o que significa que o tempo necessário para encontrar uma solução ótima cresce exponencialmente com o tamanho do problema.

1.3. Algoritmo de Bienstock and Zuckerberg [2010]

Bienstock and Zuckerberg [2010] desenvolveram um algoritmo exato que divide o problema em partes menores por meio de decomposição, resolvendo essas partes separadamente usando PLIM e depois combinando as soluções para obter o resultado ótimo do problema original. O algoritmo BZ utiliza métodos avançados, como geração de colunas usando o algoritmo *pseudoflow* e decomposição lagrangiana, para acelerar a resolução, sendo especialmente eficiente

em problemas de grande escala. As melhores soluções apresentadas pelo MineLib foram obtidas com uma versão modificada do algoritmo BZ [Espinoza et al., 2013]. Para melhor entendimento sobre o algoritmo BZ sugere-se a leitura dos seguintes trabalhos: Muñoz et al. [2018] e Muñoz [2018].

1.4. Algoritmos Exatos combinados com Algoritmos Heurísticos

Recentemente, algoritmos matheurísticos, assim chamados os algoritmos que combinam métodos exatos (programação matemática) com métodos heurísticos, têm demonstrado resultados promissores. Lamghari et al. [2015] desenvolveram um algoritmo que utiliza programação linear para obtenção de uma solução inicial e o procedimento heurístico de busca local *variable neighborhood descent* para melhorar a solução. Jélvez et al. [2016] propuseram uma heurística de agregação/desagregação que resolve o problema em janelas de tempo consecutivas através de programação matemática. Samavati et al. [2017] propuseram um algoritmo eficiente para resolver o problema de sequenciamento que usa os valores da relaxação linear como pesos para a heurística *TopoSort*. Brandão and Mazzinghy [2024] desenvolveram um algoritmo para otimização sequencial usando programação matemática e incluindo a variável geometalúrgica de horas de processamento.

1.5. Objetivos e Contribuições do Trabalho

Este trabalho tem como objetivo desenvolver um algoritmo heurístico de otimização sequencial para solução do problema de sequenciamento de blocos em minas a céu aberto. O algoritmo de Bienstock and Zuckerberg [2010] é complexo em termos de formulação matemática e implementação computacional e não está disponível de forma pública. A principal contribuição deste trabalho é o compartilhamento do código desenvolvido para que outros pesquisadores tenham um ponto de partida para desenvolver novos métodos de solução. O algoritmo desenvolvido possui formulação matemática simples e é de fácil implementação computacional.

2. Formulação Matemática

Nesta seção, apresenta-se a formulação matemática do problema de sequenciamento de blocos em minas a céu aberto. Esse problema consiste em decidir a ordem e o período em que cada bloco será extraído, respeitando-se as restrições de precedência e capacidade, e buscando-se maximizar o valor econômico. A formulação apresentada é baseada no *Constrained Pit Limit Problem* (CPIT) apresentado por Espinoza et al. [2013]. A seguir, são apresentados os conjuntos, índices, parâmetros de entrada e variáveis de decisão, para solução do problema:

Conjuntos:

B : Conjunto de blocos, $B = \{0, 1, 2, \dots, |B|\}$;

T : Conjunto de períodos de tempo, $T = \{0, 1, 2, \dots, |T|\}$;

P_b : Conjunto de blocos b' que são precedentes do bloco b (blocos que precisam ser extraídos para acessar o bloco b).

Índices:

b : Índice para um elemento do conjunto B ;

b' : Índice para um elemento do conjunto P_b ;

t : Índice para um elemento do conjunto T .

Parâmetros:

v_b : valor econômico do bloco b ;

m_b : massa do bloco b ;

d_b : destino do bloco b (1, se o destino for a usina de processamento e 0, se for a pilha de estéril);

r : taxa de desconto.

Variáveis de decisão:

x_{bt} : Variável binária igual a 1 se e somente se $b \in B$ é extraído no período $t \in T$ e 0 para outros casos;

$x_{b't}$: Variável binária igual a 1 se os blocos $b' \in P_b$ forem precedentes do bloco b devido às restrições de ângulo e talude e 0 para outros casos;

A formulação matemática do problema é apresentada a seguir:

$$\text{maximizar} \quad \sum_{b \in B} \sum_{t \in T} \frac{v_{bt} \cdot x_{bt}}{(1+r)^t} \quad (1)$$

$$\text{sujeito a} \quad x_{bt} \leq \sum_{t \in T} x_{b't} \quad \forall b \in B, b' \in P_b, t \in T \quad (2)$$

$$\sum_{t \in T} x_{bt} \leq 1 \quad \forall t \in T \quad (3)$$

$$\sum_{b \in B} d_b \cdot m_b \cdot x_{bt} \leq C_p \quad \forall t \in T \quad (4)$$

$$x_{bt} \in \{0, 1\} \quad \forall b \in B, \forall t \in T \quad (5)$$

A função objetivo (1) maximiza o valor presente líquido (VPL) da mina. O conjunto de restrições (2) representa as relações de precedência entre os blocos, garantindo a estabilidade geotécnica da cava. As restrições (3) asseguram que cada bloco seja extraído apenas em um período. As restrições (4) limitam a quantidade de material enviado para a usina de processamento em cada período. Finalmente, as restrições (5) definem as variáveis de decisão como binárias. Esta formulação pode ser estendida para incluir restrições adicionais, como capacidades mínimas de processamento, múltiplos destinos, restrições de teores médios (*blending*), inclusão de variáveis geometalúrgicas, geometrias de mina (espaçamentos mínimos para equipamentos de carga e transporte), entre outras. No entanto, para o propósito deste trabalho, focou-se na formulação básica.

3. Algoritmo Proposto

O algoritmo heurístico de otimização sequencial proposto neste trabalho resolve o problema de sequenciamento de blocos de forma sequencial, período por período. A ideia central é resolver o problema considerando apenas um período t por vez e em seguida remover do banco de dados os blocos desse período. Em seguida, resolver o problema para o período $t + 1$ e também remover do banco de dados os blocos desse período. Esse procedimento é feito até que não seja mais viável a extração dos blocos, ou seja, até que o somatório dos valores econômicos do período que está sendo otimizado seja negativo. Para os blocos selecionados, a massa é aumentada artificialmente para um valor extremamente alto, o que efetivamente remove esses blocos das considerações para períodos futuros. Os blocos selecionados também são removidos da matriz de precedência, o que libera seus sucessores para extração em períodos subsequentes. Essa abordagem sequencial combinada com a manipulação dinâmica dos parâmetros permite que o algoritmo encontre soluções

de alta qualidade para o problema de sequenciamento de blocos sem a necessidade de resolver o problema completo de uma vez, o que seria computacionalmente inviável para instâncias de grande porte. Para cada período $t \in T$, resolve-se o seguinte problema de otimização:

$$\text{maximizar} \quad \sum_{b \in B^t} \frac{v_b \cdot x_b}{(1+r)^t} \quad (6)$$

$$\text{sujeito a} \quad x_b \leq x_{b'} \quad \forall b \in B, b' \in P_b \quad (7)$$

$$\sum_{b \in B} d_b \cdot m_b \cdot x_b \leq C_p \quad \forall t \in T \quad (8)$$

$$x_b \in \{0, 1\} \quad \forall b \in B, \quad (9)$$

no qual B^t é o conjunto de blocos remanescentes após a extração dos blocos b no período t . Dessa forma, a cada etapa de otimização, o conjunto B^t é reduzido, refletindo a remoção dos blocos já extraídos. O mesmo acontece com os precedentes, pois o conjunto de blocos precedentes P_b é reduzido a cada período t . O Algoritmo 1 apresenta o pseudocódigo do algoritmo heurístico de otimização sequencial proposto baseado em programação matemática.

Algoritmo 1: Otimização Heurística Sequencial

Entrada: Modelo de blocos, lista de precedências, *ultimo_periodo*,
capacidade_processo, *taxa_desconto*

```

1 Inicializar dataframe com o modelo de blocos
2 Inicializar dataframe com lista de precedências
3 para  $t = 0$  até  $\text{ultimo\_periodo} - 1$  faça
4     Otimizar modelo (Programação Matemática)
5     se solução ótima encontrada e valor objetivo > 0 então
6         Calcular valor descontado
7          $\text{valor\_descontado} \leftarrow \frac{\text{funcao\_objetivo}}{(1+r)^t}$ 
8         Adicionar valor_descontado à lista valor_periodo
9         para cada variável  $x_b = 1$  faça
10            Marcar bloco  $b$  como extraído no período  $t$ 
11            Aumentar  $m_b$  artificialmente para evitar extração em  $t + 1$ 
12        Atualizar lista de precedentes para remover blocos extraídos
13    senão
14        Parar o loop (sequenciamento até o período anterior)
15    pare
16    Limpar modelo para o próximo período
17 Calcular Valor Presente Líquido (VPL):
18     $\text{vpl} \leftarrow \sum \text{valor\_periodo}$ 
Retorne: vpl

```

4. Experimentos computacionais

O código do algoritmo proposto foi escrito em Python 3.11.11, utilizando-se como interface o Jupyter Notebook 7.4.1. Os subproblemas de programação linear inteira mista foram

modelados com o pacote Python-MIP 1.15 [Santos et al., 2020]. Esse pacote foi desenvolvido por professores e alunos do Departamento de Computação (DECOM) da Universidade Federal de Ouro Preto (UFOP). Os subproblemas foram resolvidos com o resolvidor CBC [Forrest et al., 2024], considerando-se um gap de otimização máximo de 5% para reduzir o tempo computacional e ainda manter o objetivo de encontrar soluções de qualidade. Todos os pacotes utilizados são *open-source*.

O algoritmo proposto foi testado em um laptop de modesta configuração, com processador Intel® Core™ i7-10510U CPU@1,80 GHz, 16 GB de RAM e sistema operacional Windows 10 (64 bits). Para avaliar o desempenho do algoritmo, foram realizados experimentos computacionais utilizando dois modelos de blocos disponíveis publicamente na biblioteca MineLib (<https://mansci-web.uai.cl/minelib/>). O modelo de blocos ‘kd’ representa um depósito de cobre do Arizona nos EUA e o modelo de blocos ‘mclaughlin limit’ representa uma antiga mina de ouro localizada na Califórnia, também EUA. A Tabela 1 apresenta as principais características dos modelos de blocos utilizados.

Tabela 1: Principais informações dos modelos de blocos kd e mclaughlin.limit.

Modelos de Blocos	kd	mclaughlin.limit
Quantidade Blocos	14.153	112.687
Quantidade Precedências	219.778	3.035.483
Dimensões dos Blocos	20 × 20 × 15 metros	25 × 25 × 20 pés
Cap. de Processamento	10 milhões	3,3 milhões
Taxa de Desconto	15%	15%
Ângulo de Talude	45°	45°
Máximo de Períodos	10	10

A Figura 1 mostra as estruturas dos dados de entrada, modelo de blocos e lista de precedências para o dataset ‘kd’.

Modelo de Blocos										Lista de Precedências											
id	x	y	z	tonn	blockvalue	destination	cu	process	profit	id	n	prec1	prec2	prec3	prec4	prec5	prec6	prec7	prec8	...	
0	11	0	18	16380	-12285.0000	0	0.000	0.0000		0	0	-1	-1	-1	-1	-1	-1	-1	-1	...	
1	12	0	18	16380	-12285.0000	0	0.000	0.0000		1	0	-1	-1	-1	-1	-1	-1	-1	-1	...	
2	13	0	18	16380	-12285.0000	0	0.000	0.0000		2	0	-1	-1	-1	-1	-1	-1	-1	-1	...	
3	13	1	18	16380	-12285.0000	0	0.000	0.0000		3	0	-1	-1	-1	-1	-1	-1	-1	-1	...	
4	14	0	18	16380	-12285.0000	0	0.000	0.0000		4	0	-1	-1	-1	-1	-1	-1	-1	-1	...	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
14148	60	22	0	16260	136481.0096	1	0.924	148676.0096		14148	25	14106	13998	13980	13990	13988	13831	13782	13999	...	
14149	60	23	0	16260	132053.7237	1	0.908	144248.7237		14149	25	14107	13999	13981	13991	13989	13832	13783	14000	...	
14150	60	24	0	16260	128043.1476	1	0.888	140238.1476		14150	25	14108	14000	13982	13992	13990	13833	13784	14001	...	
14151	60	25	0	16260	117441.3854	1	0.836	129636.3854		14151	25	14109	14001	13983	13993	13991	13834	13785	14002	...	
14152	61	23	0	16260	132781.5934	1	0.915	144976.5934		14152	25	14112	14007	13990	14000	13998	13843	13795	14008	...	

Figura 1: Estrutura dos arquivos de entrada.

A segunda coluna da lista de precedentes ‘n’ indica a quantidade de blocos precedentes do bloco indicando na primeira coluna ‘id’. As demais colunas indicam os ‘ids’ dos blocos precedentes.

5. Resultados

Os resultados de Valor Presente Líquido (VPL) obtidos pelo algoritmo proposto e o tempo de execução são apresentados na Tabela 2, juntamente com os melhores resultados disponibiliza-

dos pelo MineLib. O tempo de execução do algoritmo Bienstock e Zuckerberg (BZ) modificado utilizado no MineLib não é informado.

Tabela 2: Resultados de VPL para os modelos de blocos kd e mclaughlin_limit e os tempos de execução.

<i>Dataset</i>	kd	mclaughlin_limit
VPL Algoritmo BZ - Minelib	409.498.555	1.078.979.501
VPL Algoritmo Heurístico Sequencial	408.477.039	1.078.556.812
VPL Diferença	0,25%	0,04%
Tempo de Atualização de Dados	~ 4 minutos	~ 3 horas
Tempo de Otimização	~ 18 minutos	~ 3 horas
Tempo Total	~ 22 minutos	~ 6 horas

O algoritmo proposto demonstrou excelente desempenho para os dois modelos de blocos testados. Para o modelo de blocos 'kd', a diferença de VPL foi de 0,25% e para o modelo de blocos 'mclaughlin limit' a diferença de VPL foi de apenas 0,04%.

O tempo de execução total foi de aproximadamente 22 minutos para o modelo de blocos 'kd', que possui 14.153 blocos e foi de aproximadamente seis horas para o modelo de blocos 'mclaughlin limit', que possui 112.687 blocos. Cerca de metade do tempo de execução foi utilizado para atualização dos dados (arquivos) no caso do modelo de blocos 'mclaughlin limit'. Portanto, existe espaço para reduzir o tempo computacional de leitura e atualização dos dados através do uso de pacotes mais eficientes.

As Figuras 2 e 3 mostram as capacidades de processo (usina) e de mina por período, além do teor médio, para cada modelo de blocos. A capacidade de mina (total de massa movimentada) não é uma restrição nos modelos de blocos utilizados. As restrições de capacidade de processos foram respeitadas nos dois modelos de blocos.

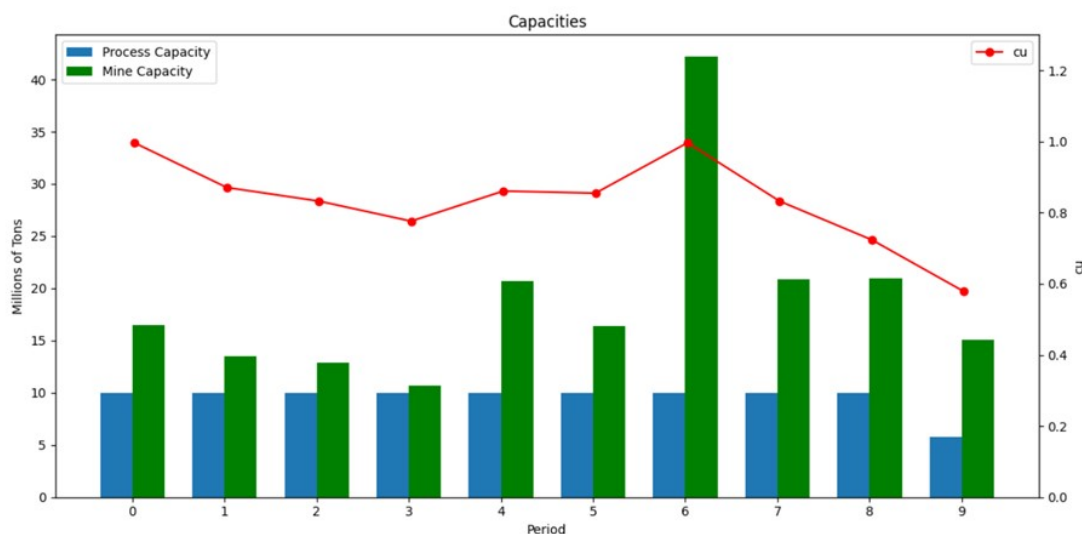


Figura 2: Capacidades e teor médio para o modelo de blocos kd.

Os modelos de blocos do MineLib consideram que blocos com valores positivos são classificados como “minério” e blocos com valores negativos ou igual a zero são classificados como “estéril”. Portanto, existe variação dos teores ao longo dos períodos, como mostrado nos gráficos de capacidades e teores. No modelo de blocos 'kd' os teores médios de cobre variaram menos ao

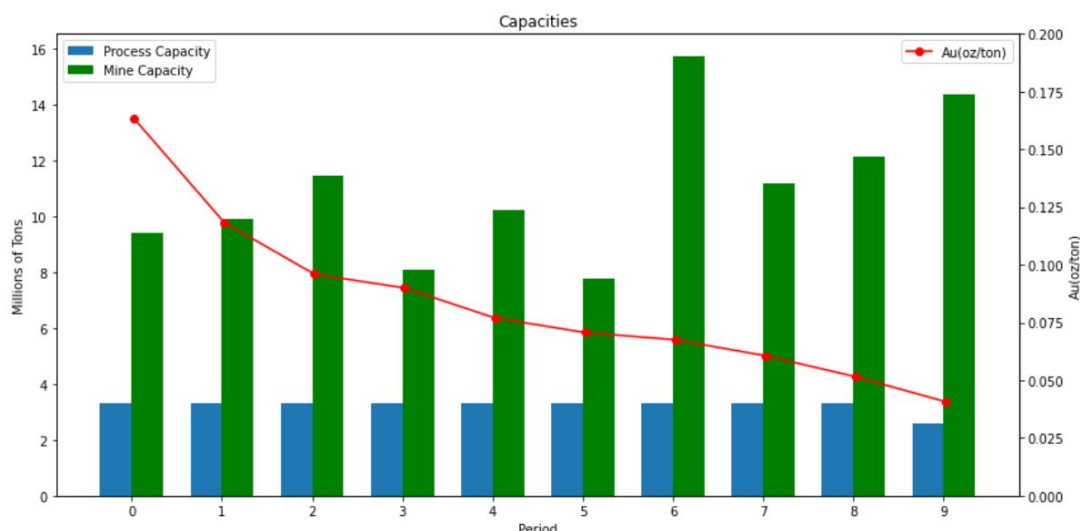


Figura 3: Capacidades e teor médio para o modelo de blocos mclaughlin limit.

longo dos períodos. Já no modelo de blocos 'mclaughlin limit' a variação do teor médio de ouro foi considerável. O algoritmo heurístico de otimização sequencial priorizou a extração dos blocos com maior conteúdo de ouro e, conseqüentemente, maior valor econômico.

A Figura 4 mostra as cavas de alguns períodos obtidas com o algoritmo heurístico de otimização sequencial. As soluções disponibilizadas pelo MineLib não consideram as geometrias de mina (espaçamento mínimo necessário para operação dos equipamentos de carga e transporte). O algoritmo desenvolvido também não considera as geometrias de mina. Portanto, existe espaço para obter soluções mais operacionais considerando as variáveis de geometrias de mina. Percebe-se blocos espalhados e pouca continuidade entre os períodos, o que dificulta a operação dos equipamentos de carga e transporte. Para incluir as restrições de geometria de mina no algoritmo será necessário aglomerar blocos (*clusters*) com o objetivo de formar bancos de lavra. Os equipamentos de carga e transporte necessitam de espaço para operação, tanto nas bancadas como no fundo da cava. A complexidade do problema será elevada significativamente com a inclusão dessas restrições geométricas.

A inclusão das variáveis geometalúrgicas também irá elavar a complexidade do problema, visto que algumas variáveis não são aditivas, como as recuperações de processo, por exemplo. Dessa forma, o problema pode se tornar não-linear. Uma possível alternativa seria aglomerar blocos e estimar a recuperação do conjunto de blocos em função do teor médio do mesmo, através de uma curva de teor-recuperação. Esse tipo de modelo geometalúrgico é obtido através de ensaios em escala de laboratório com várias amostras do depósito mineral.

A estrutura do algoritmo proposto permite sua aplicação a problemas com diferentes tipos de restrições, como capacidades mínimas de processamento, múltiplos destinos, restrições de teores médios, variáveis geometalúrgicas, geometrias de mina, entre outras.

5.1. Complexidade Computacional

O problema de sequenciamento de blocos pode ser abordado de duas formas principais: a abordagem global e a abordagem sequencial.

Abordagem Global

Na abordagem global, as decisões de extração dos blocos para todos os períodos são tomadas simultaneamente em um único modelo de otimização. Seja B o conjunto total de blocos e T o conjunto de períodos, o número total de variáveis do problema é dado pela Equação (10).

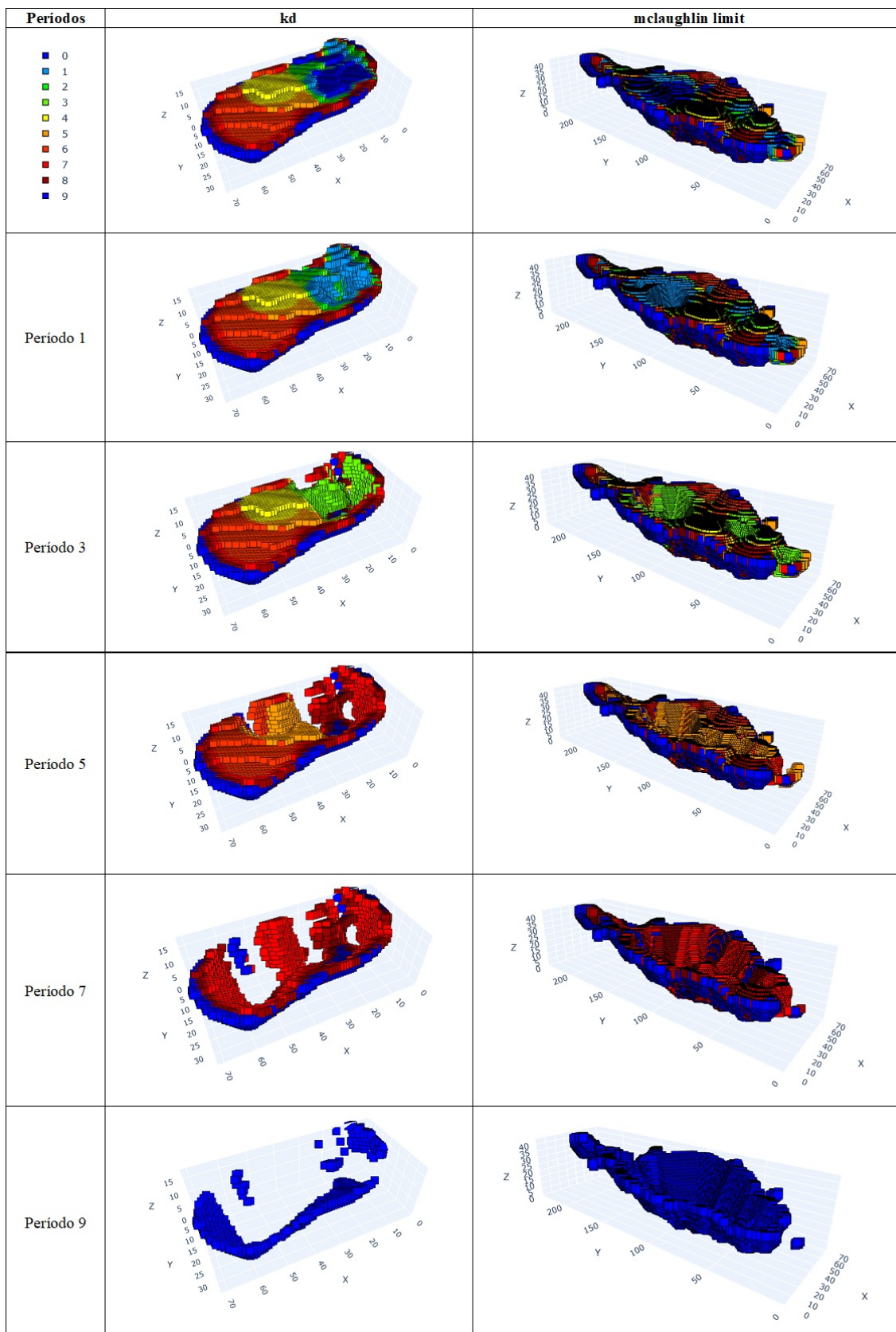


Figura 4: Sequenciamento obtido pelo algoritmo proposto.

$$N_{\text{global}} = |B| \times |T| \quad (10)$$

A complexidade computacional para resolver o problema global cresce exponencialmente com o número total de variáveis, podendo ser expressa pela Equação (11).

$$\text{Tempo}_{\text{global}} = O\left(2^{|B| \times |T|}\right) \quad (11)$$

Abordagem Sequencial

Na abordagem sequencial, o problema é dividido em subproblemas para cada período $t \in T$, considerando apenas os blocos disponíveis naquele período, denotados por $B^t \subseteq B$. O conjunto B^t representa os blocos remanescentes após a extração dos blocos nos períodos anteriores, ou seja, a cada etapa de otimização, o conjunto B^t é reduzido conforme os blocos extraídos em períodos anteriores são removidos.

A complexidade total é a soma dos tempos para resolver cada subproblema, como apresentado na Equação (12).

$$\text{Tempo}_{\text{seq}} = \sum_{t \in T} O\left(2^{|B^t|}\right) \quad (12)$$

Para simplificação, assumimos que o tamanho do conjunto B^t é aproximadamente constante e próximo de $|B|$ para todos os períodos, resultando na Equação (13).

$$\text{Tempo}_{\text{seq}} \approx |T| \times O\left(2^{|B|}\right) \quad (13)$$

Comparação entre as abordagens

A razão entre os tempos de execução das abordagens global e sequencial é aproximada pela Equação (14).

$$\frac{\text{Tempo}_{\text{global}}}{\text{Tempo}_{\text{seq}}} \approx \frac{2^{|B| \times |T|}}{|T| \times 2^{|B|}} = \frac{2^{|B|(|T|-1)}}{|T|} \quad (14)$$

Como $2^{|B|(|T|-1)}$ cresce muito mais rápido que $|T|$, a abordagem global é computacionalmente muito mais custosa do que a sequencial para conjuntos maiores de blocos e períodos. A Tabela 3 compara a complexidade computacional das abordagens global e sequencial.

Tabela 3: Comparação da complexidade computacional das abordagens global e sequencial.

Aspecto	Abordagem Global	Abordagem Sequencial
Número de variáveis	$ B \times T $	$ B^t \leq B $ por período $t \in T$
Complexidade temporal	$O\left(2^{ B \times T }\right)$	$\sum_{t \in T} O\left(2^{ B^t }\right) \approx T \times O\left(2^{ B }\right)$
Escalabilidade	Crescimento exponencial com $ B \times T $	Crescimento linear em $ T $ e exponencial em $ B $
Viabilidade prática	Limitada para valores maiores de $ B $ e $ T $	Mais viável para valores maiores de $ T $

6. Conclusões e trabalhos futuros

Neste trabalho foi desenvolvido um algoritmo heurístico de otimização sequencial, baseado em programação matemática, para resolver de forma eficiente o problema de planejamento de lavra em minas a céu aberto. O algoritmo proposto resolve o problema de forma sequencial, período por período, combinando métodos exatos (programação matemática) com métodos heurísticos. Os experimentos computacionais realizados com os modelos de blocos que representam minas reais,

‘kd’ e ‘mclaughlin’, da biblioteca pública MineLib demonstraram a eficácia do algoritmo proposto, que foi capaz de obter soluções muito próximas das melhores soluções encontradas. As principais vantagens do algoritmo desenvolvido são a simplicidade da formulação matemática e a simplicidade da implementação computacional. Como contribuições deste trabalho, está sendo compartilhado o código desenvolvido, possibilitando que outros pesquisadores utilizem e aprimorem as soluções propostas.

Para trabalhos futuros, sugere-se a extensão do algoritmo proposto para incorporar restrições adicionais, como restrições de teores médios (*blendagem*), variáveis geometalúrgicas e geometrias de mina (espaçamento mínimo necessário para operação dos equipamentos de carga e transporte). Além disso, a hibridização com outras abordagens meta-heurísticas pode levar a resultados ainda melhores. A paralelização do algoritmo pode ser uma estratégia interessante para reduzir o tempo computacional. A conversão do código desenvolvido em uma linguagem de programação mais rápida, como C++ ou Julia, pode reduzir o tempo computacional necessário para tratar datasets de grande porte com milhões de blocos.

Material suplementar

O algoritmo desenvolvido está disponibilizado publicamente no GitHub <https://github.com/douglasmazzinghy/2025-SBPO>.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001, CNPq (processos 102368/2024-4, 311074/2023-5 e 302629/2023-8), FAPEMIG (processo CEX-PPM-676-17), Instituto Tecnológico Vale (ITV), Universidade Federal de Minas Gerais (UFMG) e Universidade Federal de Ouro Preto (UFOP).

Referências

- Bienstock, D. and Zuckerberg, M. (2010). Solving lp relaxations of large-scale precedence constrained problems. In Eisenbrand, S. F., F., editor, *Integer Programming and Combinatorial Optimization, IPCO 2010. Lecture Notes in Computer Science, vol 6080*, p. 1–14. Springer, Berlin, Heidelberg. URL https://doi.org/10.1007/978-3-642-13036-6_1.
- Brandão, F. and Mazzinghy, D. (2024). Código em python para sequenciamento de lavra considerando a variável geometalúrgica de tempo de processamento bloco a bloco. In *Anais do I Simpósio de Geometalurgia, Poster*, Belo Horizonte, Brasil. Universidade Federal de Minas Gerais. URL <https://www.simposiogeomet.org/>.
- Caccetta, L. and Hill, S. P. (2003). An application of branch and cut to open pit mine scheduling. *Journal of Global Optimization*, 27:349–365. URL <https://doi.org/10.1023/A:1024835022186>.
- Caccetta, L. (2007). Application of optimisation techniques in open pit mining. In Weintraub, A. et al., editors, *Handbook of Operations Research in Natural Resources*, p. 547–559. Springer, Boston, MA, USA. URL https://doi.org/10.1007/978-0-387-71815-6_29.
- Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E., and Rubio, E. (2012). A new algorithm for the open-pit mine production scheduling problem. *Operations Research*, 60(3):517–528. URL <https://doi.org/10.1287/opre.1120.1050>.
- Deutsch, M., Dağdelen, K., and Johnson, T. (2022). An open-source program for efficiently computing ultimate pit limits: Mineflow. *Natural Resources Research*, 31(1):1175–1187. URL <https://doi.org/10.1007/s11053-022-10035-w>.

- Espinoza, D., Goycoolea, M., Moreno, E., and Newman, A. (2013). Minelib: a library of open pit mining problems. *Annals of Operations Research*, 206(1):93–114. URL <https://doi.org/10.1007/s10479-012-1258-3>.
- Forrest, J., Ralphs, T., Vigerske, S., Santos, H. G., Forrest, J., Hafer, L., Kristjansson, B., jpfasano, EdwinStraver, Jan-Willem, Lubin, M., rlougee, a andre, jpgoncal1, Brito, S., h-i gassmann, Cristina, Saltzman, M., tosttost, and to st. (2024). coin-or/cbc: Release releases/2.10.12. URL <https://doi.org/10.5281/zenodo.13347261>. Zenodo.
- Hochbaum, D. S. (2008). The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research*, 56(4):992–1009. URL <https://doi.org/10.1287/opre.1080.0524>.
- Johnson, T. B. (1968). *Optimum open pit mine production scheduling*. Ph.D. dissertation, Operations Research Center, University of California, Berkeley. URL <https://apps.dtic.mil/sti/tr/pdf/AD0672094.pdf>.
- Jélvez, E., Morales, N., Nancel-Penard, P., Peypouquet, J., and Reyes, P. (2016). Aggregation heuristic for the open-pit block scheduling problem. *European Journal of Operational Research*, 249(3):1169–1177. URL <https://doi.org/10.1016/j.ejor.2015.10.044>.
- Lamghari, A., Dimitrakopoulos, R., and Ferland, J. A. (2015). A hybrid method based on linear programming and variable neighborhood descent for scheduling production in open-pit mines. *Journal of Global Optimization*, 63(3):555–582. URL <https://doi.org/10.1007/s10898-014-0185-z>.
- Lerchs, H. and Grossmann, I. F. (1965). Optimum design of open-pit mines. *Transactions on CIM*, 68:17–24.
- Muñoz, G. (2018). Mixed integer programming techniques for strategic open-pit mine planning. Presentation, Polytechnique Montréal / COSMO - McGill University. Available at Cosmo-Repository.
- Muñoz, G., Espinoza, D., Goycoolea, M., Moreno, E., Queyranne, M., and Letelier, O. R. (2018). A study of the bienstock–zuckerberg algorithm: applications in mining and resource constrained project scheduling. *Computational Optimization and Applications*, 69(2):501–534. URL <https://doi.org/10.1007/s10589-017-9946-1>.
- Samavati, M., Essam, D., Nehring, M., and Sarker, R. (2017). A methodology for the large scale multi-period precedence-constrained knapsack problem: an application in the mining industry. *International Journal of Production Economics*, 193:12–20. URL <https://doi.org/10.1016/j.ijpe.2017.06.025>.
- Santos, H. G., Noronha, F. F., et al. *Mixed Integer Linear Programming with Python*. Python-MIP Project, 2020. URL https://python-mip.readthedocs.io/_/downloads/en/latest/pdf/.