

MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Escola de Engenharia  
Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais (PPGEM)

**METODOLOGIA HEURÍSTICA PARA OTIMIZAÇÃO DO ARRANJO DE REALCES ABERTOS  
E PILARES VERTICAIS NO MÉTODO *SUBLEVEL OPEN STOPING***

Diogo Peixoto Cordova

Tese para a obtenção do título de Doutor em Engenharia

Porto Alegre, RS

2022

MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Escola de Engenharia  
Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais (PPGEM)

**METODOLOGIA HEURÍSTICA PARA OTIMIZAÇÃO DO ARRANJO DE REALCES ABERTOS  
E PILARES VERTICAIS NO MÉTODO *SUBLEVEL OPEN STOPING***

Diogo Peixoto Cordova  
**Mestre em Engenharia**

Tese desenvolvida dentro do Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e de Materiais (PPGE3M), como parte dos requisitos para a obtenção do Título de Doutor em Engenharia.

Orientador: Prof. Dr. André Cezar Zingano  
Co-orientador: Prof. Dr. Ítalo Gomes Gonçalves

Porto Alegre, RS  
2022

### CIP - Catalogação na Publicação

Cordova, Diogo Peixoto  
METODOLOGIA HEURÍSTICA PARA OTIMIZAÇÃO DO ARRANJO  
DE REALCES ABERTOS E PILARES VERTICAIS NO MÉTODO  
SUBLEVEL OPEN STOPPING / Diogo Peixoto Cordova. --  
2022.

197 f.

Orientador: Prof. Dr. André Cezar Zingano.

Coorientador: Prof. Dr. Ítalo Gomes Goncalves.

Tese (Doutorado) -- Universidade Federal do Rio  
Grande do Sul, Escola de Engenharia, Programa de  
Pós-Graduação em Engenharia de Minas, Metalúrgica e de  
Materiais, Porto Alegre, BR-RS, 2022.

1. modelagem numérica. 2. algoritmos evolutivos. 3.  
diluição. 4. falha de pilares. I. Zingano, Prof. Dr.  
André Cezar, orient. II. Goncalves, Prof. Dr. Ítalo  
Gomes, coorient. III. Título.

Diogo Peixoto Cordova

**METODOLOGIA HEURÍSTICA PARA OTIMIZAÇÃO DO ARRANJO DE REALCES ABERTOS  
E PILARES VERTICAIS NO MÉTODO *SUBLEVEL OPEN STOPING***

Esta tese foi julgada adequada para a obtenção do título de Doutor em Engenharia, na área de Tecnologia Mineral e Metalurgia Extrativa e aprovada em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Programa de Pós-Graduação em Engenharia de Minas, Metalúrgica e Materiais (PPGE3M) da Universidade Federal do Rio Grande do Sul

---

Prof. Dr. André Cezar Zingano

Orientador

---

Prof. Dr. Ítalo Gomes Gonçalves

Co-orientador

**BANCA EXAMINADORA**

Dr. Gian Franco Napa García (UPV/Espanha) \_\_\_\_\_

Dr. Rodrigo de Lemos Peroni (PPGE3M/UFRGS) \_\_\_\_\_

Dr. Paulo Salvadorette (DEMIM/UFRGS) \_\_\_\_\_



## **AGRADECIMENTOS**

Agradeço a minha mãe pelo suporte de sempre, meu pai (*in memoriam*) por compartilhar o interesse pela matemática e física, minha esposa Cris por toda a ajuda diária e companheirismo, agradeço a minha irmã Fernanda, meu irmão Rodrigo e meus sobrinhos Juliano e Bernardo pelos bons momentos. Aos meus amigos, pela compreensão diante de ausências e pelo apoio concedidos ao longo da realização deste trabalho.

Aos meus orientadores Prof. Dr. André Cezar Zingano, e Prof. Dr. Ítalo Gomes Gonçalves pela grande amizade, dedicação e ajuda na idealização deste trabalho.

Aos meus colegas e amigos da Universidade Federal do Pampa – UNIPAMPA pelo apoio mesmo nos momentos difíceis.

Aos membros da banca examinadora, pelo aceite, disponibilidade e pelas críticas construtivas ao trabalho.

A Universidade Federal do Rio Grande do Sul – UFRGS, pela oportunidade de estudo, de excelente nível acadêmico.

A todas as pessoas que de alguma forma auxiliaram na conclusão deste trabalho e que principalmente se dispõem a lê-lo.

## RESUMO

Um projeto otimizado de lavra deve priorizar o equilíbrio entre segurança e retorno econômico. Dessa forma, o algoritmo de otimização do *layout* de lavra precisa incluir no valor econômico, o custo do desempenho geomecânico do cenário otimizado, em função da diluição do minério e em função de problemas de falha de pilares. A meta deste trabalho é a proposta de uma metodologia integrada de otimização econômica do arranjo dos realces abertos (*open stopes*) e os pilares verticais (*rib pillars*), com o desenvolvimento de um algoritmo evolutivo, incorporando os critérios estabelecidos, como tamanho mínimo dos realces, tamanho mínimo dos pilares, diluição máxima aceitável e porcentagem máxima aceitável de falha nos pilares, que não comprometa a estabilidade global da lavra. Para verificar as metas estabelecidas, foi considerado um estudo de caso de um painel de lavra, de extensão de uma mina subterrânea de ouro, que utiliza o método de lavra de realces em subníveis abertos, com sequência de lavra descendente, onde pilares verticais são deixados de maneira permanente. A primeira etapa do algoritmo é gerar cenários aleatórios, que sejam válidos do ponto de vista das restrições estabelecidas. Na segunda etapa estes cenários serão sorteados para que o algoritmo genético realize cruzamentos, gerando novos indivíduos. Cada novo indivíduo gerado precisará ser quantificado economicamente, considerando o desempenho geomecânico fornecido pelo modelo numérico. O algoritmo é finalizado quando se atinge o número máximo de gerações definida, ou ainda se durante várias iterações não ocorre melhora no resultado. Ao se comparar o cenário ótimo, obtido pela empresa, considerando a metodologia tradicional, e o cenário ótimo obtido pela nova metodologia proposta se verifica que ainda que o lucro líquido seja menor no caso da metodologia proposta, a diluição potencial foi minimizada, assim como a porcentagem de falha dos pilares, ao mesmo tempo que preserva o mesmo teor médio. Também foi possível verificar que o algoritmo é capaz de compensar as variações dos parâmetros inseridos, selecionando cenários com tamanho médio de realces e pilares maiores ou menores, e conseqüentemente variando o número total de pilares e realces obtidos, sempre em função de um melhor resultado econômico. A metodologia proposta provou que é possível realizar uma otimização integrada, considerando os custos inerentes à lavra, juntamente com o custo do desempenho geomecânico do cenário.

**Palavras-chave:** modelagem numérica; algoritmos evolutivos; diluição; falha de pilares.

## ABSTRACT

An optimized mining project must always consider the balance between safety and economic return. Thus, the optimization algorithm needs to include in the economic value, the cost of the geomechanical performance of the optimized scenario, as a function of ore dilution and as a function of pillar failure problems. The aim of this work is to propose an integrated methodology for the economic optimization of the arrangement of open stopes and rib pillars, with the development of an evolutionary algorithm, incorporating established criteria, such as minimum size of stopes, minimum pillars size, maximum accepted dilution and maximum acceptable percentage of pillar failure, without compromising the global stability of the mine. To verify the established goals, it was considered a case study of a mining panel, an extension of an underground gold mine, which uses the method of sublevel open stopes, with a top-down mining sequence, where vertical pillars are left permanently. The first step of the algorithm is to generate random scenarios, which are valid from the point of view of the established restrictions. In the second stage, these scenarios will be randomly selected for the genetic algorithm to perform reproduction, generating new individuals. Each new individual generated will need to be economically quantified, considering the geomechanical performance provided by the numerical model. The algorithm is terminated when the maximum number of generations defined is reached, or even if during several iterations there is no improvement in the result. When comparing the optimal scenario, obtained by the company, considering the traditional methodology, and the optimal scenario obtained by the new proposed methodology is that even though the net profit is lower in the case of the proposed methodology, the potential dilution was minimized, as well as the percentage of pillar failure, while preserving the same average content. It was also possible to verify, in the other optimizations performed, that the algorithm can compensate for the variations of the parameters entered, selecting scenarios with an average size of larger or smaller stopes and pillars, and consequently varying the total number of pillars and stopes obtained, always depending on of a better economic result. The proposed methodology proved that it is possible to carry out an integrated optimization, considering the costs inherent to mining, together with the cost of the geomechanical performance of the scenario.

**Keywords:** numeric modelling; evolution algorithms; dilution; pillar failures.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxograma de um processo tradicional de projeto e planejamento de mina subterrânea. ....	20
Figura 2 - Fluxograma de proposta de planejamento e otimização de lavra integrada com parâmetros geomecânicos. ....	23
Figura 3 – Vista para o norte, onde se observa a topografia local com as antigas cavas a céu aberto já lavradas em cor cinza, os realces em cor verde e a infraestrutura de acessos em cor magenta. Ao leste se observa o painel de extensão San Gregório Leste, objeto do estudo de caso. ....	25
Figura 4 - Vista para o leste, onde se observa a topografia local com as antigas cavas a céu aberto já lavradas em cor cinza, os realces em cor verde e a infraestrutura de acessos em cor magenta. Ao leste se observa o painel de extensão San Gregório Leste, objeto do estudo de caso. ....	25
Figura 5 - Desenho esquemático da sequência de lavra do método de realces por subníveis longitudinais em veios estreitos, onde é possível observar as operações unitárias em diferentes subníveis. ....	32
Figura 6 - Vista longitudinal da sequência de lavra <i>top-down</i> , com utilização de pilares permanentes, com lavra em retirada das bordas para o centro, sem uso de enchimento. Os números representam a sequência da lavra. ....	33
Figura 7 – Vista longitudinal de uma sequência de lavra <i>bottom-up</i> e em sentido de retirada, das bordas para o centro, com uso de enchimento e pilares opcionais. Os números representam a sequência da lavra. ....	34
Figura 8 – Seção vertical, com exemplo de situação em que o controle da distância entre níveis se dá pela variação do mergulho do corpo de minério. ....	35
Figura 9 – Gráfico de estabilidade gerado a partir da equação de confinamento, com todos os casos históricos dos bancos de dados consolidados. ....	37
Figura 10 - Seção longitudinal dos fatores de segurança encontrados em um pilar vertical ( <i>rib pillar</i> ) entre dois realces abertos. ....	39
Figura 11 – Desenho explicativo sobre o cálculo do raio hidráulico. ....	40
Figura 12 - Ábaco para o cálculo do fator A. ....	42
Figura 13 - Ábaco para o cálculo do fator B. ....	42
Figura 14 - Ábaco para o cálculo do fator C. ....	43

Figura 15 - Gráfico de estabilidade. ....	44
Figura 16 - Ábaco para cálculo da densidade de cabos, segundo Nickson (1992).....	45
Figura 17 - Ábaco para cálculo do comprimento de cabos, segundo Nickson (1992).....	46
Figura 18 - Otimização do arranjo dos <i>cablebolts</i> para cada cenário de aplicação. ....	48
Figura 19 – Diagrama com os principais métodos de modelagem numérica .....	60
Figura 20 – Influência na relação tensão-deformação, de acordo com o modelo constitutivo considerado.....	64
Figura 21 - Representação esquemática da redistribuição de tensões e dos mecanismos de ruptura em escavações subterrâneas. ....	65
Figura 22 – Representação do círculo de Mohr, com a envoltória de ruptura linear de Mohr-Coulomb e os valores de tensões principais $\sigma_1'$ , $\sigma_3'$ e a tensão $\sigma_1''$ no limite da ruptura, mantendo-se a tensão $\sigma_3'$ fixa.....	66
Figura 23 – Exemplo de operação de cruzamento entre dois indivíduos e seus descendentes. ....	70
Figura 24 – Fluxograma mostrando a estrutura básica utilizada em um algoritmo genético. ....	71
Figura 25 - Fluxograma geral da metodologia do algoritmo otimizador proposto. ....	77
Figura 26 – Sintaxe do arquivo de geometria STL.....	78
Figura 27 - Tipos de mutações previstas no algoritmo genético para os pilares e realces. ..	84
Figura 28 – Diagrama de blocos demonstrando as etapas do algoritmo de otimização. ....	87
Figura 29 – <i>Layout</i> da janela principal de interface com o usuário.....	88
Figura 30 - Janela de progresso com informações como número de iterações por segundo, tempo de execução, tempo restante e tempo total estimado.....	89
Figura 31 – <i>Layout</i> da janela do sistema operacional para seleção de arquivos de entrada, os arquivos são filtrados pelo formato de arquivo esperado. ....	90
Figura 32 – <i>Layout</i> da aba de “Parâmetros geométricos” .....	91
Figura 33 – <i>Layout</i> da aba “Dados econômicos” .....	92
Figura 34 – <i>Layout</i> da aba principal “Geomecânica” e da aba secundária “FLAC3D” .....	93
Figura 35 - <i>Layout</i> da aba principal “Geomecânica” e da aba secundária “Propriedades” ....	94
Figura 36 - <i>Layout</i> da aba principal “Geomecânica” e da aba secundária “Tensões <i>in situ</i> ” ..	95
Figura 37 - <i>Layout</i> da aba principal “Geomecânica” e da aba secundária “Desempenho” ....	96
Figura 38 - <i>Layout</i> da aba “Algoritmo Genético” .....	97

Figura 39 - <i>Layout</i> da aba "Executar", com a janela <i>popup</i> em primeiro plano mostrando os detalhes do progresso do algoritmo, com o tempo restante, tempo transcorrido, tempo estimado e número de iterações.....	98
Figura 40 – <i>Layout</i> da janela para abrir arquivo salvo com formulário preenchido, em formato JSON.....	99
Figura 41 - <i>Layout</i> da janela para salvar arquivo com formulário preenchido, em formato JSON.....	99
Figura 42 – Exemplo de gráfico <i>boxplot</i> mostrando o relatório evolutivo a cada nova geração do algoritmo genético. O eixo vertical é o valor da função <i>fitness (net profit)</i> e o eixo horizontal o número de gerações.....	100
Figura 43 – Exemplo de gráficos dos atributos dos cenários produzidos a cada geração após fim da otimização.....	101
Figura 44 – Diagrama de blocos de códigos executados sequencialmente, para modelagem numérica automatizada, no FLAC3D.....	103
Figura 45 - Exemplo de gráfico gerado automaticamente com a geometria do arranjo de realces e pilares otimizado, além dos subníveis fornecidos pelo usuário.....	104
Figura 46 - Seção do modelo numérico, onde observa-se a discretização gradativa dos elementos nos locais de interesse.....	105
Figura 47 – Vista isométrica do modelo numérico inicial, com a visualização das tensões verticais.....	106
Figura 48 - Gráficos do somatório das forças internas do modelo: a) somatório de forças após a realização da escavação instantânea das galerias e dos realces, até atingir um novo equilíbrio; b) somatório de forças após a realização da escavação das galerias e escavação gradual dos realces, até atingir um novo equilíbrio.....	107
Figura 49 - Figura isométrica com uma seção 3D dos isovalores da tensão $\sigma_3$ , após a escavação.....	108
Figura 50 - Exemplo de plotagem gerada automaticamente mostrando as isosuperfícies de elementos sob relaxamento de tensões no <i>hangingwall</i> .....	109
Figura 51 - Exemplo de gráfico gerado automaticamente mostrando as isosuperfícies de elementos sob falha nos pilares.....	110
Figura 52 - Mapa de localização da Ilha Cristalina de Rivera .....	112

Figura 53 – Valores de $k$ para diferentes módulos de deformação, variando com a profundidade.....	115
Figura 54 - Gráfico de estabilidade, para as faces dos realces, considerando as variações estabelecidas representadas pela seta cor azul. ....	116
Figura 55 – Avaliação do fator ELOS para o <i>hangingwall</i> .....	117
Figura 56 – Gráfico empírico de estabilidade de pilares, a seta em azul representa a zona para os pilares possíveis, considerando as variações dos parâmetros. ....	118
Figura 57 – Vista para o norte, do arranjo ótimo dos realces e pilares, considerando a metodologia tradicional utilizada. Os realces são mostrados em verde, os pilares em amarelo e os subníveis em cor magenta. O código identificador de cada realce e de cada pilar é mostrado na figura.....	119
Figura 58 - Vista para o norte, do contorno do arranjo ótimo dos realces em linha tracejada cor preta, considerando a metodologia tradicional utilizada. As linhas tracejadas em cor magenta representam os contornos dos subníveis. Os teores em g/t são demonstrados segundo a legenda.....	120
Figura 59 - Vista isométrica com o layout de realces e pilares tido como ótimo para a metodologia considerada pela empresa. Os resultados do modelo numérico, quanto a diluição potencial e a porcentagem de falha dos pilares (PFP) são mostrados. ....	123
Figura 60 - Vista isométrica com zonas de falha no <i>hangingwall</i> em vermelho, para o cenário tido como ótimo para a metodologia considerada pela empresa.....	124
Figura 61 - Vista isométrica com zonas de falha nos pilares em vermelho, para o cenário tido como ótimo para a metodologia considerada pela empresa. Os pilares P22, P23 e P15, da porção central, são observados com porcentagem de falha excessiva. ....	125
Figura 62 – Gráfico do tipo diagrama de caixas, mostrando a evolução da população ao longo das gerações do algoritmo genético.....	127
Figura 63 - Vista para o norte, do arranjo ótimo dos realces e pilares, considerando a nova metodologia proposta. Os realces são mostrados em verde, os pilares em amarelo e os subníveis em cor magenta. O código identificador de cada realce e de cada pilar é mostrado na figura.....	128
Figura 64 - Vista lateral, olhando para o norte, do contorno do arranjo ótimo dos realces em linha tracejada cor preta, considerando a nova metodologia de otimização. As	

linhas tracejadas em cor magenta representam os contornos dos subníveis. Os teores em g/t são demonstrados segundo a legenda.....	128
Figura 65 - Vista isométrica com zonas de falha no <i>hangingwall</i> em vermelho, para o cenário base otimizado.....	132
Figura 66 - Vista isométrica com zonas de falha nos pilares em vermelho, para o cenário base otimizado. ....	133
Figura 67 – Gráfico do lucro líquido em função da eficiência de suporte para ambos os cenários.....	134
Figura 68 - Gráfico comparando cenários estocásticos com o algoritmo genético.....	135
Figura 69 – Relatório evolutivo para o caso 2 (k=1).....	137
Figura 70 – Relatório evolutivo para o caso 3 (k=2).....	138
Figura 71 – Relatório evolutivo para o caso 4 ( <i>overburden</i> = 20 MPa).....	138
Figura 72 – Vista isométricas para o arranjo de realces e pilares otimizado para os 4 casos considerados.....	140
Figura 73 – Vistas isométricas com as isosuperfícies do potencial de diluição no <i>hangingwall</i> , em vermelho, para os cenários ótimos dos 4 casos considerados.....	141
Figura 74 – Vistas isométricas com as isosuperfícies de zonas de falha nos pilares, em vermelho, para os cenários ótimos dos 4 casos considerados. ....	142
Figura 75 – Vista isométrica mostrando a zona do subdomínio 1 considerado. ....	143
Figura 76 - Gráfico do tipo diagrama de caixas, mostrando a evolução da população ao longo das gerações do algoritmo genético, para o cenário com subdomínio 1. ....	144
Figura 77 - Vista isométrica com zonas de falha no <i>hangingwall</i> em vermelho, para o cenário otimizado com o subdomínio 1 na zona mineralizada. ....	144
Figura 78 - Vista isométrica com zonas de falha nos pilares em vermelho, para o caso com um subdomínio de menor resistência dentro da zona mineralizada. Os pilares com maior número de elementos com falha, dentro do subdomínio, estão circulos em azul.....	146



## LISTA DE TABELAS

Tabela 1 - Evolução dos principais algoritmos otimizadores de limites de realces subterrâneos. ....	49
Tabela 2 - Variáveis de entrada inseridas pelo usuário na interface gráfica do programa....	81
Tabela 3 - Propriedades geomecânicas consideradas para o estudo de caso. ....	114
Tabela 4 – Parâmetros econômicos e operacionais considerados para o estudo de caso.	118
Tabela 5 – Propriedades dos realces do cenário ótimo considerado pela empresa, utilizando a metodologia tradicional.....	121
Tabela 6 - Propriedades dos pilares do cenário ótimo considerado pela empresa, utilizando a metodologia tradicional.....	122
Tabela 7 – Parâmetros do algoritmo genético considerados para o estudo de caso.....	126
Tabela 8 - Propriedades dos realces do cenário ótimo obtido, considerando a nova metodologia proposta. ....	130
Tabela 9 - Propriedades dos realces do cenário ótimo obtido, considerando a nova metodologia proposta. ....	131
Tabela 10 – Comparação entre o resultado do cenário ótimo pela metodologia tradicional e do cenário ótimo obtido pela metodologia proposta.....	134
Tabela 11 – Diferença de tempos computacionais conforme valor de <i>unbalanced forces</i> adotado.....	136
Tabela 12 – Comparativo de cenários otimizados, considerando variações das tensões <i>in situ</i> . ....	139
Tabela 13 - Propriedades geomecânicas consideradas para o estudo de caso.....	143

## LISTA DE ABREVIATURAS E SIGLAS

ELOS	<i>Equivalent linear over-break sloughage</i>
FLAC	<i>Fast Lagrangian Analysis of Continua</i>
FW	<i>Footwall</i>
GSI	<i>Geological Strength Index</i>
HW	<i>Hangingwall</i>
LHD	<i>Load-Haul-Dump</i>
MDF	Método das diferenças finitas
MEC	Método dos elementos de contorno
MED	Método dos elementos discretos
MEF	Método dos elementos finitos
PAF	Porcentagem máxima aceitável de falha em pilares
PFP	Porcentagem de falha em pilares
RCU	Resistência a compressão Uniaxial
RMR	<i>Rock Mass Rating</i>
ROM	<i>Run of mine</i>
RQD	<i>Rock Quality Designation</i>
SSR	<i>Shear-Strain-Ratio</i>
UCS	<i>Uniaxial Compressive Strength</i>
US\$	Dólar americano
VPL	Valor Presente Líquido

## LISTA DE SÍMBOLOS

$\nu$	Razão de Poisson
B	Módulo de deformação volumétrica do maciço
c	Coessão
E	Módulo de Young para o maciço rochoso
$E_i$	Módulo de Young para a rocha intacta
G	Módulo de corte do maciço rochoso
$\gamma$	Peso específico do material
$\sigma_{cm}$	Resistência a compressão do maciço rochoso
$\sigma_{ci}$	Resistência a compressão de rocha intacta
$\sigma_{tm}$	Resistência a tração do maciço rochoso
$\sigma_1$	Tensão principal maior
$\sigma_2$	Tensão principal intermediária
$\sigma_3$	Tensão principal menor
$\sigma_H$	Tensão normal horizontal
$\sigma_V$	Tensão normal vertical
$\phi$	Ângulo de fricção interna
k	Razão entre tensões principais <i>in situ</i>
N	Número de estabilidade
N'	Número de estabilidade modificado
Q	Qualidade do maciço rochoso (Sistema Q)
Q'	Qualidade do maciço rochoso (Sistema Q modificado)
Jn	Número de famílias de descontinuidade (Sistema Q)
Jr	Número relativo à rugosidade da descontinuidade (Sistema Q)
Ja	Número relativo à alteração da descontinuidade (Sistema Q)

## SUMÁRIO

<b>CAPÍTULO 1</b> .....	<b>18</b>
<b>1 INTRODUÇÃO DO PROBLEMA</b> .....	<b>18</b>
1.1 CONTEXTUALIZAÇÃO DO PROBLEMA.....	18
<b>1.1.1 Planejamento de mina tradicional</b> .....	<b>18</b>
<b>1.1.2 Proposta de planejamento de lavra integrado</b> .....	<b>21</b>
1.2 MOTIVAÇÃO.....	24
1.3 META DO TRABALHO.....	26
1.4 OBJETIVOS .....	26
1.5 CONTRIBUIÇÕES E INOVAÇÕES ESPERADAS .....	26
1.6 ORGANIZAÇÃO DA TESE .....	27
<b>CAPÍTULO 2</b> .....	<b>28</b>
<b>2 REVISÃO BIBLIOGRÁFICA</b> .....	<b>28</b>
2.1 AVANÇO DOS SOFTWARES E ALGORITMOS DE OTIMIZAÇÃO DE LAVRA .....	28
2.2 EVOLUÇÃO DA MECÂNICA DAS ROCHAS .....	29
2.3 MÉTODO DE LAVRA DE REALCES EM SUBNÍVEIS.....	30
<b>2.3.1 Variação do método em corpos de minério estreitos</b> .....	<b>31</b>
<b>2.3.2 Distância entre subníveis</b> .....	<b>34</b>
<b>2.3.3 Dimensionamento de pilares</b> .....	<b>35</b>
<b>2.3.4 Dimensionamento de realces</b> .....	<b>39</b>
<b>2.3.5 Suporte do <i>hangingwall</i></b> .....	<b>47</b>
2.4 OTIMIZADORES ECONÔMICOS DOS LIMITES GEOMÉTRICOS DE REALCES .....	48
<b>2.4.1 Algoritmos exatos</b> .....	<b>50</b>
<b>2.4.2 Algoritmos heurísticos</b> .....	<b>54</b>
<b>2.4.3 Implementações em softwares comerciais</b> .....	<b>58</b>
2.5 MODELAGEM NUMÉRICA.....	59

2.5.1	Métodos contínuos .....	60
2.5.2	Métodos descontínuos .....	62
2.5.3	Modelo constitutivo.....	63
2.5.4	Critério de falha do maciço rochoso .....	64
2.6	ALGORITMOS GENÉTICOS.....	67
2.6.1	História dos algoritmos genéticos .....	67
2.6.2	Vantagens e aplicações .....	68
2.6.3	Estrutura básica.....	69
2.7	TRABALHOS RECENTES RELACIONADOS AO TEMA.....	71
<b>CAPÍTULO 3 .....</b>		<b>74</b>
<b>3</b>	<b>METODOLOGIA.....</b>	<b>74</b>
3.1	RECURSOS NECESSÁRIOS.....	77
3.2	ALGORITMO EM LINGUAGEM <i>PYTHON</i> .....	78
3.2.1	Funções para manipulação das geometrias .....	78
3.2.2	Comunicação via <i>Socket</i> com o <i>FLAC3D</i> .....	79
3.2.3	Função geradora de cenários estocásticos válidos .....	79
3.2.4	Função benefício ( <i>profit</i> ) .....	80
3.2.5	Algoritmo genético .....	82
3.2.6	Interface gráfica do usuário .....	88
3.2.7	Resultados e arquivos de saída.....	100
3.3	MODELAGEM NUMÉRICA AUTOMATIZADA.....	102
3.3.1	Importação de arquivos.....	103
3.3.2	Discretização pelo método <i>Octree</i> .....	104
3.3.3	Parâmetros de inicialização do modelo .....	105
3.3.4	Critério de falha do <i>hangingwall</i> .....	107
3.3.5	Critério de falha dos pilares .....	109

<b>CAPÍTULO 4</b> .....	<b>111</b>
<b>4 APLICAÇÃO PRÁTICA DA METODOLOGIA PROPOSTA</b> .....	<b>111</b>
4.1 ESTUDO DE CASO .....	111
4.1.1 Localização e geologia.....	111
4.1.2 Equipamentos e parâmetros operacionais .....	113
4.1.3 Propriedades e domínios geotécnicos .....	113
4.1.4 Tensões <i>in situ</i> .....	114
4.1.5 Estabilidade dos realces e do <i>hangingwall</i> .....	115
4.1.6 Estabilidade dos pilares.....	117
4.1.7 Parâmetros econômicos e operacionais.....	118
4.1.8 Cenário ótimo adotado pela metodologia tradicional .....	119
4.2 RESULTADOS E EFICIÊNCIAS CONSIDERANDO A METODOLOGIA PROPOSTA.....	125
4.2.1 Parâmetros base considerados para o algoritmo genético.....	126
4.2.2 Resultados da otimização pelo algoritmo genético – cenário base.....	126
4.2.3 Eficiência e tempos de otimização.....	134
4.3 SENSIBILIDADE DO ALGORITMO .....	136
4.3.1 Parâmetros de tensões <i>in situ</i> .....	136
4.3.2 Inclusão de subdomínios geotécnicos .....	142
<b>CAPÍTULO 5</b> .....	<b>147</b>
<b>5 CONSIDERAÇÕES FINAIS</b> .....	<b>147</b>
5.1 CONTRIBUIÇÕES DA TESE .....	147
5.2 LIMITAÇÕES DA METODOLOGIA PROPOSTA.....	147
5.3 VARIAÇÕES DO ALGORITMO EM TRABALHOS FUTUROS .....	148
<b>6 REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>150</b>
<b>APÊNDICE A – CÓDIGOS ESCRITOS EM LINGUAGEM PYTHON</b> .....	<b>157</b>
<b>APÊNDICE B – CÓDIGOS ESCRITOS NO FLAC3D</b> .....	<b>182</b>

# Capítulo 1

## 1 Introdução do problema

Neste capítulo será realizada a contextualização do problema, considerando a motivação a partir do estudo de caso, também apresentados os objetivos e as metas do trabalho, além das contribuições esperadas.

### 1.1 Contextualização do problema

Neste tópico o problema estudado será contextualizado sob suas diferentes óticas e abordagens. Os tópicos a seguir relacionam a sequência tradicional utilizada no planejamento e projeto de mina, os avanços dos pacotes de *software* de planejamento e otimização de lavra, a evolução da mecânica das rochas e a proposta de metodologia para uma otimização integrada.

#### 1.1.1 Planejamento de mina tradicional

O planejamento define os caminhos para garantir a segurança da operação, eficiência, continuidade da produção e recuperação econômica do minério, enquanto considera a vida útil da mina e suas implicações no planejamento e projeto de curto prazo. Além disso, ajuda a manter a garantia de produção a longo prazo, enquanto satisfaz o retorno econômico considerado para o projeto (VILLAESCUSA, 2014).

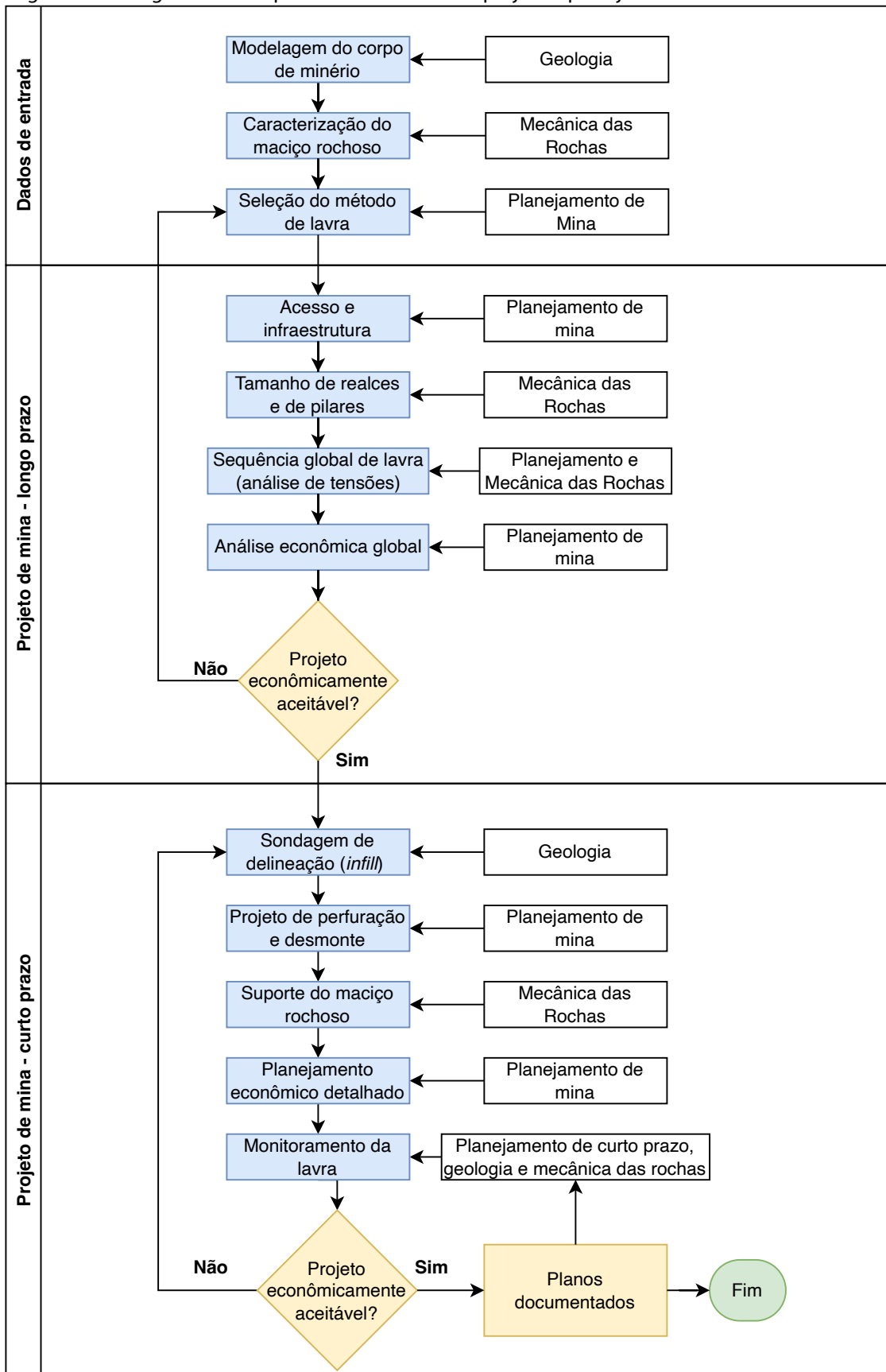
Os principais dados de entrada utilizados no planejamento de lavra são o modelo geológico, contendo o entendimento dos corpos geológicos e suas geometrias espaciais, modelo de blocos, contendo a estimativa das variáveis de interesse, e o modelo geomecânico, contendo os domínios geotécnicos e a distribuição espacial das propriedades de resistência e de qualidade do maciço rochoso. Esses modelos irão subsidiar inicialmente a escolha do método de lavra mais adequado, a definição da escala de produção a ser adotada e a definição da extração ou não de um determinado painel ou realce e, se ele for lavrado, em que momento da sequência. A Figura 1 demonstra o fluxograma tradicional deste processo de projeto e planejamento de mina subterrânea em suas fases de curto e longo prazo.

Para a tomada de decisão da lavra de um realce ou painel, deve-se aplicar um teor de corte ou realizar o cálculo da função benefício, atribuindo um valor econômico positivo

ou negativo a este realce analisado. Para este cálculo, os dados básicos de entrada necessários são os custos de processo, custos de lavra, custos gerais e administrativos, custo de refino, royalties, diluição esperada, recuperação do minério na lavra, recuperação do processo de beneficiamento, toneladas ROM (*run-of-mine*), teor médio e o preço de venda da *commodity*. Havendo qualquer variação nesses parâmetros o cálculo precisará novamente ser realizado e os realces reconsiderados. Importante salientar que o custo de lavra é função da distância de transporte, ou seja, cresce com a profundidade ou com o aumento da distância do realce extraído em relação a galeria central de transporte no nível de produção.



Figura 1 – Fluxograma de um processo tradicional de projeto e planejamento de mina subterrânea.



Fonte: Adaptado de Villaescusa (2014).

### 1.1.2 Proposta de planejamento de lavra integrado

A prática tradicional para o projeto de mina é completar cada fase sequencialmente. Desta forma, cada fase é planejada separadamente e a solução de uma fase constitui a entrada e o ponto de partida para a próxima fase. Isso continuará até que o projeto de mina seja concluído (LITTLE *et al.*, 2013).

Normalmente a etapa de planejamento de longo prazo de uma mina subterrânea sob o método por subníveis (*sublevel stope*), inicia com a definição da separação entre níveis, vãos máximos suportáveis (*spans*) para os realces abertos, além do dimensionamento empírico dos pilares verticais (*rib pillars*), pilares base (*sill pillars*) e pilares de teto (*crown pillars*). A partir dessas informações um algoritmo otimizador de realces é utilizado, por exemplo, o algoritmo *Floating Stope* (ALFORD, 1995; ALFORD *et al.*, 2007). Esse algoritmo irá gerar os contornos de realces economicamente positivos, respeitando os parâmetros de entrada estabelecidos. De posse do resultado do otimizador, o engenheiro de planejamento de longo prazo irá projetar os pilares verticais e horizontais, em um arranjo regular, agrupando as informações no software de mineração e sequenciando a extração no tempo, com objetivo de otimizar o retorno econômico pelo VPL (Valor Presente Líquido) global.

Posteriormente, o arranjo e o sequenciamento estabelecidos são validados por meio de modelos numéricos, com o objetivo de verificar a redistribuição de tensões nos pilares e galerias de acesso, analisar a diluição esperada no *hangingwall*, além de verificar a estabilidade global dos pilares e dos realces, sugerindo ajustes ou definido a necessidade de suporte mais adequado. Já no planejamento de curto prazo, ocorre um maior detalhamento das informações geotécnicas disponíveis, dessa forma as dimensões dos pilares podem sofrer ajustes e novos modelos numéricos precisam ser realizados.

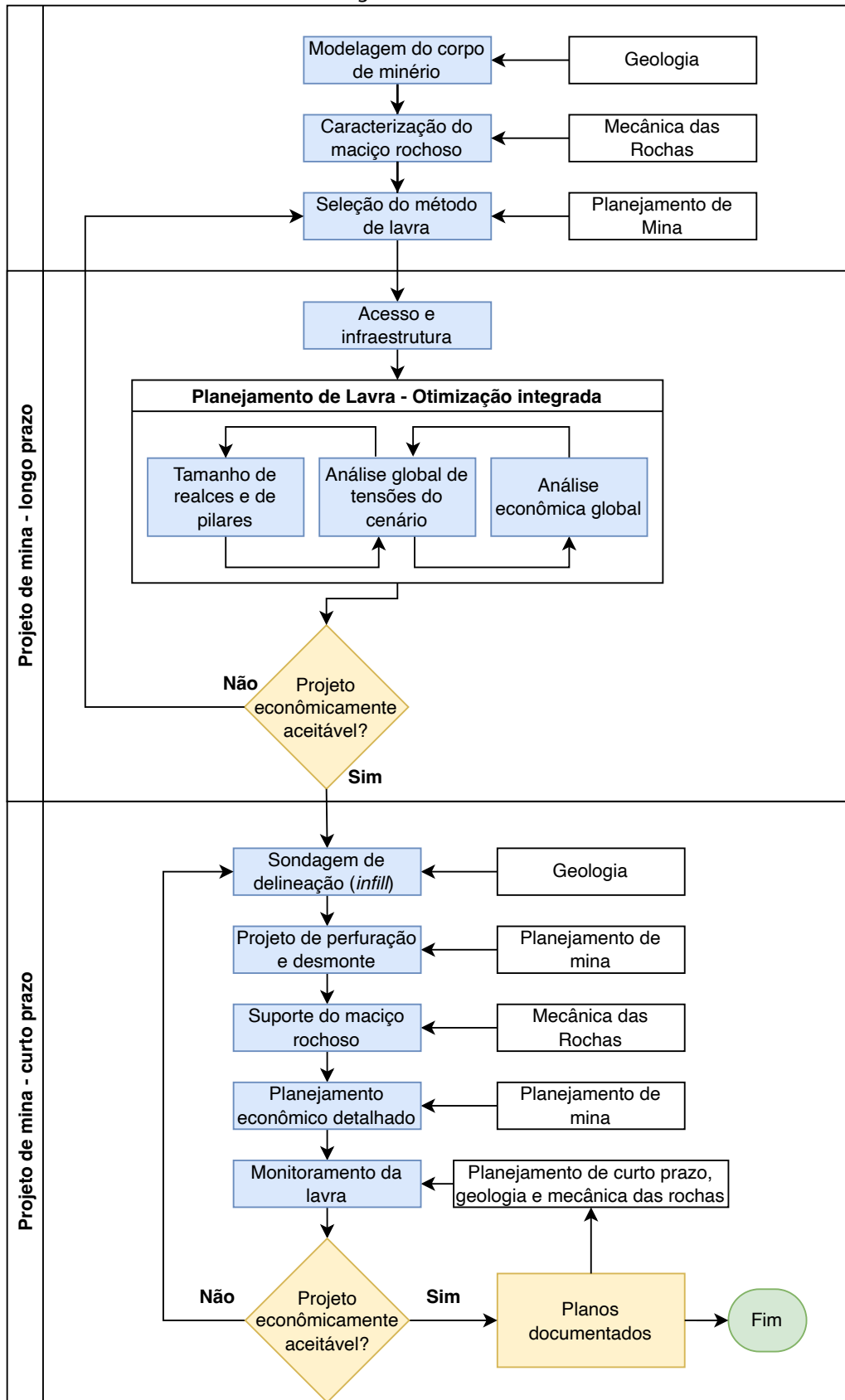
Essa abordagem, para um planejamento de longo prazo, produz algumas ineficiências, como a generalização do tamanho de pilares e de realces, que não necessariamente precisam ser constantes, pois, ao longo do depósito existem variações nas propriedades geomecânicas do maciço rochoso, em ângulos de mergulho do corpo, espessura etc. Sendo assim, a obtenção de um arranjo irregular de realces e pilares para o painel de lavra, considerando dimensões variáveis, tende a produzir um cenário mais otimizado, melhorando o retorno financeiro do painel projetado. Isso ocorre devido ao fato de se priorizar a definição de pilares em zonas de menor teor e de maior necessidade do

ponto de vista geomecânico, diminuindo a diluição potencial nos realces. Ainda sim é necessário que o cenário ótimo obtido seja operacionalizável.

De acordo com estudos sobre várias operações de mina ao longo de vários anos, Whittle (2010) concluiu que a otimização integrada de diferentes fases da cadeia de valor da mineração pode aumentar o valor das operações. Neste ponto é que se observa claramente que a otimização de realces, o dimensionamento de pilares e a otimização econômica deixam de ser aspectos dissociados entre si, impossibilitando que sejam realizados separadamente ou ainda sequencialmente. Sendo assim, os otimizadores atualmente implementados tendem a produzir otimizações que encontram pontos de máximos locais, mas que podem estar distantes do máximo global, quando considerados outros fatores, dentre eles principalmente o desempenho geomecânico.

Um projeto de lavra deve priorizar sempre o equilíbrio entre segurança e retorno econômico. Esses objetivos podem ser cumpridos, se o algoritmo de otimização incluir de forma clara, no valor econômico, o custo do desempenho geomecânico do cenário otimizado, em função da diluição do minério e em função de problemas de falha de pilares. A Figura 2 traz o fluxograma proposto para esse conceito de otimização integrada de lavra.

Figura 2 - Fluxograma de proposta de planejamento e otimização de lavra integrada com parâmetros geomecânicos.



Fonte: Autoria própria.

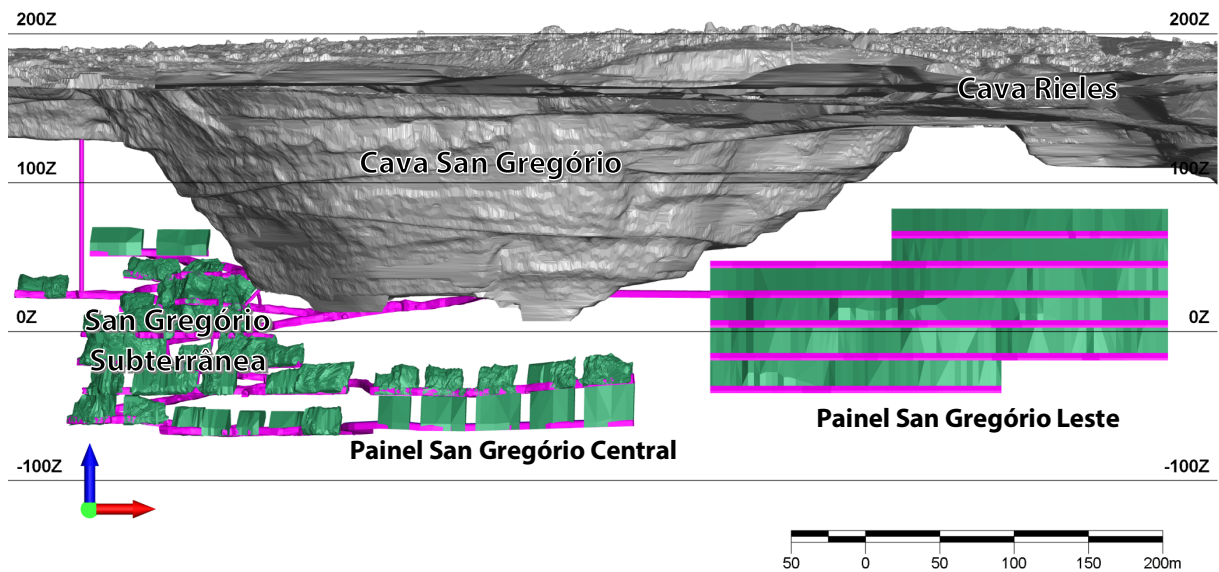
## 1.2 Motivação

A principal motivação para o desenvolvimento desta tese surge da intenção de integrar a otimização econômica com a otimização geomecânica, o que até então nas práticas da indústria, não é realizado. Com isso, permitindo ganho de tempo para obtenção de cenários positivos, mesmo em situações geomecânicas complexas, ou ainda em situações econômicas adversas, com custos de lavra elevados, baixo preço de venda do minério, ou mesmo em depósitos de baixo teor médio.

O estudo de caso motivador trata-se de um projeto de extensão de um painel de lavra ao leste de uma mina subterrânea de ouro, com baixo teor médio e vida útil de aproximadamente 2 anos. Devido a sua curta vida útil e por critério econômico da empresa, o projeto definia a adoção do método de realces abertos por subníveis (*sublevel open stoping*), lavrados no sentido descendente (*top-down*). A escolha por este método possibilita o adiantamento do início da produção de realces, distribuindo o desenvolvimento horizontal da rampa principal de acesso e dos acessos aos níveis, em estéril, ao longo da vida útil da operação.

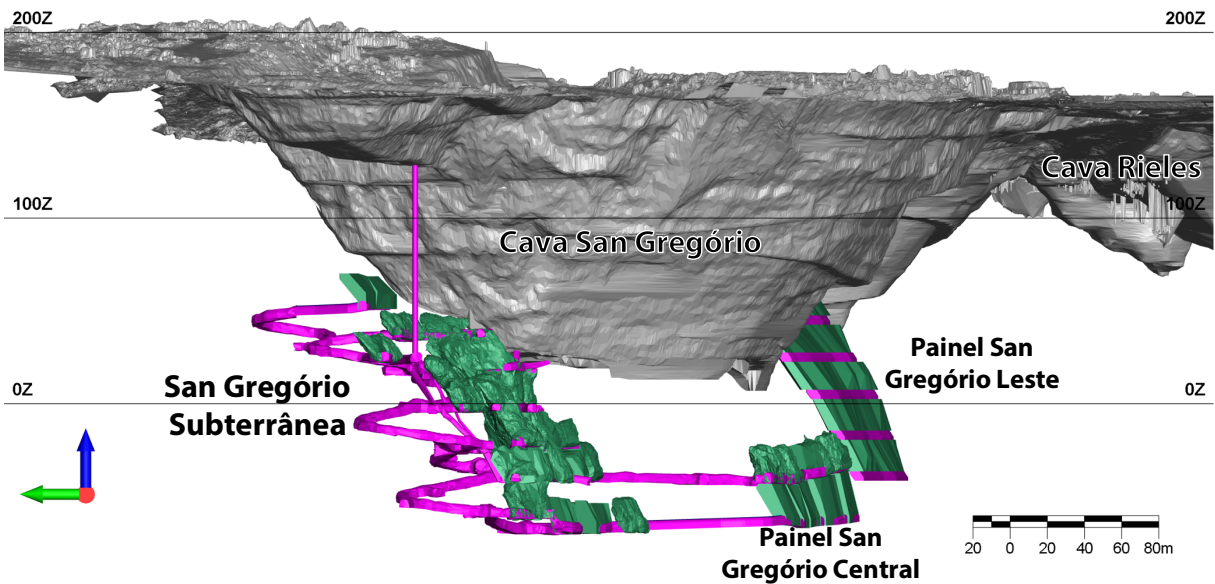
A profundidade média do depósito é de 200m. O corpo mineralizado mergulha entre 60° e 70° e sua espessura é relativamente estreita, variando de 5 a 7m. Pilares verticais (*rib pillars*) são deixados entre os realces, que permanecerão abertos sem nenhum tipo de enchimento posterior. A separação entre os subníveis foi definida em aproximadamente 15m (de piso a teto), devido o limite/eficiência dos equipamentos de perfuração, além da questão da variação do mergulho do *hangingwall*. Este painel projetado, nomeado de San Gregório Leste, faz parte do complexo São Gregório, de cavas a céu aberto e mina subterrânea, conforme pode ser observado na Figura 3 e Figura 4 a seguir.

Figura 3 – Vista para o norte, onde se observa a topografia local com as antigas cavas a céu aberto já lavradas em cor cinza, os realces em cor verde e a infraestrutura de acessos em cor magenta. Ao leste se observa o painel de extensão San Gregório Leste, objeto do estudo de caso.



Fonte: Autoria própria.

Figura 4 - Vista para o leste, onde se observa a topografia local com as antigas cavas a céu aberto já lavradas em cor cinza, os realces em cor verde e a infraestrutura de acessos em cor magenta. Ao leste se observa o painel de extensão San Gregório Leste, objeto do estudo de caso.



Fonte: Autoria própria.

O projeto de extensão era considerado internamente pela empresa como sendo de retorno marginal, já que grande parte dos custos de capital já haviam sido pagos pela lava da mina principal, San Gregório Subterrânea, com aproveitamento de equipamentos e de infraestrutura. Dessa forma, para a lava do painel de extensão projetado, surge o desafio da otimização das dimensões dos realces e dos pilares e de seus arranjos, de forma a maximizar

o retorno econômico, o controle da diluição potencial e a garantia de estabilidade das escavações considerando a porcentagem de falha nos elementos dos pilares.

### 1.3 Meta do trabalho

A meta deste trabalho é a proposta de uma metodologia integrada de otimização econômica do arranjo dos realces abertos (*open stopes*) e os pilares verticais (*rib pillars*), com o desenvolvimento de um algoritmo evolutivo, incorporando os critérios estabelecidos, como tamanho mínimo dos realces, tamanho mínimo dos pilares, diluição máxima esperada e porcentagem máxima aceitável de falha nos pilares.

### 1.4 Objetivos

Alguns objetivos parciais foram definidos para alcançar a meta proposta, sendo listados a seguir:

- Obter previamente a otimização dos contornos econômicos do corpo de minério;
- Analisar as dimensões mínimas operacionais para os pilares verticais e os realces;
- Desenvolver um código em linguagem *Python*, que permita a comunicação direta e autônoma com o software de modelagem numérica;
- Estabelecer uma função benefício que seja capaz de atribuir o valor econômico dos cenários analisados, incluindo o desempenho geomecânico;
- Analisar a eficiência do algoritmo evolutivo, comparando-o com simulações estocásticas de cenários e com o cenário ótimo definido pela empresa pela sequência tradicional de otimização;
- Analisar a sensibilidade do algoritmo proposto, conforme as variações dos parâmetros de entrada, como tensões *in situ*, e inclusão de domínios geomecânicos.

### 1.5 Contribuições e inovações esperadas

A tese proposta busca trazer as seguintes contribuições:

- Desenvolver uma metodologia integrada de otimização econômica do arranjo dos realces e dos pilares, com o desenvolvimento de um algoritmo evolutivo, incorporando os critérios estabelecidos, como tamanho mínimo dos realces, tamanho mínimo dos pilares, diluição máxima esperada e porcentagem máxima aceitável de falha nos pilares.

- Demonstrar a real necessidade de realização da otimização geomecânica integrada da otimização econômica, de forma que o cenário otimizado pela metodologia proposta, apresente teor médio maior ou igual do que o cenário pelo método tradicional, porém com menor potencial de diluição e porcentagem de falhas nos pilares.
- Proporcionar uma interface direta entre os tradicionais pacotes de *software* de mineração e o software de modelagem numérica FLAC3D, utilizando linguagem *Python*.
- Proporcionar bom desempenho de processamento, ao compará-lo com simulações estocásticas de cenários e com o próprio cenário ótimo definido pela empresa.

#### 1.6 Organização da tese

O Capítulo 2 apresenta a revisão bibliográfica dos aspectos teóricos e práticos sobre os algoritmos de otimização de realces subterrâneos. Além disso, serão apresentados aspectos teóricos sobre as técnicas de modelagem numérica e sobre os algoritmos genéticos. Também será realizada uma revisão geral do método de lavra de realces por subníveis.

O Capítulo 3 irá apresentar a metodologia proposta da tese, considerando a implementação de um algoritmo computacional em linguagem *Python* e com uso de modelagem numérica para a otimização econômica do arranjo de realces e pilares. Essa metodologia utilizará como ponto de partida a otimização do contorno econômico dos realces, segundo os algoritmos apresentados no Capítulo 2.

O Capítulo 4 apresenta a implementação prática do algoritmo por meio do estudo de caso proposto. São demonstrados os resultados obtidos, comparando-se a eficiência computacional do algoritmo genético com o método estocástico e com o cenário ótimo obtido pela empresa, além de simular variações do problema utilizando o mesmo estudo de caso.

O Capítulo 5 apresenta as considerações finais, contribuições da tese e sugestões para trabalhos futuros baseados no tema.



# Capítulo 2

## 2 Revisão Bibliográfica

Neste capítulo serão revisadas as principais referências disponíveis com relação ao método de lavra de realces em subníveis. Em maior detalhe será revisada a sua aplicação em corpos de minério estreitos com sentido de lavra descendente. Além disso, neste capítulo também serão revisados os principais otimizadores do contorno econômico de realces e suas evoluções em paralelo aos pacotes de *software* de mineração, os métodos de modelagem numérica, a técnica de otimização de problemas por algoritmos genéticos e as publicações mais recentes relacionadas a algoritmos de otimização subterrânea, com ou sem uso de modelagem numérica.

### 2.1 Avanço dos pacotes de *software* e algoritmos de otimização de lavra

Nos últimos 50 anos os processos necessários para o projeto de mina e sequenciamento de lavra eram realizados de forma manual por geólogos e engenheiros de minas, com uma visão mais focada no curto prazo da operação. Quanto ao uso de computadores, na década de 1980 as empresas de grande porte começaram a implantar o uso de pacotes de *software* específicos para os projetos de minas, com o lançamento da primeira versão do *software* Datamine® em 1983, escrito em linguagem de programação Fortran (DATAMINE, 2020). Em 1984 ocorre o lançamento do *software* Vulcan®, também desenvolvido em linguagem Fortran (MYERS, 2014). Já em 1986 é lançado o *software* Micromine® (MICROMINE, 2016).

De maneira geral, nessa época os pacotes de *software* ainda eram essencialmente ferramentas de desenho assistido, da sigla em inglês CAD (*Computer Aided Design*), sendo o planejamento e otimização realizados pelo usuário, sem muitas ferramentas específicas disponíveis nos pacotes de *software*. Já nas décadas seguintes, com a expansão do uso dos computadores, o processo de planejamento passou a ser amplamente apoiado no uso de *software* comercial, com a inclusão de ferramentas de otimização e sequenciamento da lavra.

A otimização é uma etapa extremamente importante e dinâmica do planejamento de lavra, já que diferentes cenários afetarão diretamente na geometria das escavações, na seqüência de extração e conseqüentemente no retorno econômico. Do ponto de vista das

minerações subterrâneas, as otimizações são basicamente realizadas em três áreas principais: a otimização dos desenvolvimentos horizontais, do formato dos realces e, por fim, a otimização do sequenciamento da produção, que permite maximizar o projeto quanto ao retorno financeiro e ainda no nivelamento da produção (TOPAL, 2008).

Os primeiros algoritmos de otimização, da geometria dos realces, foram desenvolvidos na década de 1970, tendo sido amplamente baseados nos já existentes algoritmos de otimização para cavas a céu aberto. Riddle (1977 citado por Ataee-pour, 2000) desenvolveu um algoritmo matemático de programação dinâmica em duas dimensões, para a otimização da geometria de minas subterrâneas com o método de lavra *block caving*, baseado no algoritmo para cavas a céu aberto de Johnson e Sharp (1971).

Ao longo dos anos diversas técnicas foram desenvolvidas para otimizar os limites econômicos dos realces, porém, todas elas têm certas deficiências que limitam seu uso e confiabilidade (TOPAL e SENS, 2010). Um dos algoritmos mais utilizados ainda hoje é o algoritmo heurístico *Floating Stope* (ALFORD, 1995), o qual teve sua primeira versão implementada no ano de 2011 e que pode ser encontrado nos pacotes de *software* comerciais Datamine®, Deswik® e Vulcan® (ALFORD MINING SYSTEMS, 2016). Em sua versão inicial, o algoritmo implementado não produzia nenhum arranjo de realces, ele apenas indicava a área na qual eles poderiam ser colocados contendo um teor mais alto do que o teor de corte. Alford et. al (2007) incluíram no algoritmo de *Floating Stope* mais dados de entrada como: separação entre níveis, dimensões mínimas e máximas, tamanhos mínimos de pilares adjacentes, diluição nos contatos com as paredes, teor de corte, ângulos mínimos e máximos de *hangingwall* e *footwall*, provendo um resultado automatizado para os realces, com os tamanhos, formatos e localizações.

Ainda sim, de maneira geral os algoritmos de otimização utilizam de dados geotécnicos de entrada bastante limitados e determinísticos. O que leva a produção de um cenário ótimo baseado em valores médios, em termos do tamanho e localização dos realces e pilares.

## 2.2 Evolução da mecânica das rochas

Na década de 1960, os desenvolvimentos no estudo do comportamento das rochas eram bem significativos devido a ocorrência de rupturas catastróficas que demonstraram que as escavações em rochas da época, estavam exigindo conhecimentos além das

habilidades em prever as consequências das ações tomadas (TERZAGHI; VOIGHT, 1979 citado por HOEK, 2007). Esses acontecimentos iriam ditar o desenvolvimento da mecânica das rochas como ciência nas décadas seguintes.

Na década de 1970 surgem os principais sistemas empíricos de classificação para os maciços rochosos. O sistema RMR desenvolvido por Bieniawski (1973), posteriormente atualizado em Bieniawski (1989) e em Celada et al. (2014) e o Sistema Q desenvolvido por Barton *et al.* (1974) e atualizado em Grimstad e Barton (1993) e em Grimstad *et al.* (2002).

Esses sistemas, inicialmente mais voltados a escavações de túneis civis, permitiam a definição da necessidade de suporte necessário e as dimensões máximas em vãos não suportados. Mathews *et al.* (1981 citado por Potvin, 1988), cria um método empírico voltado a estabilidade de realces subterrâneos. Ao longo do tempo este método sofreu adaptações por diferentes autores como Potvin (1988) e Nickson (1992) buscando aumentar sua confiabilidade no dimensionamento de realces estáveis utilizando suporte com tirantes-cabo (*cablebolts*). Clark e Pakalnis (1997) quantificaram o desempenho observado de realces usando a medida ELOS (*Equivalent linear over-break sloughage*). Eles então utilizaram os dados para desenvolver linhas de projeto, para prever o desempenho provável de realces, quanto a diluição, utilizando a classificação do valor do ELOS.

O dimensionamento e a análise de estabilidade de escavações estão atualmente alicerçados no uso de modelagem numérica por meio das aplicações computacionais, em complemento às já tradicionais ferramentas empíricas de dimensionamento citadas. No projeto de realces, os métodos numéricos são favorecidos em relação aos métodos tradicionais, pois permitem adicionar complexidades extras as análises, tais como: efeitos das tensões *in situ* e induzidas por escavações próximas, geometrias complexas de escavação, comportamento não linear de deformação do maciço rochoso, anisotropias e influência de grandes descontinuidades. Os métodos numéricos contínuos são geralmente preferidos devido à sua facilidade de uso e, dependendo da complexidade do modelo e dos recursos computacionais, permitem várias simulações dentro de um tempo razoável.

### 2.3 Método de lavra de realces em subníveis

O método de realces por subníveis é considerado um método autossuportado (HARTMAN, 1987; HAMRIN, 2001). Segundo Potvin e Hudyma (2000) este também é o método mais comum empregado em minas subterrâneas de rocha dura no Canadá. Ele é

caracterizado por cortes de realces relativamente pequenos, da ordem de 20.000 a 100.000 toneladas, com lavra relativamente rápida. As condições da rocha encaixante, de comportamento tipicamente ruim a regular, são os parâmetros a serem considerados no dimensionamento dos realces. A estabilidade do maciço rochoso é mantida pelo projeto de realces estáveis com o emprego de tirantes cabo (*cablebolts*), tanto para a capa (*hangingwall*) quanto para a lapa (*footwall*). O controle de diluição também é uma prioridade neste método (POTVIN e HUDYMA, 2000).

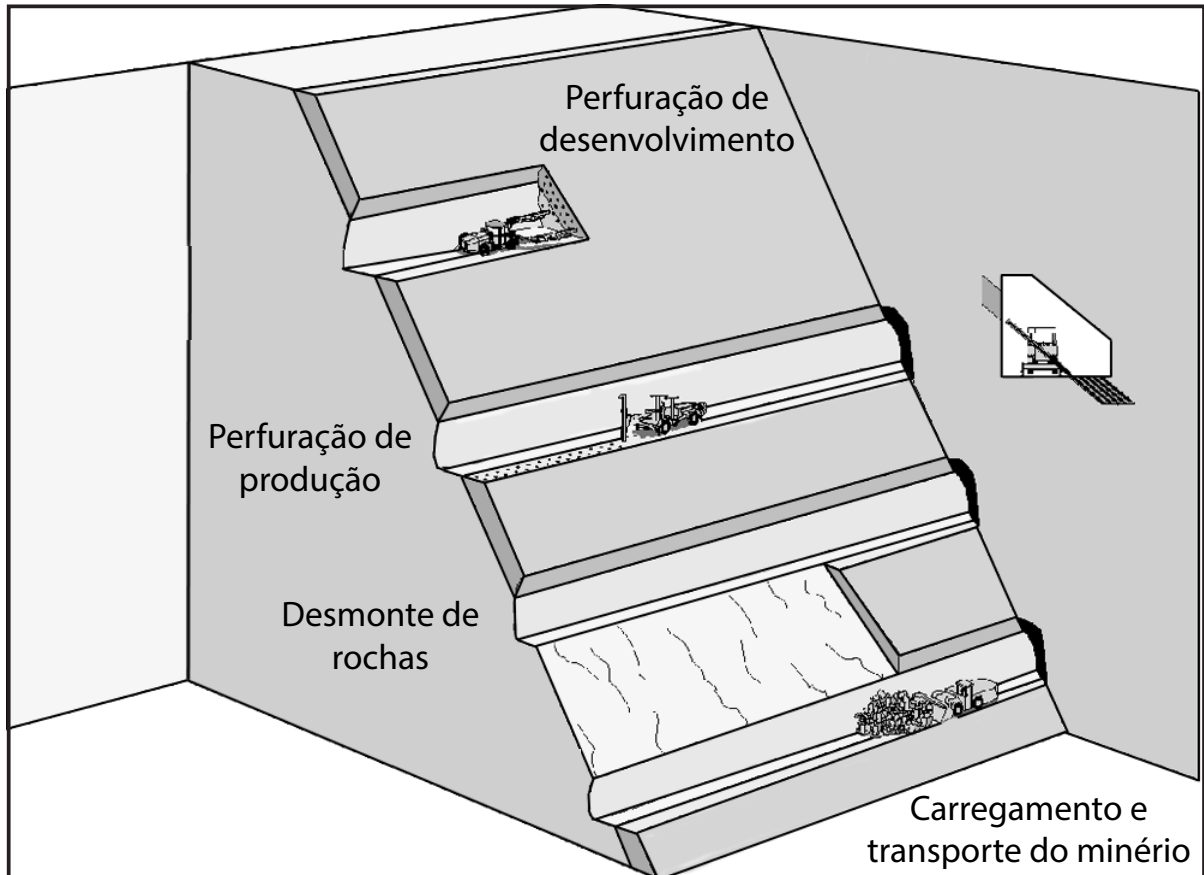
Os corpos de minério são tipicamente pequenos, de alto teor, profundos e irregulares, sendo o desenvolvimento normalmente avançando de baixo para cima. Em vários casos altas tensões *in situ* são observadas devido a profundidade elevada. No caso de ocorrência de tensões elevadas nos pilares, pode se optar pela recuperação dos pilares e uso de enchimento cimentado em situações de altos teores e corpos largos (POTVIN e HUDYMA, 2000).

Neste trabalho a variação do método de realce por subníveis que será melhor estudada é a de lavra em veios estreitos com alto mergulho, de acordo com o estudo de caso selecionado. O algoritmo proposto nesta tese é aplicado na abordagem de sequenciamento descendente, pois é nesta situação em que os pilares verticais são abandonados, e com o aprofundamento da mina passam a sofrer maiores carregamentos.

### **2.3.1 Variação do método em corpos de minério estreitos**

No caso de corpos de minério com mergulho acentuado e veios estreitos, a forma de acesso mais comum é pelo desenvolvimento de galerias a partir da rampa principal, através da lapa e perpendiculares ao corpo de minério. Depois são desenvolvidos os níveis dentro da zona mineralizada, ao longo do eixo (*strike*) do corpo. A Figura 5 mostra as operações unitárias sendo realizadas em diferentes subníveis, de maneira simultânea, com a lavra no sentido longitudinal. A extração pode ser realizada em retrocesso na direção da galeria de acesso central do corpo, tanto na direção ascendente (*bottom-up*) ou descendente (*top-down*).

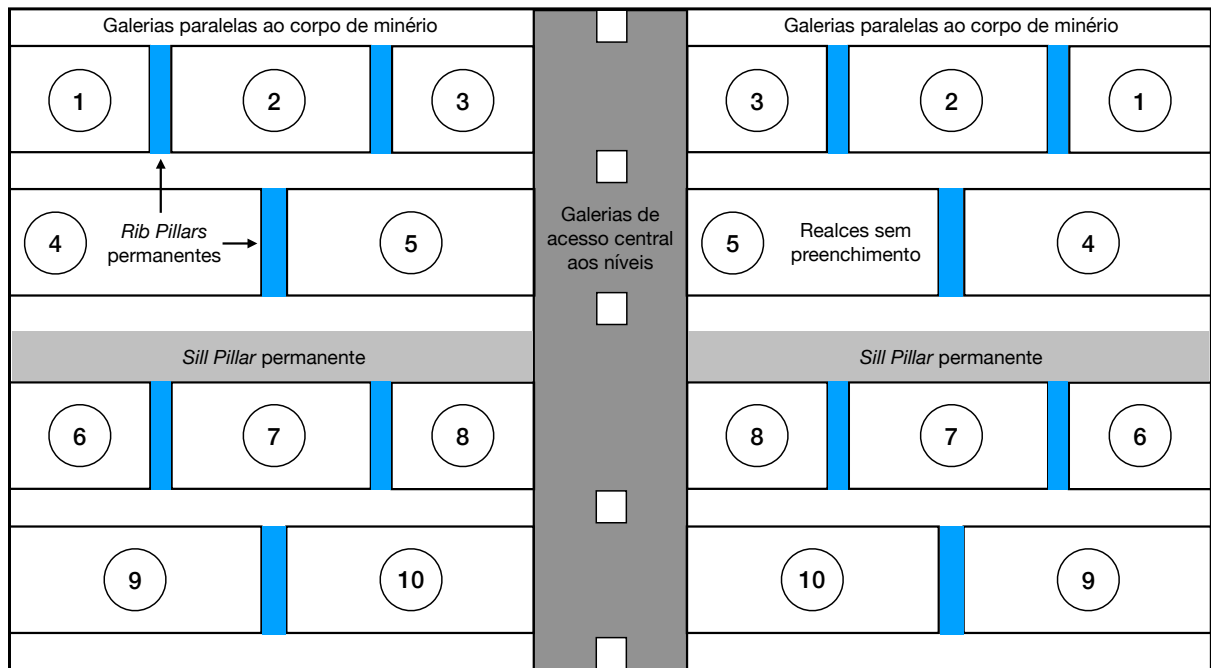
Figura 5 - Desenho esquemático da sequência de lavra do método de realces por subníveis longitudinais em veios estreitos, onde é possível observar as operações unitárias em diferentes subníveis.



Fonte: Adaptado de Hamrin (2001).

Uma sequência de lavra *top-down* normalmente necessita a utilização de pilares verticais (*rib pillars*) permanentes, situação idêntica a do estudo de caso selecionado, e que pode ser observada na Figura 6. O objetivo dos pilares verticais é minimizar a diluição entre os realces lavrados ao longo do eixo (*strike*) do corpo de minério. Importante salientar que alguns pilares horizontais (*sill pillars*), entre os níveis, podem ser necessários para controlar a estabilidade global e para conter enchimento não consolidado que possa vir a ser colocado nos realces superiores, em caso de necessidade de estabilização, de acordo com o avanço da lavra de forma descendente (VILLAESCUSA, 2014).

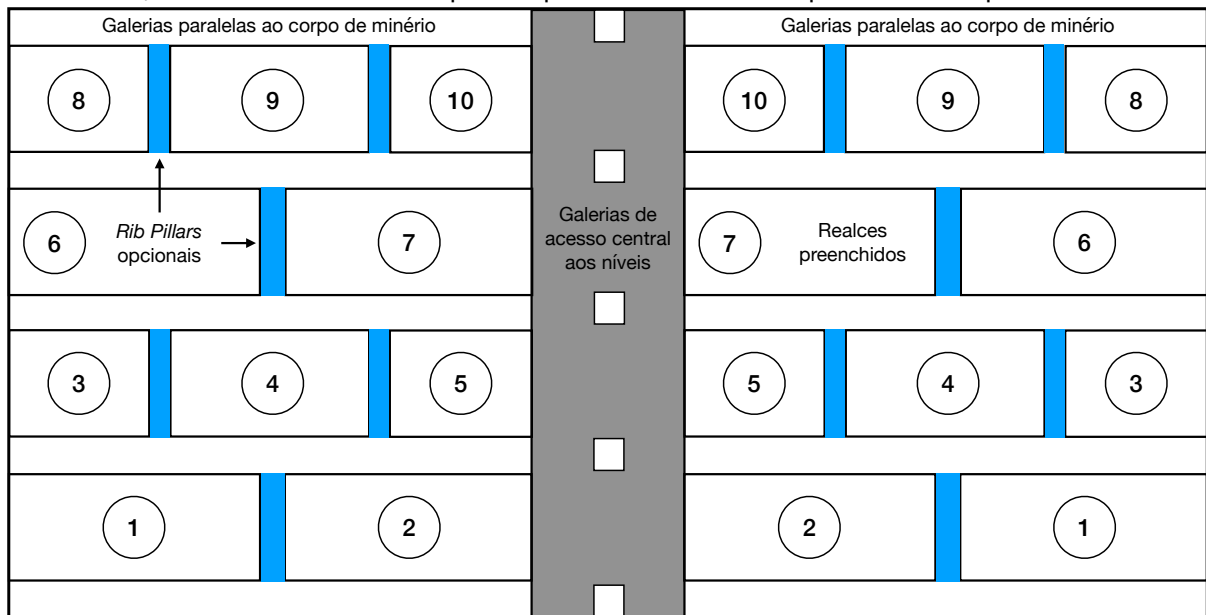
Figura 6 - Vista longitudinal da sequência de lavra *top-down*, com utilização de pilares permanentes, com lavra em retirada das bordas para o centro, sem uso de enchimento. Os números representam a sequência da lavra.



Fonte: Adaptado de Villaescusa (2014).

Por sua vez, a adoção de uma sequência de lavra ascendente (*bottom-up*) necessita preencher os realces abertos, de maneira a prover um piso de trabalho, de acordo com o avanço ascendente da lavra, conforme o observado na Figura 7. Neste caso, a necessidade tanto de *sill pillars* quanto de *rib pillars* é minimizada devido ao confinamento proporcionado pelo enchimento. Esse método ao mesmo tempo que garante um aumento significativo da recuperação global do corpo de minério, agrega um custo adicional devido o uso de enchimento, ou seja, em casos de corpos de minério de baixo teor pode não ser o melhor cenário.

Figura 7 – Vista longitudinal de uma sequência de lavra *bottom-up* e em sentido de retirada, das bordas para o centro, com uso de enchimento e pilares opcionais. Os números representam a sequência da lavra.



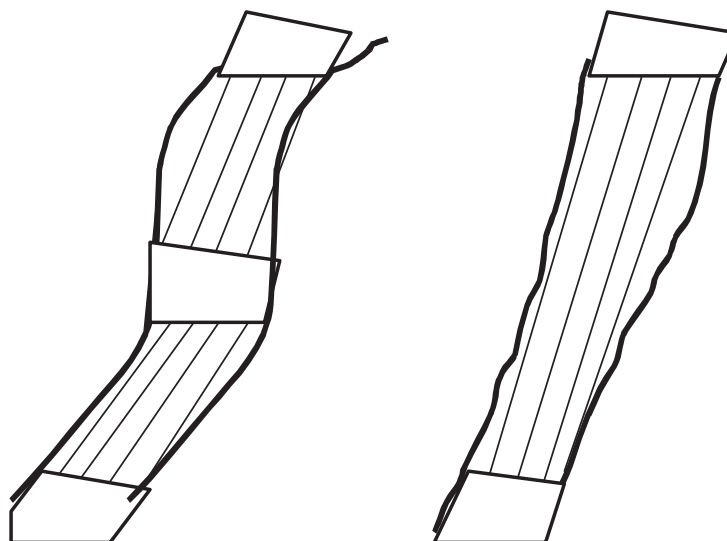
Fonte: Adaptado de Villaescusa (2014).

### 2.3.2 Distância entre subníveis

A escolha pela maior distância entre subníveis fornece o menor custo de desenvolvimento, por tonelada de minério, para o método de lavra escolhido. Entretanto, essa escolha não pode ser embasada somente no custo de desenvolvimento. Na maioria dos casos, essa decisão é baseada em fatores como a estabilidade do *hangingwall*, custo de desenvolvimento, variações no mergulho do corpo de minério e na capacidade do equipamento de perfuração disponível (HAMRIN, 2001). A Figura 8 a seguir demonstra uma situação clara, onde torna-se inviável, para o corpo de minério em questão, a escolha de uma distância entre níveis muito grande.

Os critérios subjacentes devem ser o controle da diluição e da recuperação de minério, pois intervalos de subníveis maiores reduzem o desenvolvimento horizontal necessário, mas podem aumentar a diluição e diminuir a recuperação. Conseqüentemente, é necessária uma avaliação do impacto econômico da recuperação e da diluição de minério para cada intervalo de subnível específico (VILLAESCUSA, 2014).

Figura 8 – Seção vertical, com exemplo de situação em que o controle da distância entre níveis se dá pela variação do mergulho do corpo de minério.



Fonte: Adaptado de Villaescusa (2014).

### 2.3.3 Dimensionamento de pilares

Um método de lavra com suporte natural, como é o caso do método de realces em subníveis, necessita de um adequado dimensionamento de pilares, controlando assim a estabilidade nas zonas de influência da lavra, ao redor das escavações. De maneira geral a estabilidade local dos realces abertos e a estabilidade dos pilares, podem ser analisadas e consideradas de maneira separada. Porém, a estabilidade efetiva dos pilares pode ser esperada como sendo relacionada tanto com as dimensões individuais dos pilares, tanto com a localização deles no corpo de minério, o que também depende do dimensionamento de realces realizado.

A decisão em deixar pilares dispostos ao longo do corpo de minério, pode resultar em uma esterilização temporária ou permanente das reservas. Dessa forma, um dimensionamento otimizado, de maneira econômica, implica em um menor comprometimento de reservas, ao mesmo tempo que cumpre com os requerimentos de estabilidade global da mina (BRADY e BROWN, 2005).

Em geral, a resistência dos pilares e a estabilidade são controladas por vários fatores que incluem a geologia estrutural, resistência a compressão, deformabilidade do maciço rochoso, as dimensões do pilar, a relação largura/altura, grau de confinamento, porcentagem de extração e qualidade da lavra (VILLAESCUSA, 2014).



### 2.3.3.1 Método empírico

Os desenvolvimentos iniciais no dimensionamento empírico de pilares, eram focados no método de câmaras e pilares em minas de carvão. Posteriormente, métodos empíricos aplicados para rochas duras se tornaram disponíveis e passaram a ser implementados para a lavra de realces em subníveis (VILLAESCUSA, 2014).

Lunder e Pakalnis (1997) combinaram em seu trabalho diversos bancos de dados anteriores, representando todo o conhecimento empírico para pilares em rocha dura. Dentre os trabalhos anteriores estão Lunder (1994), Hudyma (1988), Von Kimmelman *et al.* (1984), e Hedley e Grant (1972).

Hudyma (1988) analisou dados de pilares verticais disponíveis de minas canadenses e colocou as observações em um gráfico em função do carregamento normalizado nos pilares dividido pela resistência a compressão uniaxial (RCU), e pela relação largura/altura dos pilares. O banco de dados incorporou uma grande variedade de tipos de rochas e carregamento de pilares derivados de modelos numéricos elásticos tridimensionais. O autor dividiu o gráfico em três zonas gerais: ruptura, transição e estável. Os dados mostraram que pilares achatados sob baixas tensões são estáveis, conforme quadrante inferior direito da Figura 9. Os pilares começam a ficar menos estáveis quando se movem para a situação do quadrante superior esquerdo da referida figura. O banco de dados também incluiu 13 estudos de casos em que os pilares eram originalmente estáveis e posteriormente apresentaram plasticidade. Esses casos foram observados em detalhe e se moveram ao longo das três zonas do gráfico. Hudyma (1988) também sugeriu que o gráfico pudesse ser usado para prever a plasticidade de pilares no método de realces em subníveis abertos.

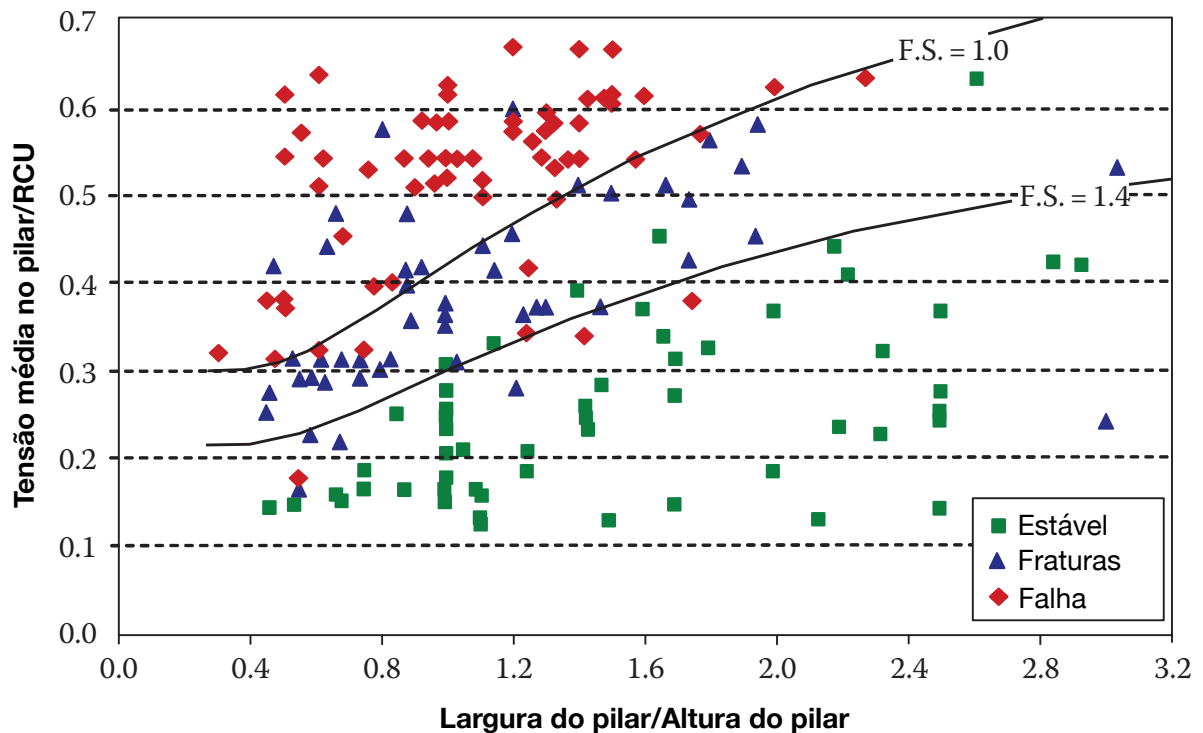
Cada um dos bancos de dados listados utiliza alguma forma de classificação de estabilidade de pilares. De maneira a trazer esses dados para a mesma referência, uma classificação simplificada foi desenvolvida, identificando apenas se a condição é estável, com deslocamentos ou com ruptura (LUNDER, 1994; LUNDER e PAKALNIS, 1997).

Duas fórmulas que podem ser utilizadas na estimativa da resistência dos pilares foram desenvolvidas por Lunder (1994), incluindo a “equação logarítmica do efeito da forma” e a “equação do confinamento”. Ambas as equações são virtualmente idênticas quando representadas no gráfico de estabilidade. Entretanto, a diferença entre a equação logarítmica é que ela é puramente empírica, já a equação do confinamento é uma modificação do critério de Mohr-Coulomb (LUNDER, 1994).

A equação do confinamento utiliza o termo de fricção, baseado no confinamento médio do pilar, determinado por modelos numéricos. Essa equação tem uma taxa de sucesso maior na determinação da performance do pilar em comparação com os métodos empíricos anteriores. Porém, assim como os métodos empíricos, também necessita de uma calibração específica para o estudo de caso a ser analisado, com objetivo de aumento no grau de confiança.

As linhas dividindo cada uma das classificações de estabilidade dos pilares, foram definidas com um fator de segurança correspondente, como pode ser observado na Figura 9. Essa definição é baseada na consideração de que a linha que divide os pilares instáveis e com rupturas tem um fator de segurança de 1. Usando isso como base, foi determinado que a linha de transição da zona de condição instável para condição estável, tem um fator de segurança calculado em 1.4 (LUNDER, 1994).

Figura 9 – Gráfico de estabilidade gerado a partir da equação de confinamento, com todos os casos históricos dos bancos de dados consolidados.



Fonte: Adaptado de Lunder (1994).

### 2.3.3.2 Modelagem numérica

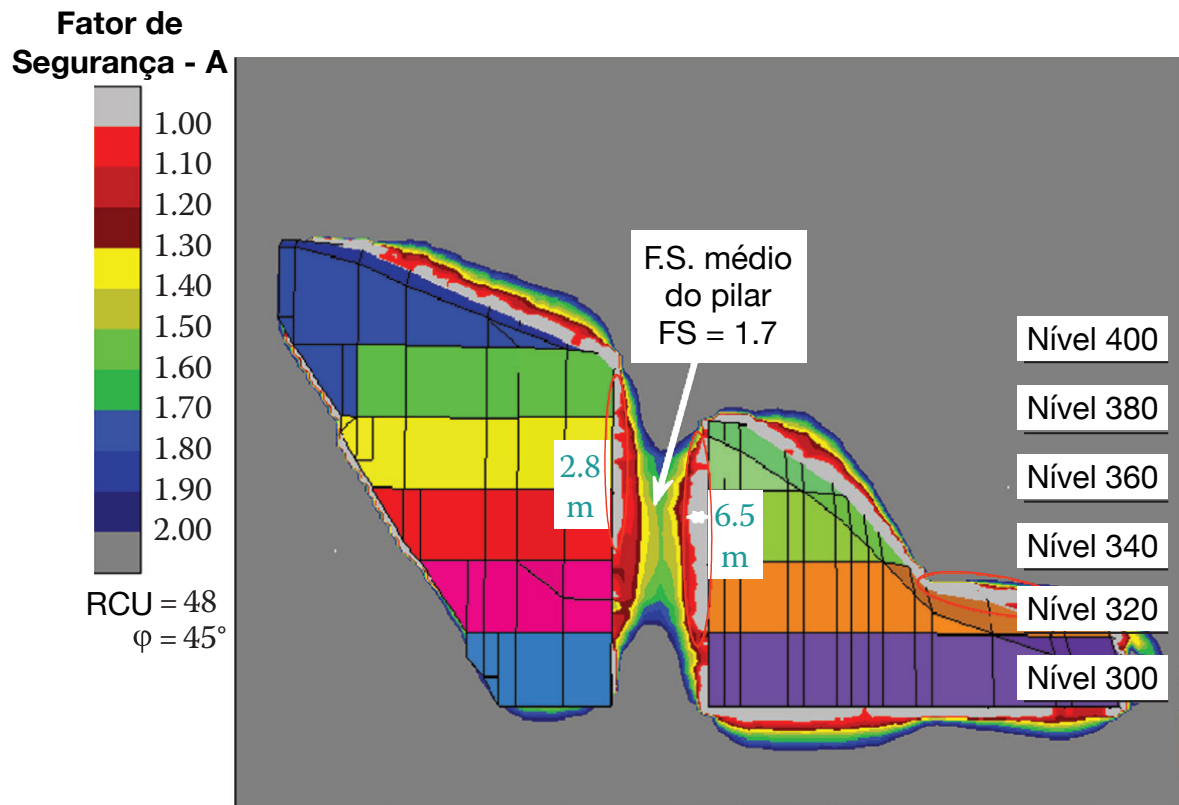
Modelos numéricos tridimensionais de comportamento linear elástico ou mesmo não elásticos podem ser utilizados no dimensionamento dos pilares. Para análises lineares elásticas, as geometrias completas são inseridas, mas os modelos normalmente são construídos com apenas um modelo constitutivo, pois a inclusão de vários modelos geralmente traz um efeito muito pequeno no resultado da distribuição de tensões. Alguns modelos também podem receber a inclusão de descontinuidades geológicas importantes. O resultado de saída dos modelos numéricos geralmente é de fácil interpretação, pois podem utilizar curvas contornos, ou isosuperfícies das tensões principais ou dos fatores de segurança (VILLAESCUSA, 2014).

As tensões induzidas pela abertura de realces são normalmente distribuídas de forma não uniforme nos pilares. Essa não uniformidade, pode causar aumento da concentração de tensões em algumas regiões dos pilares, o que pode causar rupturas mesmo considerando que na média as tensões verticais estejam dentro do limite de resistência dos pilares. Isso pode levar ao início de uma ruptura progressiva, na medida que pode se propagar ao longo do pilar (BRADY e BROWN, 2005).

A Figura 10 mostra a distribuição dos fatores de segurança ao longo de um pilar vertical entre dois realces abertos. No núcleo do pilar se observam fatores de segurança bem acima de um, já nas bordas do pilar os fatores são iguais ou menores do que um o que indica uma situação de ruptura dos elementos nas bordas do pilar.

Em outros casos, a resistência de pico de um pilar pode até ser excedida, mesmo assim o pilar pode continuar apresentando alguma capacidade de resistência e continue garantindo a estabilidade regional sem apresentar grandes deformações (VILLAESCUSA, 2014). No caso de um pilar romper e perder totalmente sua capacidade de suporte, suas tensões serão transferidas aos pilares adjacentes, fazendo com que estes sejam por sua vez sobrecarregados. Esse processo sucessivo de carregamento pode levar a uma situação progressiva de efeito “dominó” onde grandes áreas da mina podem colapsar.

Figura 10 - Seção longitudinal dos fatores de segurança encontrados em um pilar vertical (*rib pillar*) entre dois realces abertos.



Fonte: Adaptado de Villaescusa (2014).

Os modelos elásticos tridimensionais fornecem uma representação razoável da redistribuição de tensões no processo de abertura de realces, porém os pilares estão sujeitos a diferentes estados de tensões, particularmente nas faces expostas onde ocorrem desconfinamentos e na redistribuição de tensões no núcleo do pilar. Sendo assim, em uma análise ideal, seriam necessários modelos constitutivos não lineares, para se analisar com exatidão a resposta dos pilares, após as deformações plásticas permanentes (BECK *et al.*, 2010).

### 2.3.4 Dimensionamento de realces

Um dos fatores limitantes que afetam o projeto das escavações subterrâneas é o vão livre máximo que o maciço rochoso pode suportar sem ruptura. Essa ruptura pode ocorrer de várias maneiras, através de planos de descontinuidade, rupturas da rocha intacta por tração ou compressão, ou ainda em zonas de falhas geológicas.

Em muitos corpos de minério o volume que pode ser escavado com segurança, de maneira com que não ocorram rupturas nas paredes ou no teto do realces, é várias vezes

menor do que a dimensão do corpo de minério em si. Consequentemente, uma série de realces individuais precisam ser escavados para garantir a extração de toda a zona mineralizada (VILLAESCUSA, 2003).

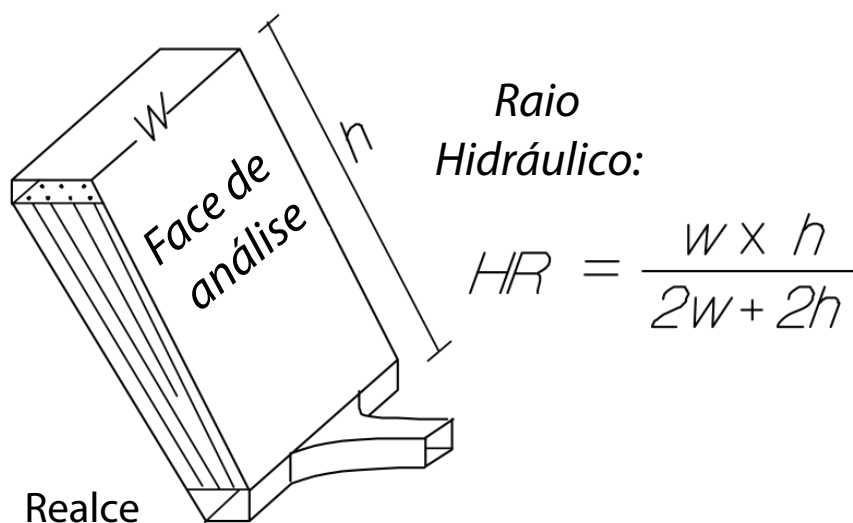
#### 2.3.4.1 Método gráfico

O método gráfico é um método empírico inicialmente demonstrado por Mathews *et al.* (1981 citado por Potvin, 1988), tendo sua aplicação altamente difundida na mineração. Ao longo do tempo este método sofreu adaptações por diferentes autores como Potvin (1988) e Nickson (1992) buscando aumentar sua confiabilidade no dimensionamento de realces estáveis utilizando escoramento com *cabolebolts*. Esse método tem sido referenciado como método de *Potvin*, método de *Mathews/Potvin*, método gráfico de estabilidade modificado ou somente método gráfico de estabilidade.

O primeiro passo para o dimensionamento de realces pelo método empírico é realizar a classificação geomecânica do maciço rochoso. Para isso é necessário um levantamento das estruturas principais do maciço. Posteriormente devem ser considerados os fatores geométricos da escavação, para o cálculo do raio hidráulico (HR) pela Equação 1. A Figura 11 exemplifica o cálculo do raio hidráulico para o *hangingwall*.

$$HR = \frac{\text{Área (m}^2\text{)}}{\text{Perímetro (m)}} = \frac{w \times h}{2 \times (w + h)} \quad \text{Equação 1}$$

Figura 11 – Desenho explicativo sobre o cálculo do raio hidráulico.



Fonte: Modificado de Hutchinson e Diederichs (1996).

A qualidade do maciço rochoso, por sua vez, é definida por meio do cálculo do Q' (sistema Q modificado), conforme mostra a Equação 2 a seguir.

$$Q' = \frac{RQD}{J_n} \times \frac{J_r}{J_a} \quad \text{Equação 2}$$

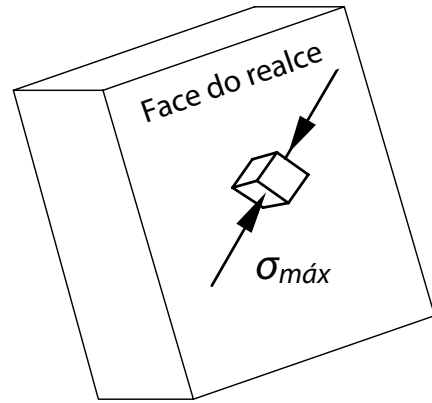
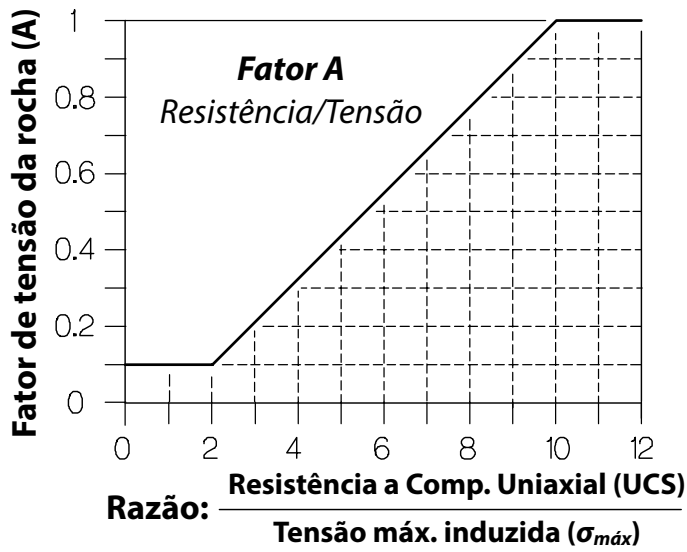
Onde:

- RQD: Índice de qualidade da rocha (%);
- J<sub>n</sub>: Número de famílias de descontinuidade;
- J<sub>r</sub>: Quantifica o grau de rugosidade da descontinuidade;
- J<sub>a</sub>: Quantifica o grau de alteração da descontinuidade;

Após determinar o valor de Q' o próximo passo é determinar o fator A (Figura 12), em função da resistência a compressão uniaxial da rocha e da tensão máxima induzida. Já o fator B (Figura 13) é calculado através do menor ângulo entre a descontinuidade principal e a face do realce. Por fim, o fator C é obtido através do ângulo de mergulho da face e do tipo de ruptura considerada (Figura 14). Esses fatores serão inseridos na equação para o cálculo do N' (número de estabilidade), demonstrado na Equação 3 a seguir.

$$N' = Q' \times A \times B \times C \quad \text{Equação 3}$$

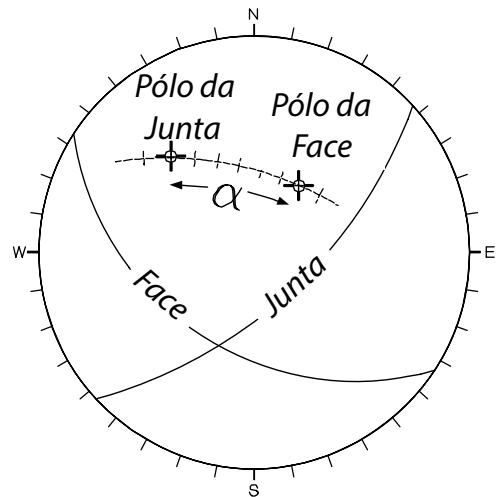
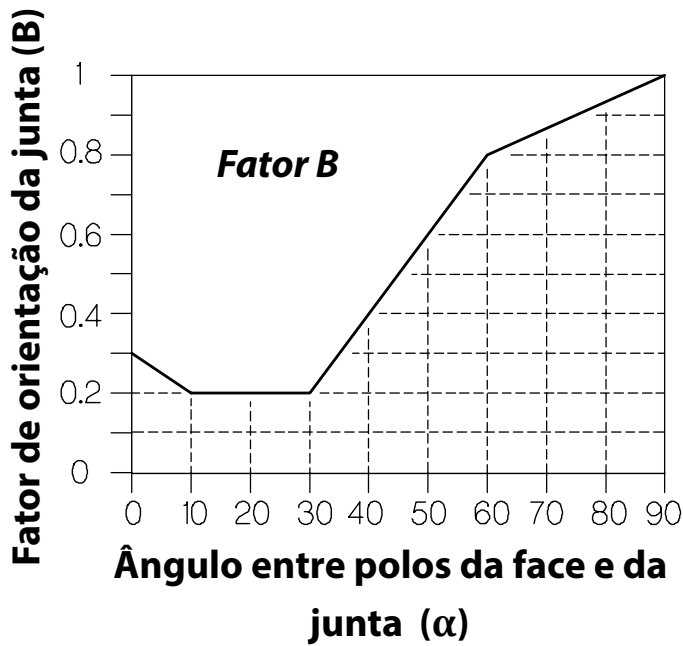
Figura 12 - Ábaco para o cálculo do fator A.



O  $\sigma_{máx}$  pode ser obtido a partir de modelos numérico de tensões 2D ou 3D

Fonte: Modificado de Hutchinson e Diederichs (1996).

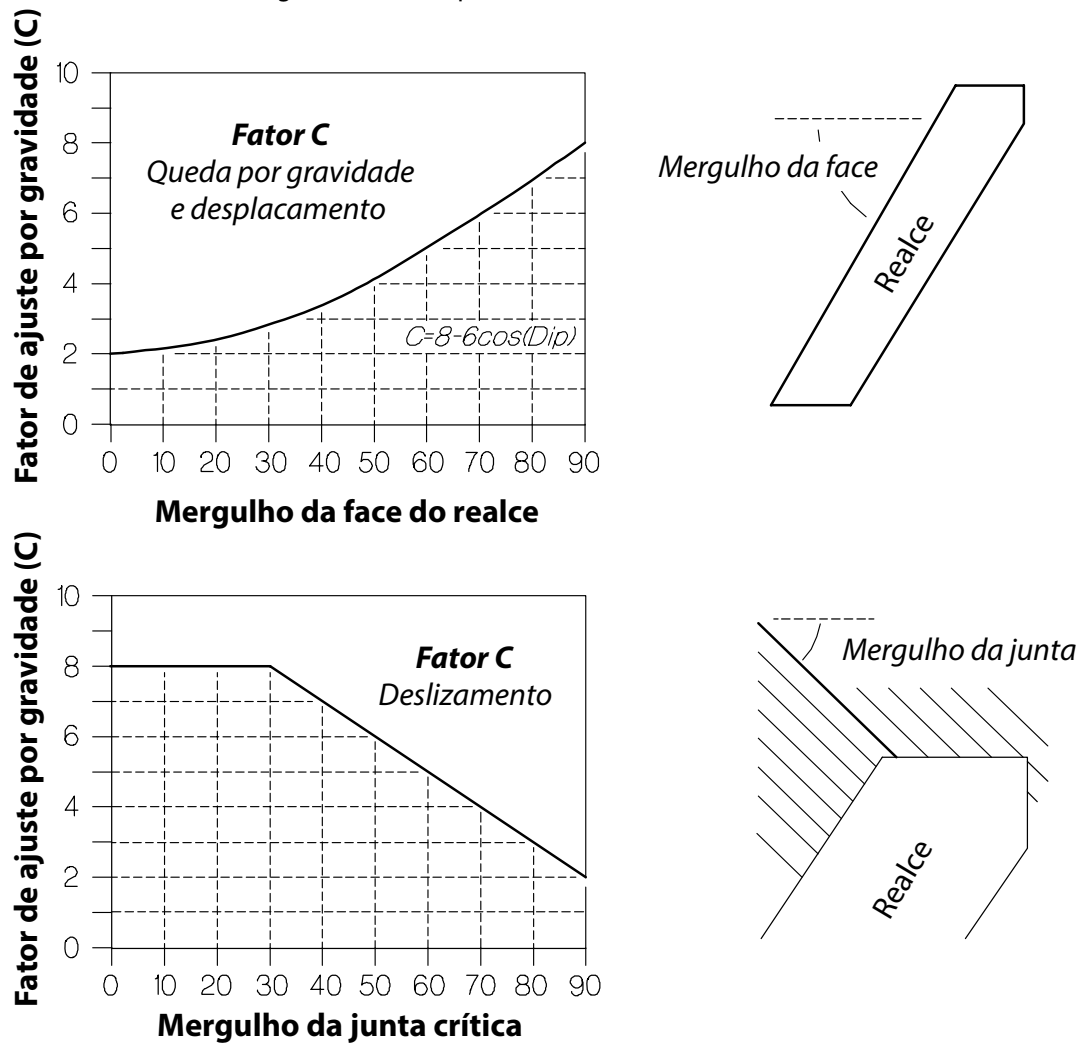
Figura 13 - Ábaco para o cálculo do fator B



Determinação do ângulo verdadeiro entre dois planos: ângulo entre pólos

Fonte: Modificado de Hutchinson e Diederichs (1996).

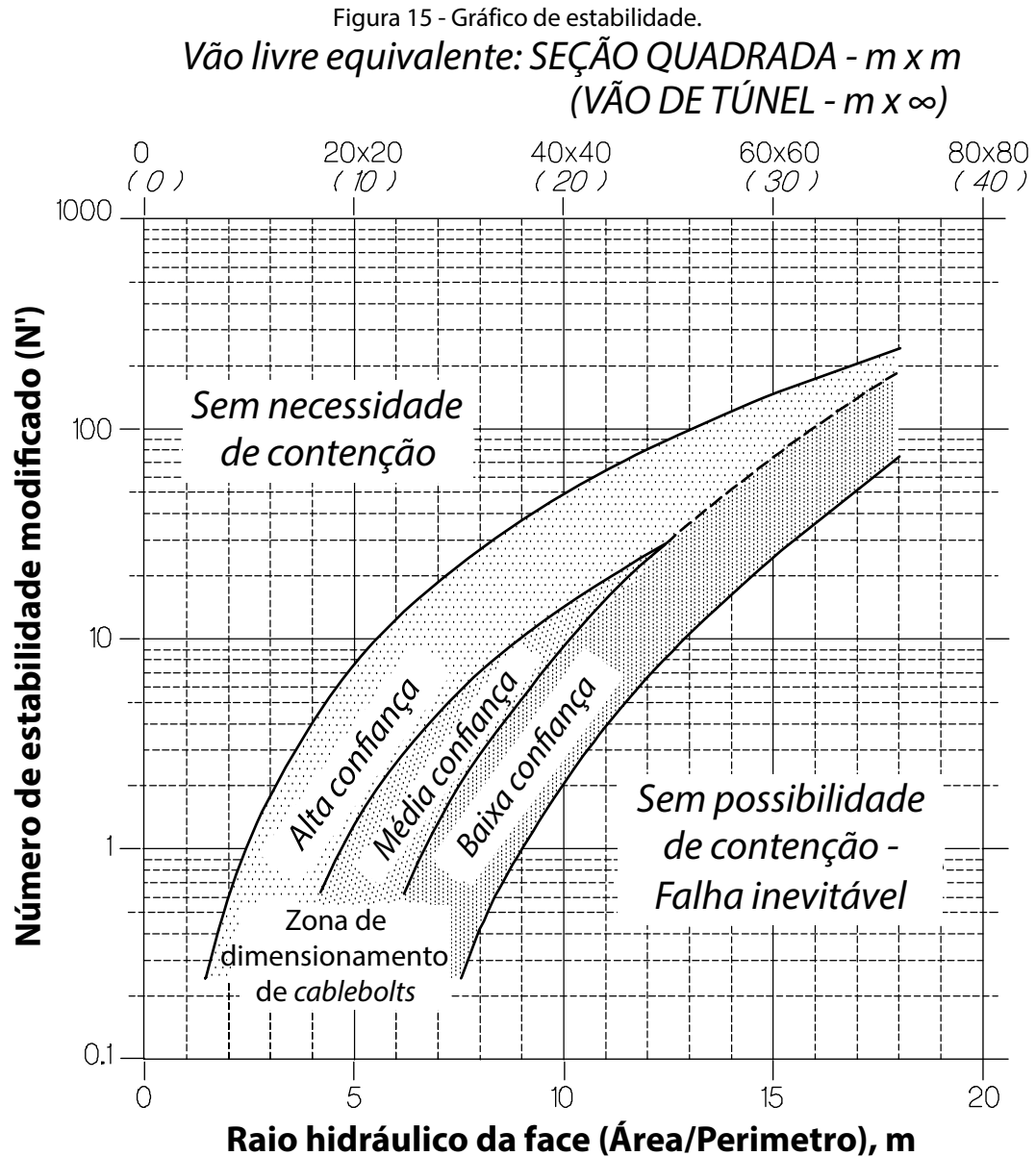
Figura 14 - Ábaco para o cálculo do fator C.



Fonte: Modificado de Hutchinson e Diederichs (1996).

O valor calculado para o número de estabilidade, juntamente com o valor anteriormente calculado para o raio hidráulico, do termo inglês *hydraulic radius* (HR), devem ser plotados no gráfico de estabilidade a seguir (Figura 15).

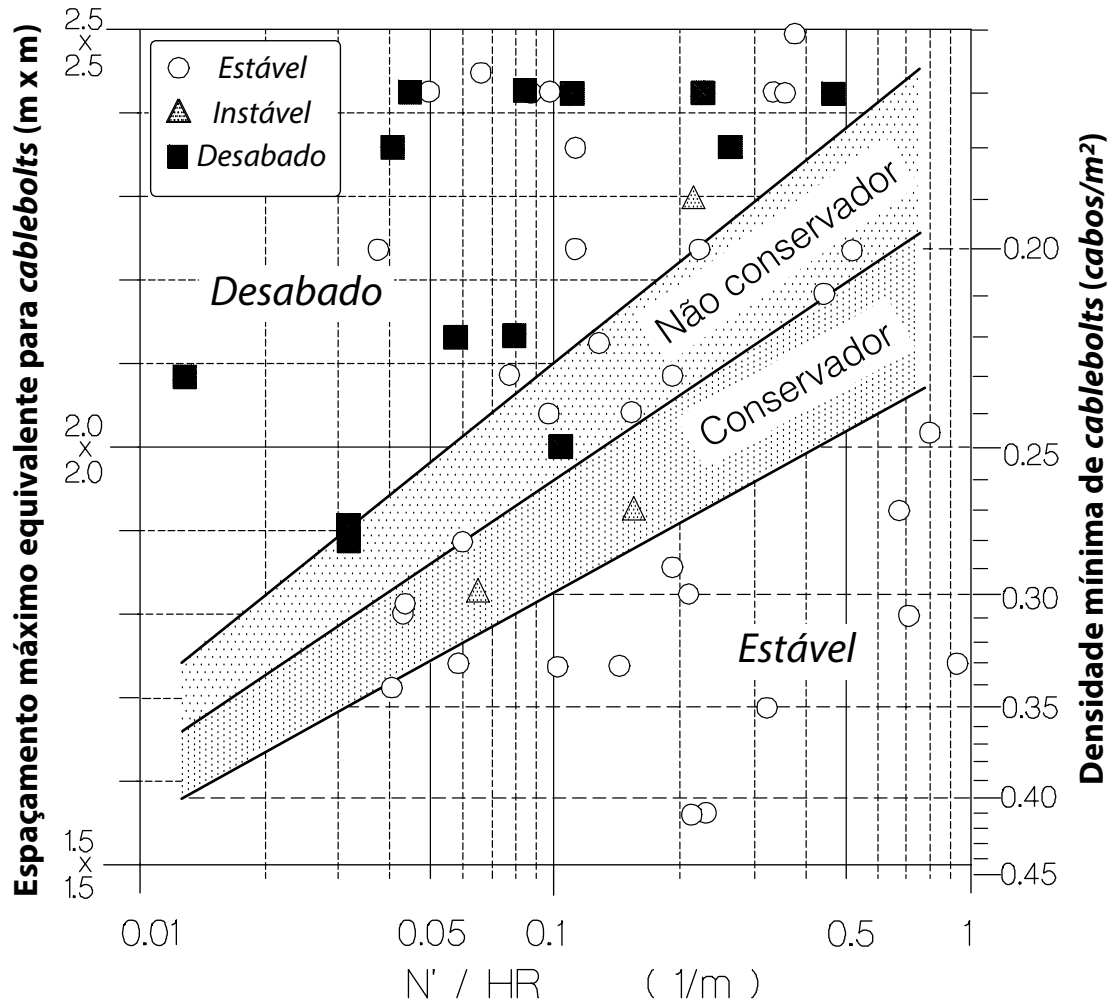




Fonte: Modificado de Hutchinson e Diederichs (1996).

Após a plotagem do ponto no gráfico de estabilidade é possível realizar o dimensionamento dos cabos, caso o ponto esteja dentro da "zona de confiança" para dimensionamento de cabos. Para o cálculo da densidade de cabos utilizando o número de estabilidade e o raio hidráulico é necessário utilizar o ábaco da figura a seguir, conforme Nickson (1992) propôs (Figura 16).

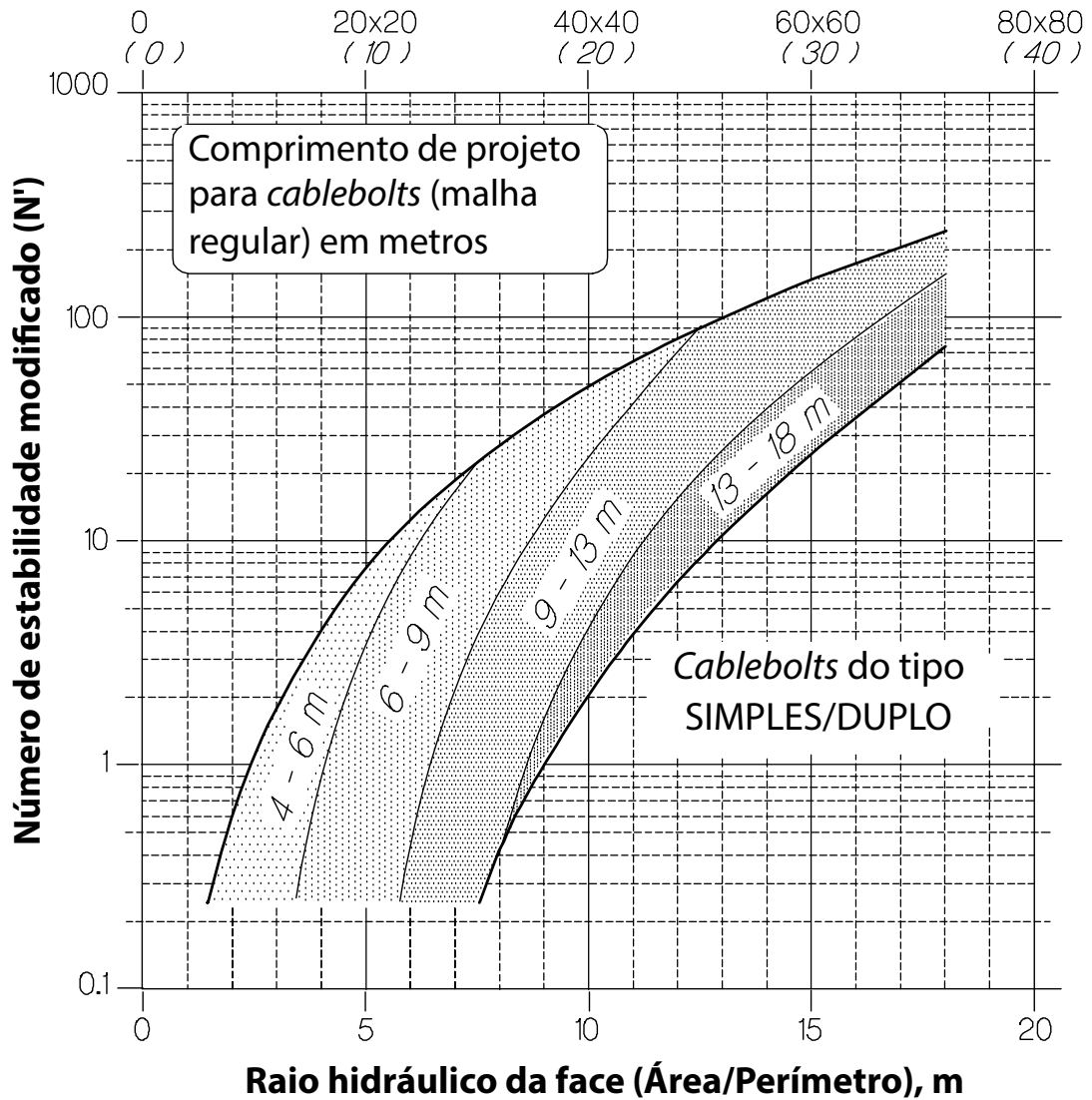
Figura 16 - Ábaco para cálculo da densidade de cabos, segundo Nickson (1992)



Fonte: Modificado de Hutchinson e Diederichs (1996).

Já para o dimensionamento do comprimento de cabos se utiliza o ábaco da Figura 17. Para sua utilização também é necessário plotar o ponto referente ao raio hidráulico calculado e o número de estabilidade.

Figura 17 - Ábaco para cálculo do comprimento de cabos, segundo Nickson (1992)



#### 2.3.4.2 Modelagem numérica

O dimensionamento de realces em minas subterrâneas, normalmente realizado pelo método gráfico, não considera as zonas de baixo confinamento do maciço rochoso ao redor dos realces escavados. Além disso, também não considera o efeito da orientação do realce em relação ao campo de tensões *in situ* (MARTIN *et al*, 1999).

O objetivo principal da modelagem numérica é quantificar o efeito das tensões induzidas na performance dos realces abertos. Isso pode ser alcançado relacionando-se diferentes níveis de tensões induzidas aos diferentes níveis de danos ao maciço rochoso ao redor do vazio produzido. A consideração por trás disso é de que as rupturas causadas por tensões ocorrem ao excederem a resistência local do maciço rochoso, causando *overbreak* no *hangingwall* (VILLAESCUSA, 2014).

Uma metodologia desenvolvida por Henning e Mitri (2007) permite avaliar o potencial de diluição a partir de um modelo numérico tridimensional, onde o tamanho do volume de diluição pode ser representado pela isolinha de  $\sigma_3$ , de valor igual a zero ( $\sigma_3 = 0$  MPa) ou pela isolinha de  $\sigma_3$  igual a resistência a tração do maciço ( $\sigma_3 = \sigma_t$ ). Essa mesma metodologia foi aplicada em Cordova (2016), para a análise da diluição em realces transversais (*transversal stopes*), utilizando modelagem numérica.

Para a análise numérica do desempenho de realces, a escolha do método de modelagem inclui a opção por modelos lineares elásticos, e não lineares contínuos ou descontínuos. Em particular, o uso de modelos descontínuos é específico para uma análise onde os realces apresentam plasticidade significativa e deformações elevadas, com grandes estruturas geológicas incorporadas no modelo (VILLAESCUSA, 2014).

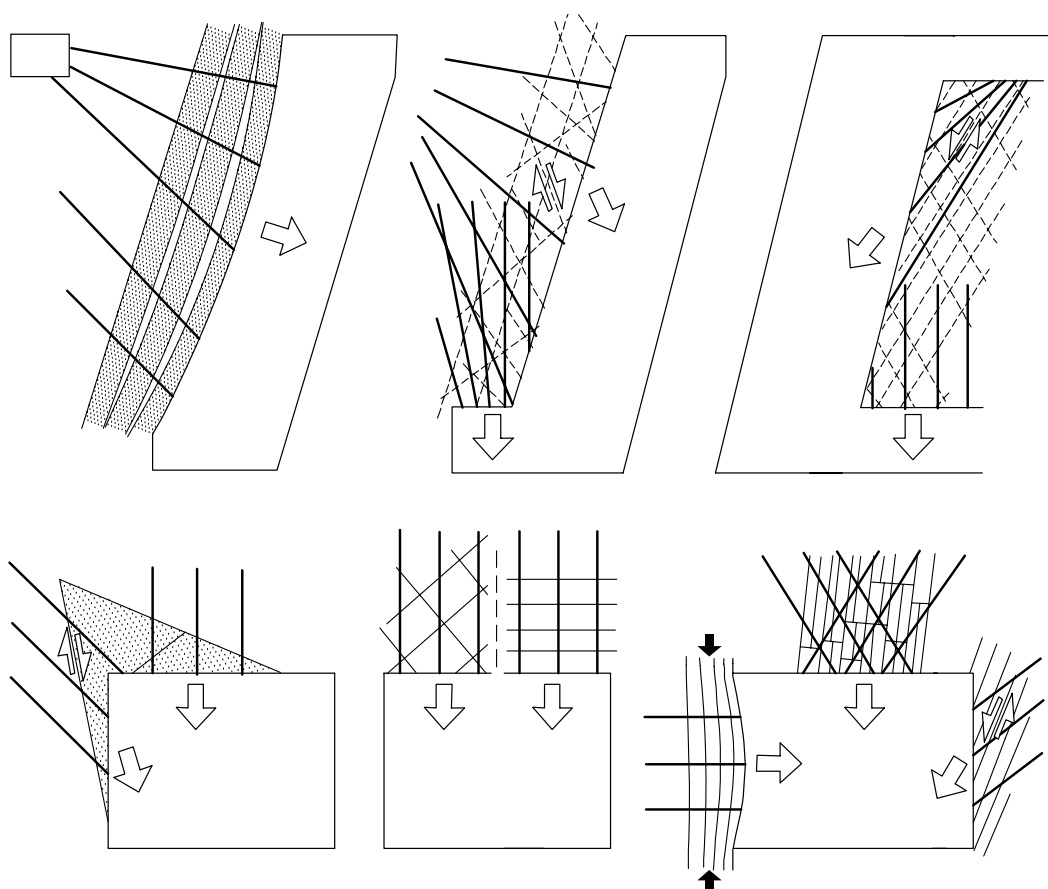
As limitações da modelagem numérica linear elástica incluem a incapacidade de prever movimento, queda de blocos, ou rupturas em falhas ou zonas de cisalhamento. Alternativamente, a modelagem numérica não linear é capaz de prever a ruptura do maciço rochoso e a redistribuição de tensões resultante dessas falhas (BECK; DUPLANCIC, 2005).

### **2.3.5 Suporte do *hangingwall***

O suporte do *hangingwall* utilizando tirantes-cabo (*cablebolts*) tem o papel de prover reforço e sustentação. Como reforço eles devem evitar a abertura e escorregamento de blocos ao longo dos planos de falha do maciço rochoso. Ao garantir efetividade para evitar a abertura das falhas e fraturas, a influência delas poderá ser minimizada na diluição, pois o suporte terá criado um “efeito viga”. Como sustentação, os cabos devem atuar mantendo esta “viga” estável, diminuindo a ocorrência de diluição pelo desprendimentos de blocos provenientes do *hangingwall* (HUTCHINSON; DIEDERICHS, 1996).

Os cabos são ineficientes na retenção de pequenos blocos soltos em maciços rochosos muito fraturados, a não ser que sejam utilizados em conjunto de outros elementos como malhas, concreto projetado ou cintas metálicas, na face do *hangingwall*. A Figura 18 a seguir demonstra cenários com geologia estrutural distinta e a orientação ótima necessária para os cabos, em cada caso.

Figura 18 - Otimização do arranjo dos *cablebolts* para cada cenário de aplicação.



Fonte: HUTCHINSON; DIEDERICHS, 1996.

No método de realces abertos, não é economicamente viável tentar prevenir todas as formas de instabilidade. Em alguns casos nenhum reforço é utilizado no *hangingwall*, pois uma certa quantidade de diluição é geralmente aceitável se dentro dos limites econômicos. Os custos e a diminuição de rentabilidade do projeto causado por essas diluições devem ser comparados contra os custos de escoramento (materiais, tempos e mão de obra) de maneira a determinar se é desejável ou não, utilizar algum tipo de suporte, ou ainda se seria interessante investir no aumento da eficiência do suporte do *hangingwall*.

#### 2.4 Otimizadores econômicos dos limites geométricos de realces

O projeto e o planejamento de realces envolvem a definição do limite geométrico (contorno) ótimo dos realces. Os realces são unidades de minério economicamente viáveis que serão lavradas enquanto os pilares são as porções que não serão lavradas, por motivos geotécnicos e/ou pelo fator econômico. Ao longo dos anos muitos algoritmos foram desenvolvidos com objetivo de resolver o problema da otimização do contorno e volume

de realces subterrâneos. Os algoritmos desenvolvidos podem ser classificados como rigorosos (exatos) ou heurísticos. Estes otimizadores são classificados como rigorosos pelo fato de encontrarem matematicamente soluções ótimas. Já os otimizadores heurísticos são assim chamados por funcionarem em quase todas as situações, porém, obtendo uma solução sem uma rigorosa prova matemática.

O primeiro algoritmo de otimização de realces subterrâneos foi apresentado por Riddle (1977 citado por Ataee-pour, 2000), com o objetivo de otimizar o arranjo dos painéis de lavra no método de lavra *block caving*. No entanto, tem havido um grande aumento no desenvolvimento de algoritmos de otimização de limite de realces subterrâneos desde 1977, com a maioria dos algoritmos desenvolvidos no período entre 1990 e 2010. A Tabela 1 abaixo resume em ordem cronológica os principais algoritmos de otimização e suas diferenças.

Tabela 1 - Evolução dos principais algoritmos otimizadores de limites de realces subterrâneos.

Tipo	Autor	Algoritmo	Método de Lavra	Dimensão	Estocástico
Heurístico	Cheimanoff <i>et al.</i> (1989)	<i>Octree Division Approach</i>	Todos	3D	Não
	Alford (1995)	<i>Floating Slope</i>	Todos	3D	Não
	Ataee-pour (2000)	<i>Maximum value neighbourhood</i>	Todos	3D	Não
	Cawrse (2001)	<i>Multiple Pass Floating Slope</i>	Todos	3D	Não
	Manchuk (2006)	<i>Simulated Annealing</i>	Todos	3D	Não
	Topal e Sens (2010)	<i>Slope Shape annealing</i>	Não informado	3D	Não
	Sandanayake <i>et al.</i> (2015)	<i>Iterative Optimization</i>	<i>Sublevel Stopping</i>	3D	Não
Rigoroso	Riddle (1977)	<i>Dynamic Programming</i>	<i>Block caving</i>	2D	Não
	Deraisme <i>et al.</i> (1984)	<i>Downstream Geostatistical Approach</i>	<i>Cut and fill; Sublevel Stopping</i>	2D	Sim
	Ovanic e Young (1995)	<i>Branch &amp; Bound</i>	Todos	1D	Não
	Ovanic e Young (1999)	<i>Mixed Integer Programming</i>	Todos	1D	Não
	Jalali e Ataee-pour (2004)	<i>Probable Slope</i>	Todos	2D	Não
	Grieco e Dimitrakopoulos (2007)	<i>Probabilistic MIP</i>	<i>Longhole stoping with backfill</i>	-	Sim
	Jalali <i>et al.</i> (2007)	<i>Optimum Limit Integrated Probable Slope</i>	Todos	2D	Não
	Bai (2013)	<i>Network Flow Method</i>	<i>Sublevel stoping</i>	3D	Sim
	Jalali <i>et al.</i> (2016)	<i>Global Optimization for Underground Mining Area</i>	Todos	2D	Não

Fonte: Adaptado de Appianing e Ben-Awuah (2018).

### 2.4.1 Algoritmos exatos

Algoritmos exatos ou rigorosos podem ser descritos como os algoritmos baseados em um modelo matemático e, portanto, garantem uma solução ótima. Tais algoritmos são robustos, possuem objetividade, são tratáveis, fornecem soluções aos modelos e facilitam a análise de sensibilidade de variação de parâmetros. Além disso, as soluções ideais fornecidas por algoritmos rigorosos refinados podem frequentemente ser alcançadas mais rapidamente com métodos meta-heurísticos básicos (NHLEKO *et al.*, 2018). Abaixo são discutidos em sequência cronológica os otimizadores exatos para geometria de realces.

#### 2.4.1.1 *Dynamic Programming*

Inicialmente, o algoritmo baseado na programação dinâmica foi utilizado por Riddle (1977 citado por Ataee-pour, 2000) com o objetivo de otimizar o arranjo dos painéis de lavra nas situações do método de lavra *block caving*. Este método foi estabelecido devido a modificação de Johnson e Sharp (1971) e este é na verdade utilizado na determinação do contorno de cavas, sendo uma aplicação sistemática do algoritmo bidimensional que primeiramente é aplicado a todas as seções transversais e, no final, a mesma rotina é realizada para as seções ortogonais das seções originais.

A programação dinâmica de acordo com Riddle (1977 citado por Ataee-pour, 2000) também é uma abordagem multiseção para solucionar situações tridimensionais. Isto significa que, para este caso, a solução busca resultados “ótimos” para situações bidimensionais, mas podendo eventualmente não ser a solução “ótima” para a situação tridimensional dos painéis subterrâneos. Outro aspecto importante é que este método está limitado a situações de lavra em *block caving*, não sendo adequado para outros métodos de lavra subterrânea.

#### 2.4.1.2 *Downstream Geostatistical Approach*

Esse algoritmo também conhecido como abordagem de morfologia matemática, foi proposto por Deraysme *et al.* (1984 citado por Bai *et al.*, 2013). A introdução desta abordagem foi determinar o contorno da forma do corpo de minério lavrável em minas subterrâneas. Esta abordagem foi construída para métodos corte e enchimento (*cut and fill*) e realces em subníveis, e foi baseada em modelos de blocos em seções 2D do depósito.

O algoritmo se beneficia de um suporte matemático e espera-se que forneça a solução ótima real, pelo menos em 2D (ERDOGAN *et al.*, 2017). No entanto, os cenários economicamente otimizados não respeitam necessariamente as restrições geométricas. Outras modificações são necessárias para fazê-los atender às restrições e isso torna os resultados não ótimos (ATAEE-POUR, 2005). A abordagem leva em consideração a incerteza dos teores no processo de otimização. O método controla apenas a geometria do realce, portanto, não considera o lucro econômico relacionado. Isso limita a otimização dos resultados em 3D (ERDOGAN *et al.*, 2017).

#### 2.4.1.3 *Branch & Bound*

Ovanic e Young (1995 citado por Ataee-pour, 2000) utilizaram a técnica *Branch & Bound* (B&B) para a otimização dos limites econômicos de realces. Um limite economicamente ótimo foi desenvolvido pela otimização da área a ser minerada através da soma cumulativa dos valores econômicos dos blocos (que deverão ser inseridos dentro dos limites dos realces) e pela avaliação capaz de identificar os blocos que deverão ser excluídos. Com isso, os resultados destas duas avaliações definem o contorno ótimo para a área estudada.

As coordenadas horizontais dos blocos, ao longo dos painéis, são incrementadas e podem ser consideradas como um valor de referência para a função não linear. Os valores econômicos dos blocos são considerados como atributos e são acumulados para a definição dos valores finais. Desde o início do painel até o final dele, poderão ter pontos discretizando a linha de avaliação e este parâmetro será definido pelo tamanho do bloco.

No algoritmo B&B não há restrição quanto aos blocos que formarão a envoltória, podendo ser incluídos blocos inteiros ou porções de blocos dentro da geometria, ou seja, os blocos não estão fixados a uma geometria uniforme, facilitando nas situações em que as interpretações geológicas são complexas, podendo se adequar a variabilidade e a descontinuidade dos sólidos geológicos.

#### 2.4.1.4 *Probable Stope*

Jalali e Ataee-pour (2004 citado por Jalali *et al.*, 2007) estabeleceram esta técnica baseados na programação dinâmica desenvolvida por Riddle (1977 citado por Ataee-pour, 2000) para a otimização dos limites dos realces em métodos que são aplicáveis estritamente



nos depósitos em veios. A principal diferença desta aplicação em relação as outras é que as restrições de geometria estão incluídas no modelo, com isso, apenas a função objetivo é necessária, facilitando a sua programação e exigindo do algoritmo um menor tempo computacional.

#### 2.4.1.5 *Mixed Integer Programming*

Ovanic e Young (1999 citado por Ataee-pour, 2000) introduziram um modelo *Mixed Integer Programming* (MIP) para otimizar os limites dos realces. O modelo funciona localizando os pontos iniciais e finais ideais para lavra dentro de uma mesma linha (painel de lavra) em um modelo de blocos, estabelecendo os limites de contorno ideais. A fim de determinar a localização ideal inicial e final de cada painel, duas funções lineares cumulativas são calculadas. A principal vantagem dessa técnica em comparação com as outras é que a geometria do bloco não precisa ser regular ou ortogonal. Uma desvantagem, entretanto, é que o algoritmo otimiza o limite dos realces ao longo da linha de blocos em apenas 1D. Portanto, não existem exemplos e resultados para o uso do algoritmo em 3D. Outra desvantagem é que o algoritmo também só permite que o usuário otimize o planejamento de lavra das linhas, e como as localizações dessas linhas são determinadas anteriormente no processo de design, o algoritmo otimiza apenas parcialmente o design da mina (TOPAL; SENS, 2010).

#### 2.4.1.6 *Probabilistic MIP*

Um algoritmo probabilístico baseado em MIP foi desenvolvido por Grieco e Dimitrakopoulos (2007 citado por Dimitrakopoulos e Grieco, 2009) e aplicado no estudo de caso da mina Kidd Creek, Canadá. Nessa abordagem, o corpo de minério é inicialmente dividido em camadas. Cada camada é então subdividida em vários painéis e cada painel é subdividido em uma série de anéis (*rings*). Cada anel é atribuído a uma variável binária do modelo inteiro misto. A função objetivo do algoritmo maximiza o conteúdo de metal. Os anéis de lavra mínimo e máximo e os tamanhos dos pilares que devem ser deixados entre dois realces primários são limitados pelas restrições do modelo.

Embora o método proposto seja a primeira metodologia de projeto de realces que considera a incerteza geológica, ele tem desvantagens potenciais. Como a metodologia é baseada em anéis, que foram predefinidos em termos de localização e tamanho para

determinar os realces mais lucrativos, ela encontra o arranjo de realces ideais com base nesses anéis. Esta abordagem não permite o exame preciso do corpo de minério em áreas menores e o exame de realces em locais diferentes. Além disso, como cada anel é representado como uma variável binária dentro do modelo MIP, isso incentiva um tempo de solução muito longo à medida que o número de anéis aumenta no modelo (TOPAL; SENS, 2010; SANDANAYAKE, 2014).

#### 2.4.1.7 *Optimum Limit Integrated Probable Stope*

*Optimum Limit Integrated Probable Stope* (OLIPS) é um algoritmo desenvolvido em 2007 com base no método de Programação Dinâmica de Jalali *et al* (2007). O algoritmo satisfaz todas as restrições técnicas e geométricas e fornece prova matemática da otimização. O algoritmo tem duas etapas principais, na primeira etapa um modelo econômico convencional de painel de mineração é construído e, na segunda etapa, o modelo econômico de realces prováveis são derivados de um modelo convencional. Com base no algoritmo OLIPS, um programa de computador denominado *Stope Boundary Optimizer* (SBO) foi desenvolvido e validado por modelos hipotéticos 2D (SOTOUDEH *et al.*, 2017).

#### 2.4.1.8 *Network Flow Method*

O método *Network Flow* é um algoritmo de otimização que foi introduzido por Bai (2013) para otimizar o projeto de realces com base na teoria gráfica e especificamente aplicável para o método de realces em subníveis. O algoritmo é baseado em um sistema de coordenadas cilíndricas, que é definido em torno de um poço vertical inicialmente especificado (BAI, 2013; SANDANAYAKE, 2014). No processo, o gráfico é construído usando arcos verticais com as restrições de inclinação da capa e da lapa, com arcos horizontais para a restrição de largura dos realces. A seleção de um bloco de mineração específico para um realce deve ter como objetivo maximizar o valor do realce sujeito a duas restrições adicionais: a distância máxima de um bloco em relação ao poço vertical e a largura horizontal necessária para trazer o bloco de mineração mais distante até o poço. O retorno econômico do realce é otimizado em função da localização e altura do poço e, conseqüentemente, são identificados o melhor local e altura para outros poços. Finalmente, o desempenho do algoritmo é comparado com o algoritmo *floating stope* e obtendo

melhores valores de solução. No entanto, o algoritmo é limitado a depósitos subverticais relativamente pequenos extraídos pelo método de realces em subníveis (SANDANAYAKE, 2014; ERDOGAN *et al.*, 2017).

#### **2.4.2 Algoritmos heurísticos**

Porumbel (2012) explica que uma abordagem prática para resolver muitos problemas de grande tamanho é o uso de algoritmos de busca heurística ou simplesmente heurística. Eles geralmente são usados para explorar apenas uma pequena parte de um vasto espaço de busca. Os algoritmos heurísticos utilizam recursos razoáveis e são capazes de produzir soluções aceitáveis, mas sem garantia matemática de que apresentam a melhor solução global. Ainda sim, eles têm a capacidade de produzir resultados competitivos em qualquer problema computacional, para o qual algoritmos exatos ou rigorosos exigem um custo computacional proibitivo. Abaixo são discutidos em sequência cronológica os otimizadores heurísticos para a geometria de realces.

##### *2.4.2.1 Octree Division Approach*

Cheimanoff *et al.* (1989 citado por Ataee-pour, 2000), desenvolveu o algoritmo de divisão espacial *octree*. Uma *octree* é uma estrutura de dados em árvore na qual cada nó interno tem exatamente oito filhos. Os *octrees* são usados com mais frequência para particionar um espaço tridimensional, subdividindo-o recursivamente em oito octantes. O algoritmo utiliza essa metodologia para remover blocos de mineração indesejados com base no tamanho mínimo. O algoritmo é projetado em três etapas, a primeira etapa é a utilização de contornos interpretados, fornecidos por geólogos e engenheiros de minas. Esses dados são usados para construir um modelo geométrico dos recursos.

A segunda etapa é transformar os recursos geológicos modelados em reservas lavráveis. Nesta etapa, diferentes fatores de produção são simulados para construir possíveis volumes utilizáveis, utilizando os recursos geológicos disponíveis.

Na etapa final, o algoritmo, conseqüentemente, divide as reservas em subvolumes para futuras avaliações econômicas. Esses subvolumes são armazenados ou removidos do modelo, caso não cumpram com as restrições geométricas ou econômicas (TOPAL; SENS, 2010).

Alford (1995) observa que o algoritmo de divisão gera uma solução 3D viável para a geometria de realces, mas leva à adição de mais resíduos ao arranjo final dos realces, pois o algoritmo não analisa os subvolumes em conjunto. Portanto, o algoritmo não garante a otimização global dos arranjos de realces, resultando em uma solução heurística.

#### 2.4.2.2 *Floating Stope*

Este algoritmo foi desenvolvido por Alford (1995) como uma ferramenta de otimização para definir o limite ideal para minério lavrável ou envelope de realces que podem ser economicamente extraídos por lavra subterrânea e a partir de um modelo de blocos econômicos (SHAHRIAR *et al.*, 2007). As suas características são semelhantes ao modelo heurístico de cones flutuantes, muito conhecido em otimização de lavra a céu aberto.

A metodologia é implementada em um modelo de blocos regulares definido por uma envoltória geológica. Teores de cortes são pré-definidos de modo a discriminar o limite de estéril e minério. As principais restrições são as geometrias dos realces, que são definidos pelo transladar de uma unidade mínima de lavra, sendo o maior problema a definição de quais blocos deverão ser incluídos nesta etapa de flutuação da unidade mínima de lavra, de modo que o realce obtenha um valor econômico aceitável. Os diferentes objetivos de otimização oferecidos pelo algoritmo de *stope* flutuante incluem a maximização das toneladas de minério, o teor e o retorno econômico.

Os métodos *Floating Stope* e cones flutuantes possuem diferentes restrições geométricas, no entanto, outra discrepância é o fato de que no algoritmo de cones flutuantes a capacidade de avaliar os cones é feita separadamente, não identificando situações em que ambos os cones possam ser lavrados juntos para ter-se um resultado economicamente viável. Já para o método *Floating Stope*, a limitação é oposta, pois quando a sua avaliação individual é feita, poderá identificar realces economicamente viáveis, mas quando unidos resultará em um *stope* único inviável.

#### 2.4.2.3 *Maximum value neighbourhood*

Ataee-Pour (2000) desenvolveu o algoritmo *Maximum Value Neighbourhood* (MVN) (MVN) baseado no princípio do algoritmo *Floating Stope*. Este algoritmo define os blocos de lavra que devem ser lavrados juntos, como blocos vizinhos para satisfazer a restrição de

tamanho mínimo de realces. Esses blocos são limitados pelos requisitos geotécnicos e de lavra. Para cada bloco no modelo de blocos é atribuído um valor econômico. O tamanho mínimo do realce é obtido identificando o conjunto de blocos vizinhos viáveis para cada bloco do modelo, que fornece o máximo valor líquido. Durante o processo de otimização, são feitas verificações para excluir blocos desnecessários, como aqueles com valores econômicos negativos, já sinalizados, ou aqueles em que o MVN do bloco é negativo (LITTLE *et al*, 2013; SANDANAYAKE, 2014).

Este algoritmo é baseado em uma premissa simples o que facilita a implementação computacional. No entanto, sua principal desvantagem é gerar diferentes soluções dependendo do ponto de partida, portanto, ele falha em garantir uma solução ótima global.

#### 2.4.2.4 *Multiple Pass Floating Stope*

O *Multiple Pass Floating Stope Process* (MPFSP) foi desenvolvido por Cawrse (2001). O algoritmo depende dos mesmos princípios do algoritmo de *stope* flutuante, sendo dividido em três etapas: definição dos parâmetros de entrada, execução e geração de arquivos. Os parâmetros de entrada são vários, definidos pelo usuário, como o teor de alimentação da planta, teor de corte e estéril máximo. Durante a etapa de execução, envelopes de *stopes* econômicos são criados para cada conjunto de parâmetros. Por fim, os arquivos de dados ou os arquivos estatísticos gerados no processo são convertidos em um formato de tabela CSV (*Comma-separated values*).

Os envelopes gerados, com base nos parâmetros de entrada, podem fornecer informações extras durante o planejamento da mina, ajudando os engenheiros a melhorar a eficiência do projeto. No entanto, ele não elimina as desvantagens do algoritmo *Floating Stope*. Portanto, este processo não pode garantir uma solução ótima global para o arranjo de realces (TOPAL; SENS, 2010; SANDANAYAKE, 2015).

#### 2.4.2.5 *Simulated Annealing*

O algoritmo *Simulated Annealing* (SA) foi desenvolvido por Manchuk (2006) e modificado por Manchuk e Deutsch (2008 citado por Bai, 2013) para geometria de realces e otimização de sequenciamento de lavra. Nesse método, um realce é parametrizado como um objeto geométrico constituído por um conjunto de vértices e arestas que formam uma malha triangularizada. Isso facilitou notavelmente a manipulação de restrições geométricas

de realces na otimização. A lógica por trás da otimização é ajustar aleatoriamente a forma, respeitando as restrições geométricas, a fim de encontrar o volume envolvendo o maior retorno econômico. O algoritmo oferece uma solução 3D geral, integrando restrições completas de geometria, independentemente dos métodos de lavra selecionados.

Uma das limitações do algoritmo é que ele pode ser excessivamente lento, e em algumas situações o tempo de convergência para a otimização pode ser infinito. Isso ocorre especialmente quando o número de vértices é grande e gera uma geometria complexa. Portanto, na prática o algoritmo SA é considerado heurístico, pois seu resultado com a aproximação da solução ótima real é difícil de avaliar (BAI, 2013).

#### 2.4.2.6 *Stope Shape Annealing*

Topal e Sens (2010) desenvolveram um algoritmo heurístico para otimização do limite de realces. O algoritmo regulariza o modelo de blocos fornecido, conseqüentemente os realces gerados são restringidos pela altura, comprimento e largura, por meio do modelo de bloco regularizado. A partir daí, o software *Matlab* é usado para implementar o algoritmo com base nos valores econômicos dos realces. Os resultados são então visualizados no software *MineSight*<sup>®</sup>, com base nos parâmetros pré-definidos (ERDOGAN *et al.*, 2017; SANDANAYAKE, 2014).

Este algoritmo permite que arranjos de realces sejam gerados de acordo com diferentes estratégias de projeto, por exemplo, utilizando um tamanho fixo de realces, ou então utilizando uma gama de tamanhos variáveis. O algoritmo permite encontrar um tamanho ideal de realces e gerar o arranjo de realces com base a este tamanho, e ainda pode gerar desenhos de realces começando com aquelas de maior lucro por metro cúbico.

A principal vantagem deste algoritmo é que ele pode gerar soluções de arranjos de realces em 3D. Por outro lado, os realces incluídos no arranjo final são escolhidos em ordem decrescente, excluindo aqueles sobrepostos. Portanto, ele desconsidera a probabilidade de múltiplas combinações, que podem ser derivadas de um determinado conjunto de realces. Algumas dessas combinações múltiplas podem render um arranjo com maior valor econômico total (TOPAL; SENS, 2010).

#### 2.4.2.7 Iterative Optimization

Sandanayake *et al.* (2015) propôs e desenvolveu um novo algoritmo heurístico 3D para a otimização do arranjo de realces. Ele encontra uma solução única que maximiza o valor econômico sob restrições físicas e geotécnicas. O algoritmo sugerido recomenda uma solução única para o arranjo de realces e gera realces não sobrepostos. Além disso, inclui tamanhos variáveis de realces com ou sem pilares e atende às restrições de mineração e geotécnicas. No entanto, para encontrar a solução ideal, o algoritmo precisa avaliar todas as combinações únicas possíveis, fato que requer um recurso computacional significativo em aplicações de grande escala. Isso limita o algoritmo para encontrar as soluções ideais em grandes conjuntos de dados (ERDOGAN *et al.*, 2017).

Segundo Sandanayake *et al.* (2015) o estudo de validação utilizado para testar o algoritmo proposto gera soluções mais lucrativas do que o algoritmo MVN disponível comercialmente para modelos de blocos reais. Embora o método proposto produza realces únicos e não sobrepostos, ele está limitado aos tamanhos de realce predefinidos. Além disso, o algoritmo ignora as variações de formato dos realces e não considera a utilização de pilares de rocha entre realces consecutivos para melhorar a estabilidade nas zonas de produção.

#### 2.4.3 Implementações em pacotes de software comerciais

Um dos algoritmos mais utilizados é o algoritmo heurístico *Floating Stope* e sua extensão *Multiple Pass Floating Stope Process* (MPFSP), sendo sua primeira versão implementada em software comercial no ano de 2011 e que pode ser encontrado nos pacotes de software comerciais Datamine®, Deswik® e Vulcan® (ALFORD MINING SYSTEMS, 2016). O algoritmo *Maximum Value Neighbourhood* (MVN) está implementado no software MineSight® (atualmente conhecido como MinePlan®) (ERDOGAN *et al.*, 2017). O software Micromine® permite usar programação matemática para determinar o arranjo ideal de realces. Porém, caso definido pelo usuário, o software também pode buscar uma solução heurística. De acordo com a combinação dos parâmetros de projeto especificados, o algoritmo heurístico pode ser capaz de encontrar apenas uma solução aproximada - mas deve exigir menos tempo do que o algoritmo matemático para isso (MICROMINE, 2021).

No trabalho de Erdogan *et al.* (2017), as capacidades e limitações de quatro algoritmos heurísticos são avaliados e comparados entre si e, em seguida, comparados com

os resultados de otimização de realces em um estudo de caso real de mina subterrânea. Os algoritmos e os pacotes de software comercial selecionados foram *Floating Stope* (Datamine®), *Maximum Value Neighbourhood* (Minesight®) e dois aplicativos especiais que foram desenvolvidos por Topal e Sens (2010) e Sandanayake *et al* (2015). *Mineable Shape Optimizer* (MSO®), uma ferramenta de otimização introduzida pelo software Datamine®, também foi incluída no estudo para comparação com outros algoritmos. Para o estudo de comparação, foi utilizado uma seção do modelo de blocos que representa um depósito de ouro localizado no sul da Austrália, com profundidade estendendo-se de 150 m até 600 m abaixo da superfície.

A pesquisa teve como objetivo encontrar soluções ótimas e maximizar o valor da operação com base nas restrições e regras definidas. Os resultados refletiram que, em termos de valores de lucro líquido, a ferramenta MSO produziu o mais alto, seguida pelo algoritmo de Sandanayake *et al* (2015), algoritmo MVN, o caso real da mina, algoritmo Topal e Sens (2010) e por fim a de *Floating Stope*, respectivamente. No entanto, nenhum deles pode fornecer uma solução realmente ótima, apenas soluções aproximadas em 3D.

## 2.5 Modelagem numérica

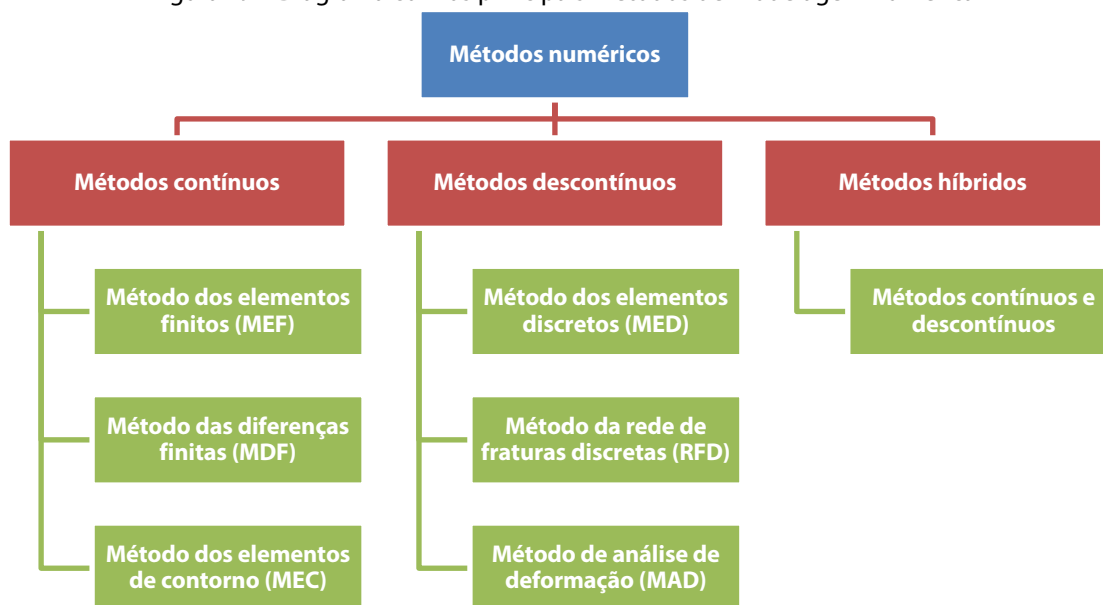
Conforme salientado anteriormente, várias das etapas de projeto de mina, além do projeto de detalhe da geometria dos realces de produção, necessitam de algum tipo de análise de tensões. Nas práticas atuais de mineração são empregados métodos numéricos computacionais, para análises de tensões e deformações, para verificar o comportamento e estabilidade ao redor das escavações (VILLAESCUSA, 2014).

Os métodos numéricos são amplamente utilizados como auxiliares na tomada de decisão para os arranjos globais dos realces e para a sequência de extração deles. Além disso, são empregados no projeto de detalhe e no dimensionamento de componentes de infraestrutura global da mina, como os acessos, poços de ventilação e pilares (JING, 2003).

Jing e Hudson (2002), Jing (2003) apresentam os diferentes métodos numéricos e suas aplicações em cada caso na mecânica das rochas. O diagrama da Figura 19 abaixo mostra a classificação de cada um dos principais métodos de modelagem numérica.



Figura 19 – Diagrama com os principais métodos de modelagem numérica



Fonte: Cordova (2016).

Os métodos de modelos híbridos são uma combinação dos métodos mencionados acima, de maneira a otimizar os resultados e o consumo computacional. Diferentes combinações de modelos são mencionadas na literatura, tais como MEF/MEC, MED/MEF além do MED/MEC (JING, 2003).

### 2.5.1 Métodos contínuos

Nos métodos contínuos o domínio do problema é geralmente subdividido em um número finito de subdomínios (elementos) cujo comportamento é aproximado por descrições matemáticas. Esses subdomínios devem satisfazer tanto as equações governantes do problema quanto a condição de continuidade em suas interfaces com elementos adjacentes, já que *continuum* implica que em todos os pontos de domínio os materiais não podem ser desmembrados em pedaços menores. Os principais métodos contínuos serão revisados nos tópicos a seguir.

#### 2.5.1.1 Método dos elementos finitos

No método dos elementos finitos (MEF) todo domínio do problema é dividido em pequenas regiões contendo elementos não sobrepostos, que estão conectados uns aos outros através de pontos, denominados nós. Como a divisão do domínio pode ser qualquer, este método apresenta grande vantagem no tratamento de casos com geometria complexa (VELLOSO *et al.*, 1998).

O comportamento de cada elemento precisa ser descrito satisfazendo as condições de equilíbrio, propriedades constitutivas do material e as condições de contorno para que estes possam ser então unidos pelos nós. Esse é um dos métodos mais populares para modelagem de problemas de engenharia, mesmo necessitando de amplo poder computacional, pois muitas equações precisam ser armazenadas para então serem resolvidas e possibilitar a obtenção das soluções.

#### 2.5.1.2 *Método das diferenças finitas*

No método das diferenças finitas (MDF) o sólido é dividido em uma malha formada por elementos quadrados, sendo a estratégia de solução diferente do MEF. O método consiste na substituição da equação diferencial, que governa o fenômeno em estudo, por uma equação algébrica. Assim, na solução de um problema pelo MDF é necessário se traçar uma malha ortogonal, sendo a solução obtida nos pontos de intersecção da malha (JING, 2003).

Nesta técnica não é formulada nenhuma matriz, e o processo é explícito no equilíbrio de forças atuantes no ponto de intersecção do material, resultando numa aceleração da massa associada com o ponto.

Aplicando a segunda lei do movimento de Newton, a equação diferencial produz incrementos de deslocamentos, e aplicando as relações constitutivas apropriadas, se obtêm as novas forças, e assim sucessivamente, para cada ponto de integração no modelo.

Diferente do MEF, nenhum processo iterativo é adotado (ITASCA, 2005). Os requerimentos de memória computacional são baixos, pois não são necessários gravar grandes matrizes, mas o tempo para resolução pode ser maior devido aos intervalos de tempo entre cada etapa para garantir a convergência numérica. Em alguns casos a convergência numérica pode não ser alcançada, normalmente devido algum erro na modelagem.

#### 2.5.1.3 *Método dos elementos de contorno*

No método dos elementos de contorno (MEC) a discretização é realizada somente nas bordas do modelo. O método é apropriado para problemas que possuem uma baixa relação entre superfície e volume. Todo o conceito do método pode ser descrito da seguinte forma: para a escavação se considera uma série de trações negativas aplicadas na borda da

escavação, ou seja, para cada elemento uma força de tração fictícia é aplicada e se assume que esta é igual, mas em sentido contrário, a tensão *in situ* anterior a escavação (JING, 2003).

Um processo matemático iterativo é adotado para coincidir a força fictícia de maneira que as componentes externas de tensão de cisalhamento e de tensão normal sejam iguais as tensões internas de cisalhamento e de tração negativas.

## **2.5.2 Métodos descontínuos**

Nos método descontínuos, o maciço rochoso é representado como um conjunto de blocos discretos. As juntas e descontinuidades são vistas como interfaces entre os corpos distintos. A seguir alguns dos principais métodos descontínuos de modelagem são citados.

### *2.5.2.1 Método dos elementos discretos*

No método dos elementos discretos (MED) a malha é composta de elementos independentes que se comunicam com os elementos circundantes por meio do contato entre bordas, os quais podem variar em função do tempo. Em contraste com o MEF e o MDF nenhum nó é comum a mais do que um elemento. Em aplicações típicas de mecânica das rochas os elementos representam a rocha intacta enquanto os espaços entre os elementos podem ser representados como juntas.

O MED permite tratar não-linearidades que podem surgir a partir de grandes deslocamentos, rotação, escorregamentos e separações. O método é particularmente adequado para maciços fraturados onde grandes deslocamentos são esperados. Em outras palavras, se as fraturas da rocha são comparáveis ao tamanho da escavação o MED é o mais adequado (PAN; REED, 1991).

### *2.5.2.2 Método de rede de fraturas discretas*

Esse método de rede de fraturas discretas (RFD) é o mais adequando para o estudo do fluxo de fluidos e transporte de massas em rochas fraturadas, através de um sistema de fraturas conectadas no qual um modelo contínuo equivalente for muito difícil de ser modelado (VILLAESCUSA, 2014).

### 2.5.2.3 Método de análise de deformação

O método de análise de deformação (MAD) possui uma análise totalmente descontínua que se assemelha e segue os procedimentos desenvolvidos para o MEF com um esquema de solução implícita. Já o MED tem um método de solução explícita enquanto o MAD é implícito, no MED as deformações e forças são desconhecidas enquanto no MAD os deslocamentos são desconhecidos, semelhante aos deslocamentos em programas baseados no MEF (BOBET *et al.*, 2009).

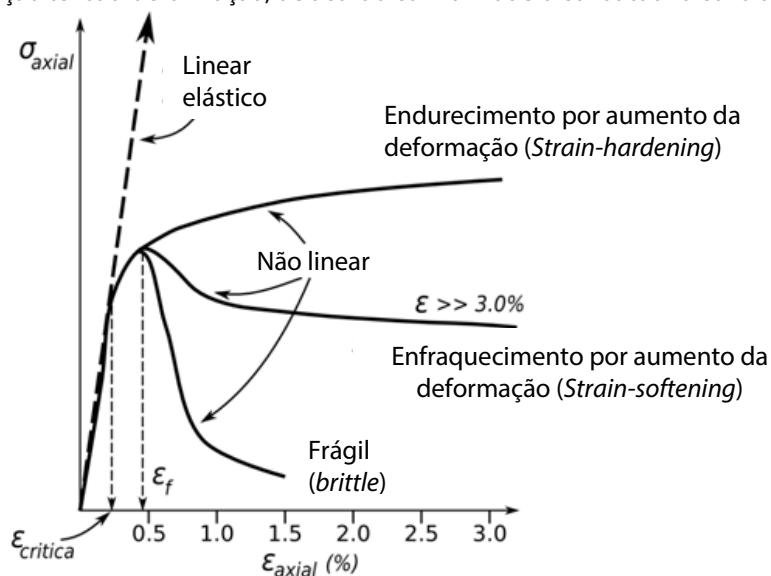
### 2.5.3 Modelo constitutivo

O modelo constitutivo é de interesse primordial, pois fornece a relação entre tensões e deformações que podem ser sustentadas pelo maciço rochoso. Os modelos constitutivos clássicos são os modelos baseados principalmente na teoria da elasticidade e plasticidade, mas com considerações especiais sobre os efeitos da fratura. O modelo de elasticidade linear baseado na lei de Hooke generalizada é a suposição amplamente adotada para o comportamento mecânico de rochas, especialmente para rochas duras. Isso ocorre pois a obtenção das constantes elásticas anisotrópicas é geralmente muito difícil. Quando a suposição CHILE (Contínuo, Homogêneo, Isotrópico, Linear Elástico) é adotada para maciços rochosos, os valores constitutivos simplesmente são caracterizados por duas propriedades de materiais independentes, o módulo de Young ( $E$ ) e o coeficiente de Poisson ( $\nu$ ) ou ainda através do módulo *Bulk* ( $K$ ) e o módulo *Shear* ( $G$ ) (JING, 2003).

De maneira geral todos os modelos geralmente se comportam de forma linear elástica até algum valor crítico de deformação ( $\epsilon_{\text{crítico}}$ ). Com o modelo elástico linear, a tensão aumenta linearmente com o aumento da deformação, porém os modelos de comportamento plástico, não lineares, são incapazes de sustentar a tensão desta forma.

Para o modelo plástico, não linear, a deformação no pico de tensão ( $\epsilon_f$ ) pode ser significativa. A escolha do modelo constitutivo é, portanto, extremamente importante quando se tenta entender o comportamento pós-deformação crítica (*post-yield*) dos maciços rochosos (Figura 20). Para tensões abaixo do ponto de ruptura as rochas duras têm comportamento elástico. Já no momento da ruptura normalmente apresentam ruptura frágil sem nenhuma resistência residual (HAJIABDOLMAJID; KAISER; MARTIN, 2002).

Figura 20 – Influência na relação tensão-deformação, de acordo com o modelo constitutivo considerado.



Fonte: Adaptado de Cepuritis (2010).

#### 2.5.4 Critério de falha do maciço rochoso

A situação de falha do maciço rochoso pode ser definida de várias maneiras, do ponto de vista estrutural, a ruptura é a condição na qual o maciço rochoso deixa de cumprir seu propósito funcional ou atinge um estado limite crítico. Os mecanismos reais envolvidos no estado de falha podem ser bastante complexos. A fundamental importância nos modelos contínuos é definir ou quantificar sob quais condições o maciço rochoso falha e seu comportamento após a falha.

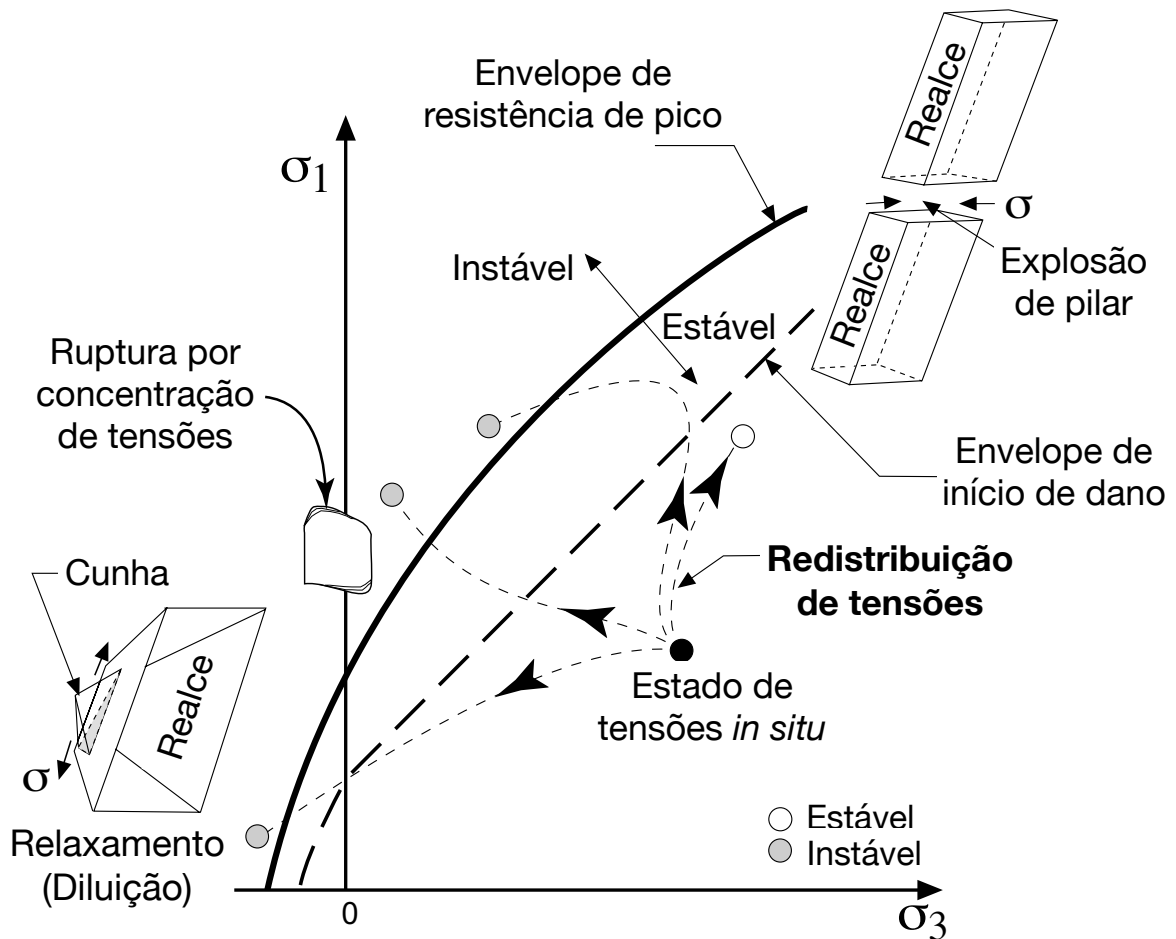
Modelos de plasticidade e elasto-plasticidade foram desenvolvidos e amplamente aplicados a rochas fraturadas desde 1970, com base principalmente na teoria clássica da plasticidade, com modelos típicos usando critérios de ruptura de Mohr-Coulomb e Hoek-Brown (HOEK, 1983; HOEK; BROWN, 1997, 2019).

O critério de Mohr-Coulomb considera as tensões normais e de cisalhamento para definir a envoltória linear de falha. Já o critério de falha de Hoek-Brown (1980) utiliza as tensões principais máximas e menores (espaço de tensão principal  $\sigma_1, \sigma_3$ ) sendo que a influência da tensão intermediária é ignorada.

No contexto das minas em rocha dura ou de comportamento frágil, sob o método de realces abertos, as altas tensões induzidas pelas aberturas podem causar grandes deformações, propagação de fraturas, deslocamentos ou colapso completo da escavação. Portanto, reconhecer o processo de ruptura e os modos de falha subsequentes desempenham um papel importante na avaliação da instabilidade dos realces.

Em um sentido amplo, as rupturas por queda de blocos foram identificadas como ocorrendo apenas sob relaxamento de tensões, por exemplo no *hangingwall* dos realces, provocando diluição. Já nos pilares, rupturas abruptas/explosões tendem a ocorrer em situações de alto confinamento. Esses fenômenos são possíveis conforme for a redistribuição das tensões, a partir das tensões *in situ* atuantes (Figura 21). Dependendo da relação largura/altura do pilar, podem romper por esmagamento ou por cisalhamento do núcleo do pilar, assumindo gradativamente uma geometria de ampulheta, com sua ruptura iniciando pelas bordas (DIEDERICHS, 1999, CASTRO et. al, 2012).

Figura 21 - Representação esquemática da redistribuição de tensões e dos mecanismos de ruptura em escavações subterrâneas.



Fonte: Adaptado de Martin et al (1999).

#### 2.5.4.1 Strength-Stress Ratio – SSR

A tensão desviadora é a diferença entre as tensões principais máximas e mínimas e seu uso mais comum é através do conceito de tensão máxima de cisalhamento. Este critério é frequentemente usado para avaliar o efeito de tensões induzidas pelas escavações no

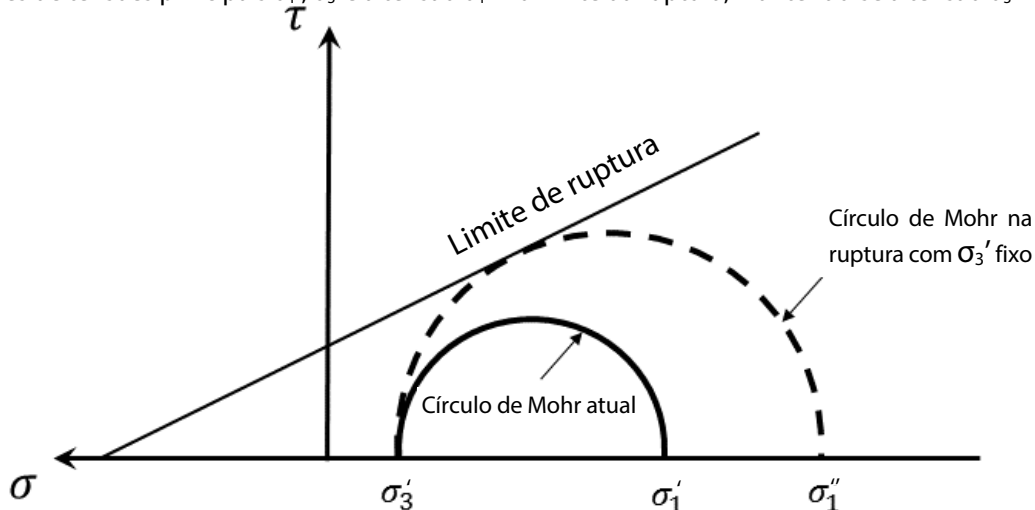
maciço rochoso, especialmente em um ambiente de escavações profundas ou de rochas com comportamento frágil (DIEDERICHS, 1999).

Ao utilizar essa abordagem, modelos numéricos elásticos 2D e 3D são aplicados para estimar áreas onde o dano poderia se acumular e criar o potencial para rupturas abruptas ao redor das aberturas devido a altas tensões compressivas induzidas (CASTRO et al., 2012).

O  $\sigma_1'$  e o  $\sigma_3'$  são as tensões principais efetivas calculadas para o elemento em questão e o  $\sigma_1''$  é a tensão principal máxima que a rocha pode resistir até o instante da ruptura, mantendo-se o  $\sigma_3'$  constante. O  $\sigma_1''$  é função da envoltória de resistência da rocha, no caso do critério de Mohr-Coulomb, da coesão e do ângulo de atrito. Um valor de SSR próximo a 1 indica que o elemento atingiu o limite de resistência e está em estado de ruptura. A equação 4 a seguir demonstra o cálculo do SSR. A Figura 22 a seguir mostra a representação do círculo de Mohr, com as tensões principais e com a envoltória de ruptura de Mohr-Coloumb.

$$SSR = \left| \frac{\sigma_1'' - \sigma_3'}{\sigma_1' - \sigma_3'} \right| \quad \text{Equação 4}$$

Figura 22 – Representação do círculo de Mohr, com a envoltória de ruptura linear de Mohr-Coulomb e os valores de tensões principais  $\sigma_1'$ ,  $\sigma_3'$  e a tensão  $\sigma_1''$  no limite da ruptura, mantendo-se a tensão  $\sigma_3'$  fixa.



Fonte: Adaptado de Itasca (2005).

Para a análise de estabilidade dos pilares pode-se utilizar este critério, sendo escolhido o valor de SSR que indicaria os elementos em estado de ruptura devido a concentração de tensões.

#### 2.5.4.2 Ruptura por baixo confinamento

Um indicador de instabilidade também bastante utilizado é o da presença de tensão de tração na superfície das escavações. É notável que a resistência à tração de maciços rochosos é praticamente nula, portanto onde há um confinamento muito baixo, ou tensão de tração está presente, existe um potencial de falha pela gravidade. Quando a magnitude da tensão  $\sigma_3$  exceder o limite de resistência à tração do maciço rochoso, independentemente do valor da tensão  $\sigma_1$ , a escavação é então considerada sob falha (DIEDERICHS, 1999).

### 2.6 Algoritmos genéticos

Os algoritmos genéticos (AGs), em inglês *Genetic Algorithms* (GAs), fazem parte do grupo dos algoritmos evolutivos. Esse é um grupo de algoritmos capaz de aprender ou ajustar-se enquanto é executado, sendo sua programação computacional relativamente simples (COLEY, 1999). Os algoritmos genéticos também são definidos como heurísticos, pois eles fornecem um bom resultado na aproximação de uma solução ótima, de forma muito mais rápida de que nos métodos tradicionais. Como o tempo para encontrar uma solução exata, se uma existir, é muito maior do que uma solução heurística, os algoritmos heurísticos tendem a ser utilizados para problemas complexos ou em grandes banco de dados (PEARL, 1984). A solução obtida é geralmente mais complexa e mais eficiente que um engenheiro poderia conseguir, modelando o problema manualmente.

#### 2.6.1 História dos algoritmos genéticos

Nas décadas de 1950 e 1960 vários cientistas da computação estudaram de maneira independente os sistemas evolutivos, com a ideia de que a evolução poderia ser utilizada como ferramenta de otimização para problemas de engenharia. A ideia central em todos esses sistemas era evoluir uma população de candidatos até a solução do problema, usando operadores inspirados na variação genética e seleção natural (MITCHELL, 1996).

O primeiro algoritmo genético foi inventado por John Holland na década de 1960 e desenvolvimentos posteriores dele e de seus colegas foram realizados na década seguinte. O objetivo inicial de Holland era estudar o fenômeno de evolução e adaptação que ocorria na natureza e explorar como esses conceitos poderiam ser integrados em computadores (COLEY, 1999).



No seu livro lançado em 1975, ele apresenta o algoritmo como uma forma abstrata da evolução biológica. Além disso, ele forneceu a base para adaptação do algoritmo genético. O seu método de mover de uma população para uma nova incorpora as técnicas básicas de acasalamento, mutação e inversão. Trabalhos anteriores apenas utilizavam a mutação como ferramenta de evolução. No seu modelo os indivíduos são apresentados por um conjunto de valores binários, ou seja, “zeros” e “uns” onde os melhores indivíduos podem produzir mais descendentes do que os piores. Pesquisas com algoritmos genéticos aumentaram na década de 1970 e 1980 em virtude do avanço computacional (MITCHELL, 1996).

### **2.6.2 Vantagens e aplicações**

Em comparação com os métodos tradicionais de otimização baseados em cálculo, busca exaustiva e aleatórios, os algoritmos genéticos apresentam resultados muito mais satisfatórios. No método tradicional de cálculo, o gradiente da função objetivo permite apenas encontrar o ponto de máximo local, que pode não estar próximo do ponto de máximo global. Os métodos exaustivos, requerem um espaço de procura finito de possíveis soluções para a função objetivo, sendo então testados todos, um por vez, para encontrar o máximo ou mínimo. Já no caso dos algoritmos aleatórios, amostras representativas são sorteadas dentro da população de soluções e dentro desse conjunto o maior valor é encontrado. Ainda que esse seja um método mais rápido, em comparação com o método exaustivo, o resultado não será melhor, podendo chegar próximo da solução ótima ou apenas ficando muito distante dela (GOLDBERG, 1989).

Os algoritmos genéticos são paralelos, ou seja, onde os outros algoritmos conseguem explorar a solução em apenas um espaço, uma direção por vez, os algoritmos genéticos podem trabalhar em múltiplas direções por vez. Isso significa que onde os outros algoritmos acabam ficando presos e precisam recomeçar, os algoritmos genéticos podem facilmente eliminar este caminho, seguindo em uma direção mais promissora (MITCHELL, 1996).

Os algoritmos genéticos funcionam bem para problemas onde o cenário de análises é amplo e complexo. Não são necessárias informações específicas nos problemas a serem resolvidos, pois as mudanças produzidas nos indivíduos candidatos são aleatórias e uma

função avaliativa determina quando essas mudanças produzem alguma melhora (COLEY, 1999).

As vantagens já mencionadas dos algoritmos genéticos, fizeram com que fosse decidido pela utilização deste método como meio para otimizar o problema de arranjo de realces abertos e pilares verticais proposto neste trabalho. O algoritmo genético a ser desenvolvido permitirá lidar com grandes banco de dados, além de poder ser adaptado a outros cenários no futuro, com pequenos ajustes em seu funcionamento, reduzindo o tempo de execução.

### **2.6.3 Estrutura básica**

Dadas as variações, pode-se dizer de maneira geral que os chamados algoritmos genéticos têm pelo menos os seguintes elementos em comum: população de indivíduos com cromossomos distintos, seleção de acordo com a aptidão, cruzamento para produzir novas gerações e mutação aleatória das novas gerações (COLEY, 1999).

A seleção é o operador que seleciona cromossomos na população para reprodução. Quanto mais apto o cromossoma, mais vezes ele pode ser selecionado para reprodução. A operação de cruzamento (*crossover*) escolhe randomicamente um *locus* e troca as subsequências antes e depois desse *locus* entre dois cromossomos para criar dois descendentes. Por exemplo, as cadeias 10000100 e 11111111 poderiam ser cruzadas após o terceiro *locus* em cada uma para produzir os dois descendentes 10011111 e 11100100 (Figura 23). O operador de cruzamento imita aproximadamente a recombinação biológica entre dois organismos de cromossomo único (MITCHELL, 1996).

Figura 23 – Exemplo de operação de cruzamento entre dois indivíduos e seus descendentes.

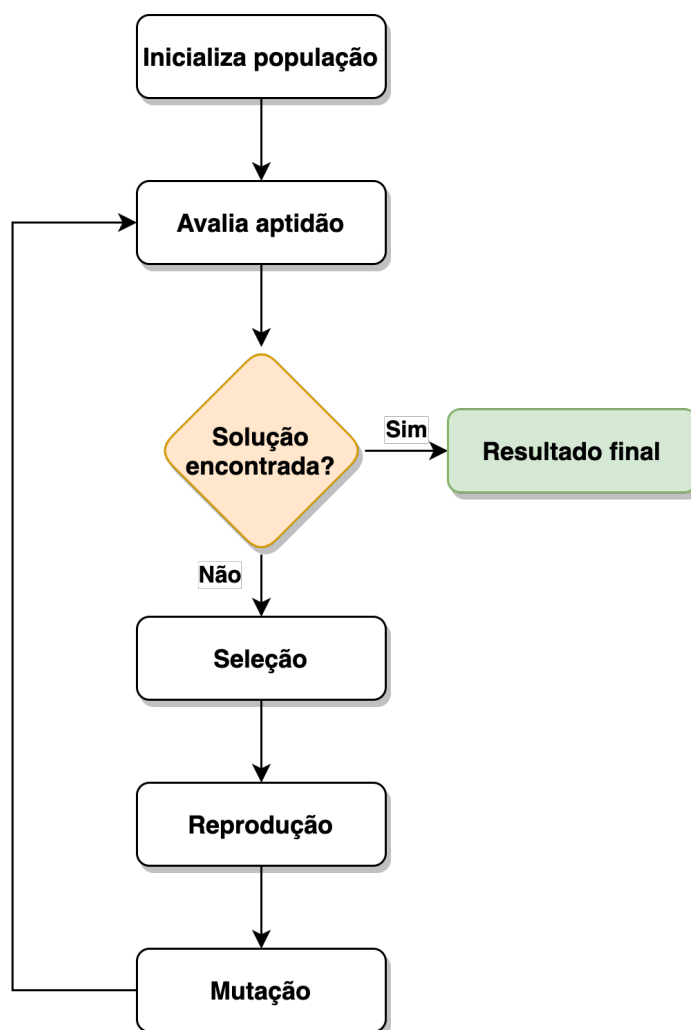


Fonte: Autoria própria.

Por sua vez o operador de mutação inverte aleatoriamente alguns dos *bits* em um cromossomo. Por exemplo, a sequência 00000100 pode sofrer mutação em sua segunda posição para gerar a sequência 01000100. A mutação pode ocorrer em cada posição de bit com uma probabilidade geralmente muito pequena, por exemplo, 0,001 (MITCHELL, 1996).

Na Figura 24 a seguir é possível visualizar um fluxograma geral da estrutura básica de um algoritmo genético.

Figura 24 – Fluxograma mostrando a estrutura básica utilizada em um algoritmo genético.



Fonte: Autoria própria.

## 2.7 Trabalhos recentes relacionados ao tema

Alguns trabalhos recentes vêm sendo desenvolvidos por autores, com objetivo de otimizar o dimensionamento de pilares e realces. Alguns destes trabalhos buscam apenas a otimização geomecânica, enquanto outros tentam relacioná-la com a otimização econômica.

De maneira geral os trabalhos elencados a seguir apresentam metodologias e aplicações distintas, sendo possível destacar o uso de modelagem numérica, redes neurais, algoritmos genéticos ou equações matemática de otimização. Os artigos mais pertinentes ao tema são apresentados a seguir em ordem cronológica de sua data de publicação.

O artigo de título: *"Pillar design by combining finite element methods, neural networks and reliability: a case study of the Feng Huangshan copper mine, China"* (DENG et al, 2003)

combina modelagem numérica por elementos finitos e redes neurais, com o objetivo de otimizar os pilares a partir da incerteza nas propriedades do maciço rochoso. Sendo como resultado do trabalho um banco de dados gerado com índice de confiança e probabilidade de falha para um dado diâmetro de pilar escolhido.

A dissertação de mestrado de título: “Dimensionamento ótimo de realces abertos/câmaras inclinadas e pilares via programação matemática não-linear” (SOUZA, 2011), utiliza a metodologia inicialmente proposta por Figueiredo e Curi (2003 citado por Figueiredo e Curi, 2004) onde um problema padrão de programação matemática é formulado com o objetivo de maximizar a recuperação, respeitando entretanto, as restrições de segurança dos pilares e os requisitos de estabilidade e operacionais dos realces. Utilizando-se fórmulas de resistência consagradas, são apresentados exemplos de solução para arranjos realistas aplicados à lavra por realces abertos e por câmaras e pilares inclinados. Resultados de estudos paramétricos são relatados, comparando a recuperação por meio da metodologia usualmente utilizada e a metodologia de dimensionamento ótimo proposta. Têm-se ainda, os resultados avaliados via modelos numéricos por elementos de contorno.

O artigo de título: “*Economic optimization of rib pillars placement in underground mines*” (ANDRADE *et al*, 2019) utiliza programação matemática para definir um arranjo de realces e pilares verticais, utilizando teor médio das fatias (*slices*) analisadas e variando o vão (*span*) de realces dentro de um intervalo mínimo e máximo, mantendo o tamanho de pilares constante. Sendo assim o algoritmo busca o melhor cenário econômico, alterando de posição no espaço os realces abertos e os pilares, onde podem existir casos que com vãos menores em áreas com teor baixo, possibilitam um melhor posicionamento em áreas de alto teor. Entretanto, o método não permite variar o tamanho dos pilares.

O artigo de título: “*Optimization of room-and-pillar dimensions using automated numerical models*” (NAPA-GARCÍA *et al*, 2019) apresenta uma metodologia de otimização utilizando modelos numéricos automatizados, em função da recuperação de lavra no método de câmaras e pilares, estabelecendo fator de segurança mínimo para os pilares e limite de convergência máxima do teto. Tamanhos mínimos operacionais também foram estabelecidos para as galerias e para os pilares. O modelo foi realizado representando somente um pilar e a metade da galeria projetada.

O artigo de título: *“A planning approach for polymetallic mines using a sublevel stoping technique with pillars and ultimate stope limits”* (SARI; KUMRAL, 2020) tem como proposta um algoritmo novo de otimização dos subníveis, minimizando os custos com desenvolvimentos horizontais e posteriormente definindo os limites para os realces. Ao ajustar a altura dos realces para a maior possível, isso permite aumentar a separação entre os subníveis, ou em outros casos diminuir a altura dos realces para poder acessar os corpos de minério. Com relação ao tamanho dos pilares, precisa ser especificado no início do algoritmo e com isso seja gerado um arranjo de realces e pilares.

O artigo de título: *“Evaluation of the effect of geometrical parameters on stope probability of failure in the open stoping method using numerical modeling”* (HEIDARZADEH et al, 2019) busca avaliar o efeito dos parâmetros geométricos na estabilidade de um estudo de caso em mina canadense de realces abertos. Os autores utilizam simulações estocásticas de cenários que são analisados por modelagem numérica de diferenças finitas, sendo sua métrica a probabilidade de falha. O artigo discute o efeito da variação do raio hidráulico e do mergulho do *hangingwall* na probabilidade de falha observada.

O artigo de título: *“The integrated optimization of underground stope layout designing and production scheduling incorporating a non-dominated sorting genetic algorithm (NSGA-II)”* (FOROUGHIA et al, 2019) utiliza algoritmos genéticos no processo integrado de otimização e sequenciamento dos realces, considerando parâmetros como tamanho mínimo e máximo de realces, separação ótima entre níveis e restrições de lavra simultânea de realces adjacentes que serão posteriormente preenchidos. O trabalho tem objetivo múltiplo de otimizar o Valor Presente Líquido (VPL) e a recuperação de metal, portanto o resultado obtido é um gráfico do VPL pela recuperação de metal. Entretanto, o artigo não considera a análise geomecânica. Em suas conclusões, o artigo relata o ganho computacional em se utilizar o algoritmo heurístico.

# Capítulo 3

## 3 Metodologia

Nos capítulos anteriores, foram apresentadas as metas e objetivos da tese (Capítulo 1), a revisão de aspectos teóricos sobre algoritmos otimizadores do contorno e localização de realces, sobre os métodos numéricos e os algoritmos genéticos (Capítulo 2). Nesse capítulo, será apresentada a metodologia utilizada para o alcance das metas e objetivos, as etapas realizadas no estudo e os recursos que foram necessários. O desenvolvimento do algoritmo basicamente se divide em relação ao código em linguagem *Python* e aos códigos do FLAC3D em linguagem de comandos, além do uso de sua linguagem de programação interna *FISH*, o que possibilita a realização da modelagem numérica automatizada de cenários. Os códigos completos e comentados, produzidos em linguagem *Python*, estão disponibilizados no Apêndice A. Por sua vez, no Apêndice B estão contidos os códigos completos do FLAC3D, também com os devidos comentários.

Para cumprir com as metas estabelecidas, foi considerado um estudo de caso de um painel de lavra, de extensão de uma mina subterrânea de ouro, que utiliza o método de lavra de realces em subníveis abertos, com sequência de lavra descendente, onde pilares verticais são deixados de maneira permanente. Inicialmente os arquivos do estudo de caso, *wireframes*, modelo de blocos, topografias, modelos geológicos etc., foram importados no *software* de mineração Micromine®.

No *software* foi realizada a otimização do contorno dos realces, sendo realizadas fatias (*slices*) verticais ao longo do eixo do *strike* do corpo mineralizado, com espessuras de 1m, produzindo volumes fechados de *wireframes*. Os *slices* tiveram seu teor médio de ouro calculados por meio do modelo de blocos disponível para o depósito. Também foram calculadas as distâncias médias de transporte, obtidas entre os pontos centrais de cada *slice* em relação ao ponto central do subnível, onde ocorre o carregamento do minério nos caminhões.

Com os *slices* quantificados individualmente, dentro do algoritmo em *Python* é criada a população inicial, por simulação estocástica, onde cada indivíduo gerado possui um arranjo de realces e pilares, a partir da aglutinação de *slices* sorteados. Esse arranjo precisa ser válido, ou seja, respeitar as restrições mínimas e máximas para as dimensões atribuídas para os realces e os pilares. Os requisitos estabelecidos têm o objetivo de garantir que os

indivíduos sejam minimamente operacionalizáveis, com um tamanho mínimo definido pelo usuário do algoritmo, para os pilares e para os realces.

Após este processo de inicialização de uma população aleatória válida, os indivíduos são enviados para análise numérica separadamente, via protocolo *socket*, para o software de modelagem numérica escolhido. Para o problema em questão a versatilidade de programação do FLAC3D® e sua integração com a linguagem *Python*, condicionaram sua escolha. Nesta etapa de modelagem, ocorre a geração de um grid automático de elementos que serão refinados nos locais de interesse pelo método *octree*. Portanto, próximos aos realces e pilares o tamanho dos elementos será menor em comparação com as outras zonas distantes do modelo, conseqüentemente a resolução dos resultados e o desempenho global do modelo são melhorados. Após, o modelo numérico é inicializado, com as condições de tensões *in situ*, condições de borda, propriedades e modelo constitutivo. O modelo tem então as galerias escavadas e após atingir o equilíbrio interno é salvo um arquivo de estado (*saved state*). Nas etapas seguintes o mesmo arquivo de estado salvo será utilizado como ponto de partida para a escavação de realces. Em cada etapa de cálculo do modelo numérico, o objetivo é atingir o equilíbrio interno (*unbalanced forces*) definido, para o somatório de forças (ITASCA, 2005).

Ainda no FLAC3D, uma função interna escrita em linguagem *FISH* (linguagem de programação do FLAC3D), fornece o desempenho dos pilares e da diluição, em função da estabilidade do *hangingwall*, para cada indivíduo. Após, cada indivíduo é avaliado pela função *fitness* do algoritmo genético, sendo então atribuído um valor econômico. A atribuição deste valor econômico leva em conta os parâmetros de desempenho geomecânico: diluição esperada, porcentagem de falha nos pilares, além dos parâmetros quanto a distância média de transporte, teor médio, toneladas de minério etc., a partir dos custos e preço de venda que foram previamente informados pelo usuário.

Neste ponto finaliza a etapa de inicialização da população, e inicia o algoritmo genético. Sendo assim, indivíduos da população serão sorteados aleatoriamente com o objetivo de reprodução entre eles, gerando descendentes que compartilham parte de seus códigos genéticos. Os descendentes poderão sofrer mutação, garantindo sua variabilidade genética. Após, são enviados para modelagem no FLAC3D e posteriormente avaliados economicamente pela mesma função *fitness*. Com um valor econômico atribuído a estes

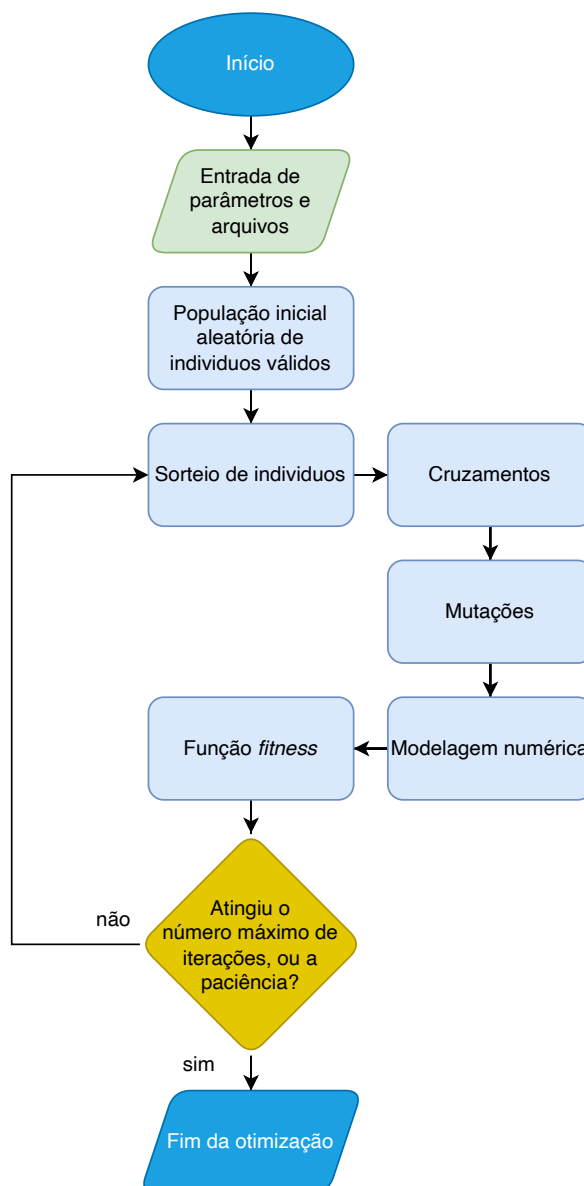


descendentes, caso o valor econômico obtido seja maior do que o dos seus pais, estes substituirão seus pais na população.

Cada etapa de seleção de indivíduos, cruzamento, mutação e avaliação pela função *fitness* é considerada uma geração (iteração) realizada. O algoritmo segue sorteando novos indivíduos e realizando as demais etapas até que o número máximo de iterações, definido pelo usuário, seja alcançado. Ao se atingir o número máximo de gerações definidas o algoritmo é finalizado e o modelo otimizado do arranjo de realces abertos e pilares foi obtido.

O algoritmo também pode ser finalizado após um determinado número de iterações sem que ocorra melhora do resultado econômico. Esse limite de iterações é definido pelo usuário como “paciência” e caso alcançado o algoritmo também é finalizado e o modelo ótimo do arranjo de realces abertos e pilares foi obtido. Sendo assim, o critério de otimização e parada é o valor econômico global do cenário. A Figura 25 mostra o fluxograma geral proposto para o algoritmo nesta tese.

Figura 25 - Fluxograma geral da metodologia do algoritmo otimizador proposto.



Fonte: Autoria própria.

### 3.1 Recursos necessários

Os recursos necessários para o desenvolvimento da tese consistem em um computador pessoal com hardware atualizado, provido de pacotes de *software* que executem as diferentes etapas do algoritmo. Os principais pacotes de *software* necessários e adotados, são o Micromine®, com licença educacional válida obtida pela UNIPAMPA, o software FLAC3D®, com licença educacional válida pela UFRGS, além da plataforma de programação (IDE) gratuita Anaconda, com os respectivos pacotes para programação de código em linguagem *Python*.

O computador necessário para o desenvolvimento da tese deve ser capaz de processar os modelos numéricos e o algoritmo genético para o estudo de caso utilizado, exigindo um tempo de processamento compatível com o tempo disponível. De maneira geral, o hardware não necessita de um processamento muito superior aos computadores utilizados pelos engenheiros de minas nas empresas de mineração. Principalmente porque o algoritmo proposto visa a criação de uma metodologia capaz de ser aplicada no dia a dia dos engenheiros de planejamento das minerações subterrâneas.

### 3.2 Algoritmo em linguagem *Python*

O algoritmo programado em linguagem *Python*, é responsável pelo trabalho de importação, preparação dos arquivos, sendo iniciado desde o software Micromine®, ou do seu próprio console de execução, capaz de se comunicar com o software FLAC3D®, realizar a classificação econômica do cenário, otimização com o algoritmo genético e finalização com produção de relatórios, tabelas e os arquivos contendo as geometrias otimizadas.

#### 3.2.1 Funções para manipulação das geometrias

Uma função importa os *slices*, exportados em formato ASCII do software Micromine, junta os *slices* que forem definidos, conforme o cenário aleatório, e exclui os triângulos duplicados das faces comuns de união entre os *slices*. Após agrupar os *slices* de um cenário, dois arquivos de geometria em formato STL são gerados, um de realces e outro de pilares. Por efeito da precisão numérica e de ganhos computacionais é realizada nesta etapa uma translação nas coordenadas, trazendo-as próxima da origem. A Figura 26 abaixo demonstra a sintaxe de parte de um arquivo STL de exemplo.

Figura 26 – Sintaxe do arquivo de geometria STL.

```
solid best_pilares_scenario
facet normal -1.0 0.0 0.0
  outer loop
    vertex 14.142 47.938738 -36.417
    vertex 14.142 49.52849 -33.116918
    vertex 14.142 55.373495 -36.417
  endloop
endfacet
```

Fonte: Autoria própria.

### 3.2.2 Comunicação via *Socket* com o FLAC3D

Para realizar a comunicação entre o código *Python* e o software FLAC3D é necessário a utilização do pacote *Itasca* de biblioteca *Python*, desenvolvido por Furtney (2020), disponível para *download* no repositório PyPI. Essa biblioteca implementa uma conexão via protocolo de rede *Socket* entre o *kernel* do *Python* e os pacotes de *software* de modelagem numérica da *Itasca*, sendo neste estudo o FLAC3D. Este tipo de conexão permite comunicar-se com computadores diferentes em rede, porém considerando o uso do algoritmo tanto o servidor quanto o cliente foram utilizados ambos no mesmo computador. O servidor da conexão é o *kernel* do *Python* e o cliente o software FLAC3D.

As funções desta biblioteca permitem iniciar o programa via código *Python*, estabelecer a conexão *socket* e assim permitir o envio e recebimento de variáveis. A existência desta biblioteca é que possibilita o algoritmo genético realizar várias simulações diferentes e automatizadas ao longo do processo de otimização. Primeiramente todos os parâmetros de configuração para realizar a modelagem numérica são enviados do *Python* para o FLAC3D via conexão *socket*, sendo iniciada a modelagem numérica após esta etapa.

Com a finalização da modelagem, o FLAC3D envia ao *Python* um valor numérico de “-666” via conexão *socket*, indicando a finalização da execução da modelagem e obtenção dos resultados. Neste instante, tanto a sessão aberta do FLAC3D, como a conexão *socket* estabelecida, são encerradas. Para a execução de um novo modelo no FLAC3D a conexão *socket* precisará ser reestabelecida e o envio dos parâmetros novamente realizado.

### 3.2.3 Função geradora de cenários estocásticos válidos

Uma função geradora de cenários foi estabelecida, tendo como variáveis de entrada o número de *slices* de cada nível e o tamanho mínimo e máximo de pilares e de realces. Dentro desta função geradora, duas classes foram definidas, uma delas se refere a construção dos realces de um único nível, criando um objeto “*Level*”. A outra se refere a um cenário completo, que engloba todos os níveis, criando um objeto “*Design*”. Cada uma destas classes possui um módulo de inicialização (*init*), além de módulos que alteram o objeto, quando forem solicitados.

Quando a classe “*Level*” é chamada, um objeto é criado de forma a preencher o nível em questão com porções de pilares e realces de forma alternada. Para iniciar o preenchimento do primeiro volume ocorre um sorteio para definir o tipo, se pilar ou realce.

Definido o tipo de volume, outro sorteio é realizado de forma a definir o tamanho, considerando os limites mínimos e máximos previamente estabelecidos para cada tipo.

Caso o primeiro volume sorteado seja do tipo pilar, neste caso será desconsiderada a restrição do tamanho mínimo. O mesmo pode ocorrer caso o último volume do nível também seja sorteado como pilar, neste caso a restrição do tamanho mínimo também será desconsiderada. A função também prevê que se no extremo de um nível o volume sorteado for do tipo realce, e se o tamanho restante for menor do que o tamanho mínimo para realce, a função passará a considerá-lo como pilar, sendo este volume unido ao pilar anterior, ainda que ultrapasse o tamanho máximo definido. Ainda dentro da classe “*Level*” outro módulo denominado “*encode\_as\_binary*” irá traduzir os elementos definidos com as palavras “pilar” e “realce” em “0” e “1”, respectivamente.

### **3.2.4 Função benefício (*profit*)**

A função benefício, que é capaz de avaliar economicamente um cenário, foi denominada no código *Python* como função *profit*. Para o cálculo desta função são necessárias as variáveis de entrada, preenchidas pelo usuário na interface do programa (Tabela 2), os atributos dos *slices*, que estão em um arquivo indicado pelo usuário em formato *Excel*® (.xlsx) que é lido pelo *Python*, contendo o teor, tonelada e distância média de transporte (DMT) de cada *slice*.

Importante salientar que o valor econômico atribuído ao cenário é o próprio valor presente líquido (VPL), considerando que o prazo de lavra do painel otimizado é de cerca de dois anos, por tanto não será considerada taxa de atualização ao valor. Em estudos de caso com vida útil maiores que dois anos, seria importante atribuir uma taxa de atualização ao valor econômico obtido, além de considerar a sequência de lavra dos realces no tempo.

Além destes dados econômicos, também são necessários os resultados obtidos na modelagem numérica, que são armazenados automaticamente pelo FLAC3D® em um arquivo formato ASCII (.dat) ao final da modelagem e lido pelo *Python*. O arquivo contém a porcentagem de falha dos pilares, porcentagem de diluição potencial no cenário, volume de diluição, volume de elementos de pilares considerados sob falha e a recuperação global de lavra do cenário.

Tabela 2 - Variáveis de entrada inseridas pelo usuário na interface gráfica do programa.

Variável de entrada	Unidade	Variável Python
Preço de venda do ouro	US\$/g	<i>price</i>
Recuperação de lavra nos realces	%	<i>r1</i>
Recuperação do processo de beneficiamento	%	<i>r2</i>
Diluição máxima aceitável com o <i>hangingwall</i>	%	<i>max_dil</i>
Porcentagem máxima aceitável de falha em pilares	%	<i>PAF_max</i>
Custo de lavra	US\$/t	<i>c_lavra</i>
Custo de transporte	US\$/t/m	<i>c_transp</i>
Custo de processo	US\$/t	<i>c_process</i>
Custos gerais e administrativos	US\$/t	<i>c_ga</i>
Custo de refino	US\$/g	<i>c_ref</i>
Custo de <i>royalty</i>	% do <i>gross value</i>	<i>royalty</i>
Custo de suporte secundário no <i>hangingwall</i> (quando acima da máxima diluição permitida)	US\$/(m <sup>3</sup> de diluição no cenário)	<i>hw_support</i>
Eficiência do suporte secundário no <i>hangingwall</i>	%	<i>hw_support_efficiency</i>
Custo de reforço de pilares	US\$/(m <sup>3</sup> de pilar com falha)	<i>pillar_support</i>

Fonte: Autoria própria.

Inicialmente é calculada a receita bruta (*gross value*) dado pela Equação 4 a seguir, onde *ton* representa a massa em toneladas dos realces para o cenário avaliado:

$$gross\_value = price * teor * ton * r1 * r2 \quad \text{Equação 4}$$

Um teste lógico é realizado para verificar se a porcentagem de falha de pilares (PFP) do cenário analisado é maior ou igual do que a porcentagem máxima aceitável de falha em pilares (*PAF\_max*), caso verdadeiro será calculado o custo com reforço de pilares.

Portanto, se  $PFP \geq PAF\_max$  então o custo com reforço de pilares pode ser calculado pela Equação 5. Caso a lógica não seja verdadeira, o *pillar\_support\_cost* será definido como sendo igual a zero, pois a porcentagem de falha de pilares está dentro do limite aceitável, definido pelo usuário.

$$pillar\_support\_cost = (pillar\_support * pillar\_vol\_failure) \quad \text{Equação 5}$$

Outro teste lógico é realizado para verificar se a diluição do cenário é maior do que a diluição máxima aceitável com o *hangingwall*. Portanto se  $dil \geq max\_dil$  então o custo com suporte secundário no *hangingwall* será dado pela Equação 6 abaixo.

$$hw\_support\_cost = hw\_support * dil\_vol \quad \text{Equação 6}$$

Por sua vez, considerando a eficiência atribuída ao suporte secundário adicionado no *hangingwall*, a diluição real será calculada pela Equação 7 abaixo. Caso a diluição do cenário atual for menor ou igual do que a máxima diluição aceitável, o custo de suporte secundário será considerado zero e a diluição real será a própria diluição estimada pelo modelo numérico.

$$actual\_dil = dil - ((dil - max\_dil) * hw\_support\_efficiency) \quad \text{Equação 7}$$

A Equação 8 a seguir demonstra o cálculo dos custos.

$$cost\_value = (((ton * r1 * (1 + actual\_dil)) * (c\_lavra + c\_process + c\_ga + (c\_transp * dist))) + (teor * ton * r1 * r2 * c\_ref) + (royalty * gross\_value) + pillar\_support\_cost + hw\_support\_cost)$$

Equação 8

Por fim, o lucro líquido (*net profit*) será calculado pela receita bruta (*gross value*) menos os custos (*cost value*), conforme Equação 9.

$$profit = gross\_value - cost\_value \quad \text{Equação 9}$$

### 3.2.5 Algoritmo genético

Para a implementação completa do algoritmo genético, as funções parciais relativas à etapa de cruzamento (*crossover*), mutação e aplicação da função *fitness* precisam ser construídas, garantindo o funcionamento do algoritmo evolutivo de otimização. Nos tópicos a seguir, cada uma destas funções tem a sua implementação demonstrada.

#### 3.2.5.1 Função de cruzamento

A função de cruzamento (*crossover*) será responsável pelo cruzamento entre dois indivíduos da população selecionados, sendo estes denominado pais. O resultado será a geração de dois filhos, tendo os filhos uma porção do código genético de seus pais. O código genético a ser transmitido nos cruzamentos se dará pela troca de níveis de produção entre

os pais. O número de níveis a sofrerem a troca será sorteado aleatoriamente sendo no mínimo um e no máximo a metade mais um do número de níveis, arredondado ao inteiro mais próximo.

#### 3.2.5.2 Funções de mutação

Para a etapa de mutação do algoritmo genético, quatro formas diferentes foram consideradas. Sendo elas definidas a seguir: mutação de escavação (*dig mutation*), mutação de preenchimento (*fill mutation*), mutação de aglutinação (*merge mutation*) e mutação de separação (*split mutation*). Sempre que a função de mutação for invocada, ela irá modificar o objeto *Design* em questão. Já o tipo de mutação será definido por sorteio aleatório dentre os tipos elencados acima. Para qualquer uma das mutações possíveis, inicialmente será sorteado o nível de produção o qual ela irá ocorrer e posteriormente a posição dentro do nível.

A mutação *dig* é do tipo escavação e poderá atuar reduzindo o tamanho de um dos lados do pilar, dentre os pilares elegíveis. São considerados elegíveis os pilares que tiverem um tamanho maior do que o tamanho mínimo estabelecido. O tamanho da redução a ser realizada no pilar será o valor sorteado entre 1m e o tamanho atual do pilar menos o tamanho mínimo, ou seja, de forma que mesmo após a mutação o tamanho mínimo de pilar seja respeitado. O lado a ser reduzido também é sorteado, e caso o pilar sorteado seja dos extremos e coincida com o lado sorteado, um realce novo será inserido se houver condições.

Por sua vez a mutação *fill* é do tipo de preenchimento, justamente o inverso da mutação *dig*, ou seja, essa mutação é capaz de preencher um volume sorteado que antes representava um realce, de forma a agregar dimensão a um dos pilares adjacentes, conforme sorteio do lado. Realces elegíveis são escolhidos e dentre estes um é sorteado para sofrer a mutação. O tamanho do preenchimento a ser realizado também é sorteado, com a condição de que mesmo após a mutação a dimensão resultante do realce se mantenha dentro do intervalo previsto.

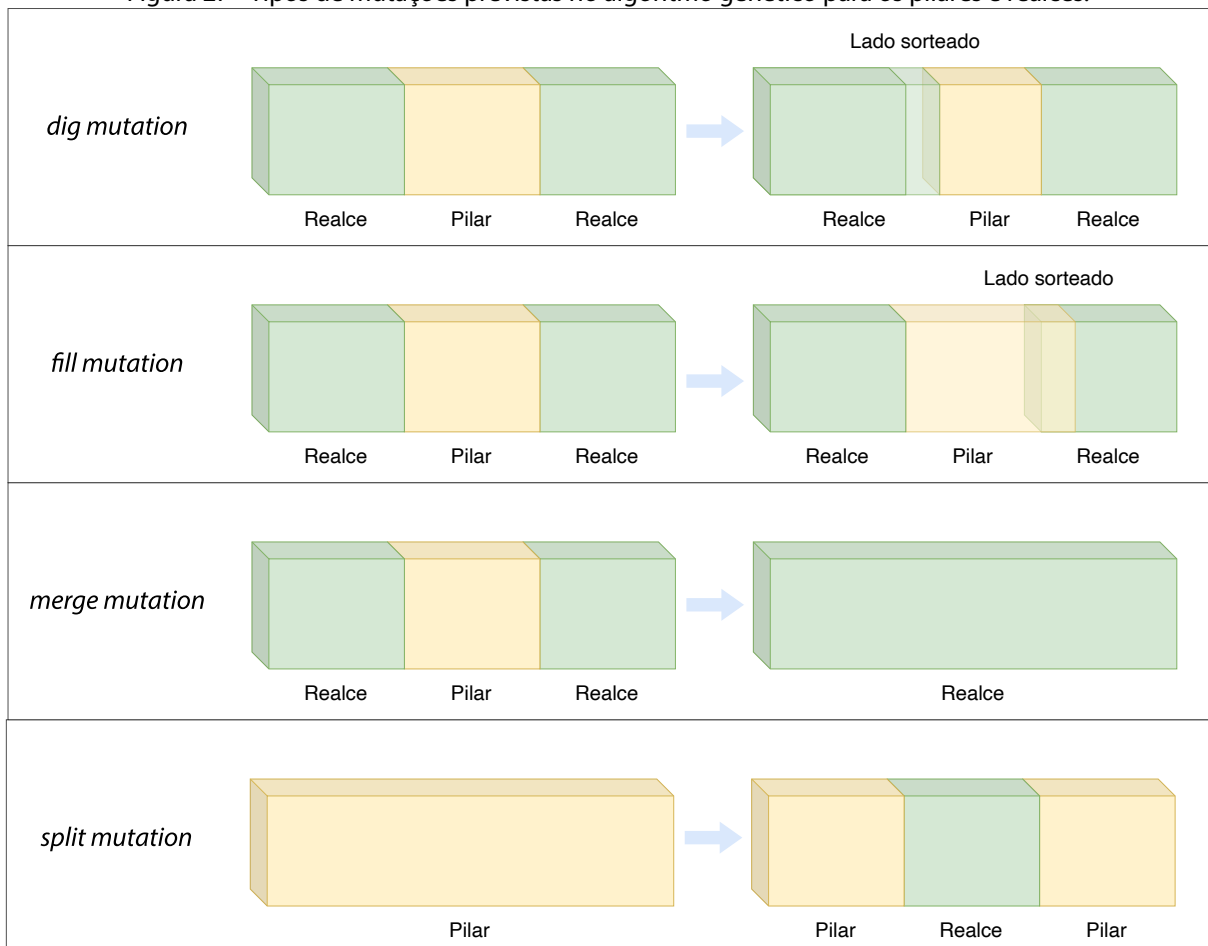
A mutação *merge* é do tipo aglutinação, ou seja, realizará a união entre três volumes, de forma a unir pilares que estão entre um realce, ou unir realces separados por um pilar, caso cumpram com a restrição de tamanho máximo para o único realce ou único pilar resultante da união. Caso o volume sorteado para mutação esteja na posição inicial ou no



fim do nível de produção, então neste caso o volume será unido apenas ao outro volume adjacente a ele, se o critério de tamanho máximo de pilar ou de realce for cumprido.

De forma oposta, a mutação *split* é do tipo separação, ou seja, irá separar um único volume sorteado em três. Essa mutação irá ocorrer somente se o volume sorteado tiver dimensão suficiente para produzir três volumes dentro do tamanho mínimo estabelecido para realces e pilares. Caso o volume sorteado seja o de um realce, a mutação irá resultar em um volume de realce em um extremo, um de pilar ao centro e outro de realce no outro extremo. Por sua vez, caso o volume sorteado seja de um pilar, a mutação irá resultar em um volume de pilar em um extremo, realce no centro e pilar no outro extremo. A Figura 27 a seguir demonstra os quatro tipos diferentes de mutações previstas no algoritmo.

Figura 27 - Tipos de mutações previstas no algoritmo genético para os pilares e realces.



Fonte: Autoria própria.

### 3.2.5.3 Função *fitness*

A função *fitness* do algoritmo genético é a própria função benefício, anteriormente descrita. Sendo no caso do algoritmo genético apenas utilizada uma função de transferência para cálculo do valor quando requisitado pelo algoritmo.

Nesta mesma função também é armazenado, em formato de planilha, um histórico dos valores *fitness* obtidos. Sendo automaticamente os valores gravados e incluídos em sequência em um arquivo formato ASCII (.txt). As informações armazenadas para cada cenário são: o próprio *fitness (profit)*, porcentagem de falha de pilares (PFP), recuperação global de lavra, diluição com o *hangingwall*, volume de diluição, volume de elementos com falha em pilares, distância média de transporte, toneladas e teor médio do cenário.

### 3.2.5.4 Função de otimização

A função de otimização propriamente dita é a função que operacionaliza o algoritmo genético. Inicialmente a função utiliza-se das variáveis de entrada fornecidas pelo usuário via interface gráfica do programa para iniciar as funções específicas.

A primeira etapa do algoritmo consiste na criação de uma população de cenários randômicos, conforme o tamanho da população desejada inserida pelo usuário. Após a criação de cada cenário, a função *fitness* calcula o resultado do cenário, o que inclui a modelagem numérica do cenário e o cálculo do retorno econômico, e o armazena na memória.

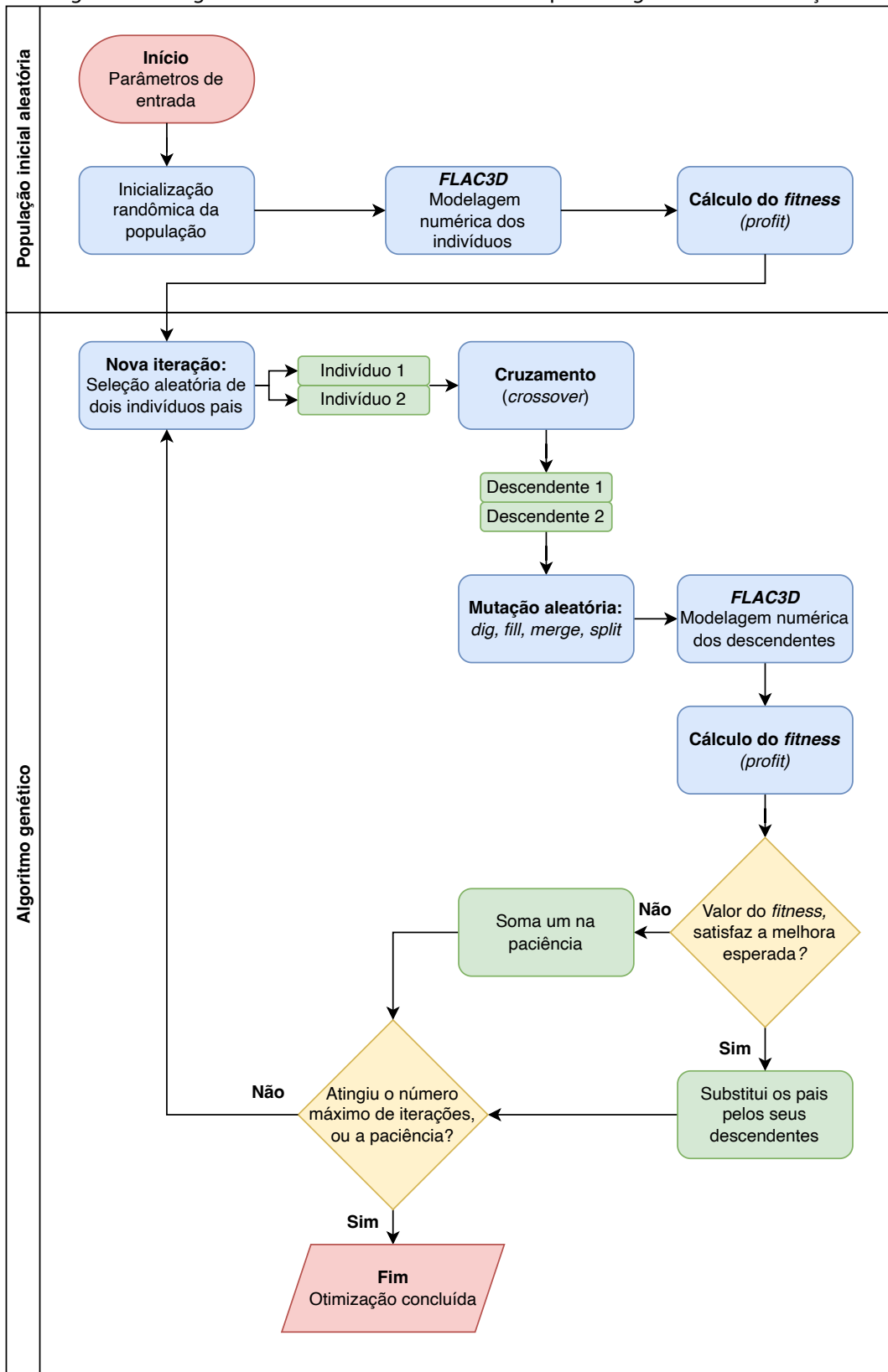
Ao fim da inicialização da população inicial, inicia a fase de iteração, onde dois cenários aleatórios da população inicial são utilizados como pais para o cruzamento e geração de descendentes. Os descendentes produzidos ainda sofrem mutação, conforme os tipos definidos previamente e respeitando a probabilidade de mutação definida, por padrão essa probabilidade é 1, ou seja, sempre ocorrerá mutação. Após a mutação a função *fitness* é calculada para os novos descendentes e caso os valores obtidos forem melhores que os valores dos pais, estes serão substituídos da população pelos seus descendentes. Essa definição de sempre realizar mutação e de substituir os pais, se os descendentes forem melhores, preserva a diversidade dos cenários.

O processo de iteração segue e em cada nova geração de descendentes o valor máximo para a função *fitness* é verificado. A finalização do algoritmo genético pode se dar de duas formas, sendo uma delas quando se atinge o número máximo de iterações

definidas. A segunda possibilidade de finalização se dá ao se atingir um número máximo de iterações sem que haja uma melhora do resultado. Esse parâmetro do número máximo de iterações sem melhora é definido como paciência, juntamente com o valor da tolerância quanto a melhora do resultado a cada iteração, sendo ambos escolhas do usuário.

Ao longo de cada iteração o maior valor de *fitness* obtido, bem como a paciência (estagnação ou número de iterações sem melhora), são mostrados em tela para o usuário. Ao final um gráfico *boxplot* gerado mostrará a cada geração os valores de *fitness* obtidos. A Figura 28 a seguir demonstra o diagrama de blocos das etapas do algoritmo de otimização.

Figura 28 – Diagrama de blocos demonstrando as etapas do algoritmo de otimização.



Fonte: Autoria própria.

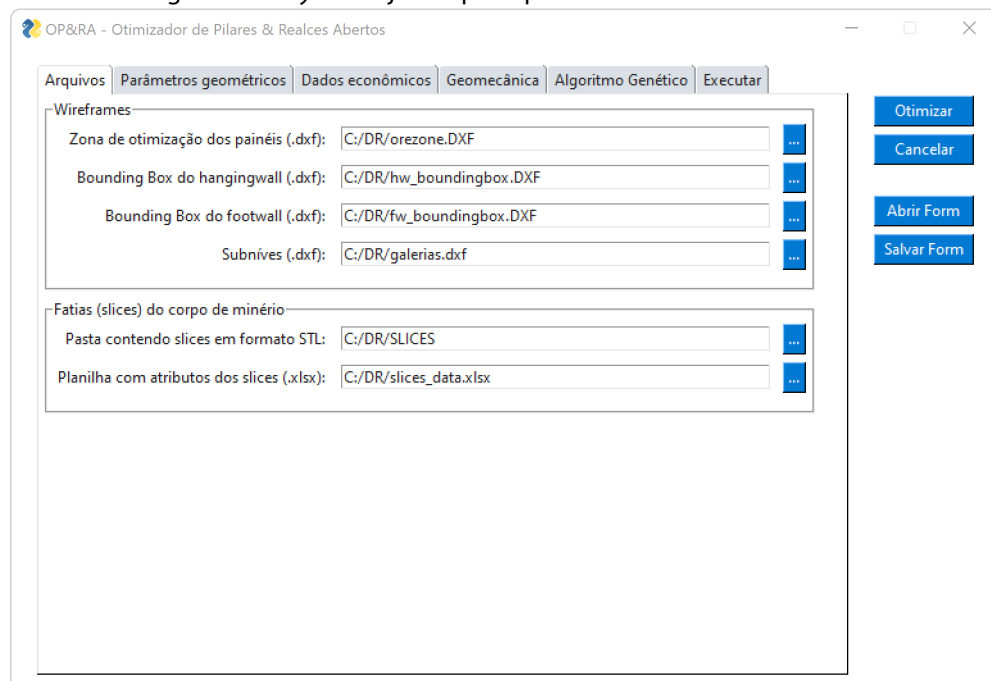
### 3.2.6 Interface gráfica do usuário

A interface gráfica do usuário, do inglês *Graphical User Interface* (GUI), para o programa foi produzida utilizando-se o pacote de biblioteca *Python PySimpleGUI* (PYSIMPLEGUI, 2022). Este pacote é baseado no pacote mais avançado Tkinter (TKINTER, 2022), que acompanha a distribuição oficial do interpretador Python, tornando-se mais simples a implementação de interfaces gráficas do usuário. Este pacote permite a construção de janelas complexas contendo todos os tipos de elementos, botões, *sliders*, *checkbox*, abas, caixas de textos etc.

#### 3.2.6.1 Janela principal e janela de progresso

Para o programa desenvolvido utiliza-se uma janela principal e nesta janela os elementos estão dispostos em seis abas distintas, conforme a seção do programa. As abas são as seguintes: “Arquivos”, “Parâmetros geométricos”, “Dados econômicos”, “Geomecânica”, “Algoritmo Genético” e “Executar”. Na janela do programa, são apresentados quatro botões sempre visíveis, independente da aba selecionada, sendo eles: “Otimizar”, “Cancelar”, “Abrir Form” e “Salvar Form”. A Figura 29 a seguir demonstra a interface da janela principal do programa.

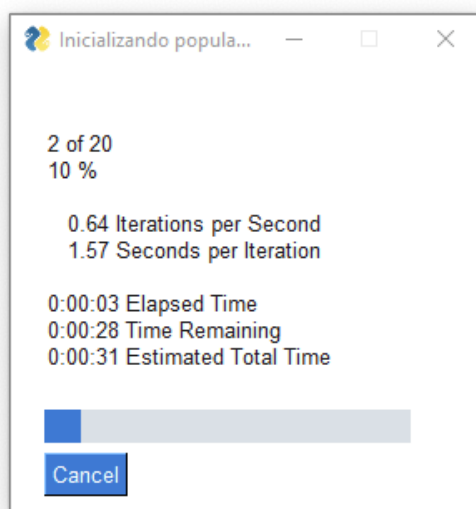
Figura 29 – Layout da janela principal de interface com o usuário.



Fonte: Autoria própria.

Durante a etapa de inicialização da população uma nova janela de progresso permanece aberta até a finalização da etapa. Quando inicia a etapa de iteração outra janela com barra de progresso semelhante se abre permanecendo aberta até a finalização do algoritmo. Na janela de progresso, algumas informações úteis são demonstradas como: número de iterações realizadas com porcentagem concluída, número de iterações por segundo, tempo transcorrido desde o início da etapa, tempo restante e tempo total estimado. Nesta janela, um único botão de cancelar é demonstrado. A Figura 30 abaixo demonstra a janela de progresso com a informação do número de iterações por segundo, tempo de execução, tempo restante e tempo total estimado.

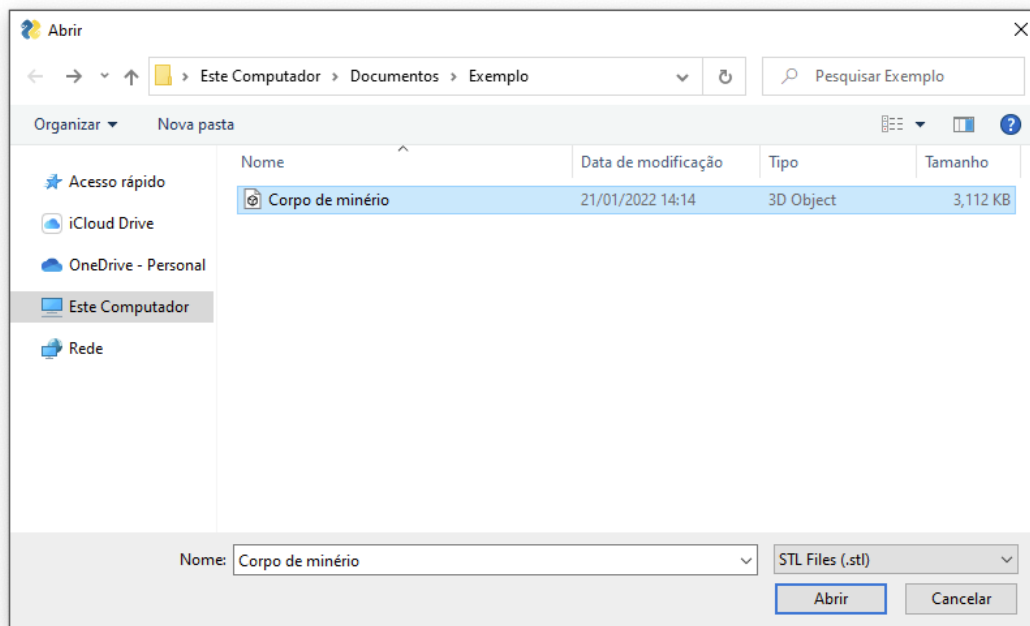
Figura 30 - Janela de progresso com informações como número de iterações por segundo, tempo de execução, tempo restante e tempo total estimado.



Fonte: Autoria própria.

Por sua vez, para a seleção de arquivos de entrada solicitados pelo programa, uma janela de navegação do sistema operacional é aberta e os arquivos mostrados são filtrados pelo formato da extensão esperada. Na Figura 31 é possível visualizar como exemplo a janela para seleção do arquivo *wireframe* que representa o corpo de minério.

Figura 31 – *Layout* da janela do sistema operacional para seleção de arquivos de entrada, os arquivos são filtrados pelo formato de arquivo esperado.



Fonte: Autoria própria.

### 3.2.6.2 Abas do programa

As abas (*tabs*) são seções do programa que permitem diferentes *layouts* que vão sendo alternados dentro da mesma janela. Cada uma das abas agrupa uma série de informações relacionadas com a seção que representa.

A primeira aba mais a esquerda da janela é a aba “Arquivos”, esta aba é a que por padrão está selecionada quando o programa inicia. Nesta aba, dois quadros (*frames*) são apresentados, o primeiro mais acima é denominado “*Wireframes*” e trás quatro campos de entrada para arquivos a serem carregados. O primeiro campo é para seleção da: “Zona de otimização dos painéis (.dxf)”, o segundo é: “*Bounding Box* do *hangingwall* (.dxf)”, o terceiro é: “*Bounding Box* do *footwall* (.dxf)” e o quarto é: “Subníveis (.dxf)”. A direita destes campos estão botões com o escrito “...” em cada um deles, que quando são apertados abrem uma janela de navegação de pastas, para seleção do arquivo solicitado. Por sua vez, o quadro mais abaixo é denominado: “*Fatias (slices)* do corpo de minério”. Neste quadro dois campos de entrada são solicitados, um deles solicita: “Pasta contendo *slices* em formato STL” e o outro: “Planilha com atributos dos *slices* (.xlsx)”.

A segunda aba da janela é a aba “Parâmetros geométricos”, esta aba é subdividida em três quadros diferentes: “Pilares”, “Realces” e “Subníveis”. O primeiro quadro possui duas

caixas de texto para preenchimento da “Largura mínima dos pilares (m)” e “Largura máxima dos pilares (m)”. O segundo quadro também possui duas caixas de texto: “Largura mínima dos realces (m)” e “Largura máxima dos realces (m)”. Por fim o último quadro, referente aos subníveis, também solicita o preenchimento de dois campos, o de “Quantidade” e “Número de *slices* por subníveis (separado por vírgulas)”. A Figura 32 abaixo permite visualizar a aba em questão e os campos para preenchimento.

Figura 32 – Layout da aba de “Parâmetros geométricos”.

Fonte: Autoria própria.

A terceira aba é a de “Dados econômicos”, nesta aba as informações solicitadas são: custo de lavra (US\$/t), custo de transporte (US\$/t/DMT), custo de processo (US\$/t), custo G&A (US\$/t), custo de refino (US\$/g metal), royalty (%), preço do ouro (US\$/g), recuperação global de lavra (%) e recuperação planta de processo (%). Um segundo quadro mais abaixo, intitulado “Teor de corte” permite que este seja calculado rapidamente, para verificação, sendo necessário apenas inserir a DMT e pressionar o botão “Calcular”. A Figura 33 a seguir mostra a respectiva aba com os as informações solicitadas e opção de cálculo do teor de corte.



Figura 33 – Layout da aba "Dados econômicos".

OP&RA - Otimizador de Pilares & Realces Abertos

Arquivos | Parâmetros geométricos | **Dados econômicos** | Geomecânica | Algoritmo Genético | Executar

Dados econômicos

Custo de lavra (US\$/ton)	16.5
Custo de transporte (US\$/ton/DMT)	0.00014
Custo de processo (US\$/ton)	18.6
Custo G&A (US\$/ton)	18.1
Custo de Refino (US\$/g metal)	0.5
Royalty (%)	1.5
Preço do Ouro (US\$/g)	55
Recuperação de lavra nos realces (%)	90
Recuperação Planta de Processo (%)	93

Teor de corte

DMT (m)	3500
Teor de corte calculado (g/t):	1.18

Otimizar  
Cancelar  
Abrir Form  
Salvar Form

Fonte: Autoria própria.

A quarta aba é a de “Geomecânica” e ao ingressar nela quatro abas extras surgem, são elas: “FLAC3D”, “Propriedades”, “Tensões *in situ*” e “Desempenho”. Na primeira aba são solicitadas as informações referentes ao FLAC3D, sendo a janela subdividida em três quadros (Figura 34). O primeiro quadro refere-se aos parâmetros do software, como o diretório de instalação do programa onde se encontra o arquivo executável (.exe), o somatório de forças desbalanceadas (*unbalanced forces*) a ser considerado para finalização do modelo numérico, a configuração do modo determinístico, do modo de escavação gradual (*relax excavation*) e do modo Mohr-Coloumb *flag-brittle*. O segundo quadro é o “*Bounding Box*” onde são inseridas as coordenadas x, y, z mínimas e máximas e o número de elementos em x, y, z. Por fim, o último quadro é referente a configuração do método *Octree*, onde são inseridas a subdivisão de segmentos, tamanho mínimo em metros e a extrapolação além do limite de geometria também em metros.

Figura 34 – Layout da aba principal “Geomecânica” e da aba secundária “FLAC3D”.

The screenshot shows the 'OP&RA - Otimizador de Pilares & Realces Abertos' window. The main menu includes 'Arquivos', 'Parâmetros geométricos', 'Dados econômicos', 'Geomecânica', 'Algoritmo Genético', and 'Executar'. The 'Geomecânica' tab is selected, and the 'FLAC3D' sub-tab is active. The interface is divided into several sections:

- Parâmetros:**
  - Arquivo executável do FLAC3D (.exe): C:/Program Files/Itasca/FLAC3D700/exe64/fl ...
  - Somatório de forças desbalanceadas (default = 1e-4): 1e-3
  - Modo determinístico:
  - Escavação gradual - relax excavation:
  - Mohr-Coloumb flag-brittle:
- Bounding Box:**

	min	max		
Coordenadas X:	-46	376	Nº de elementos em X:	13
Coordenadas Y:	-26	170	Nº de elementos em Y:	6
Coordenadas Z:	-70	126	Nº de elementos em Z:	6
- Octree:**
  - Subdivisão de segmentos: 2
  - Tamanho mín. (m): 1.9
  - Extrapolação do limite (m): 3

On the right side of the window, there are four buttons: 'Otimizar', 'Cancelar', 'Abrir Form', and 'Salvar Form'.

Fonte: Autoria própria.

Ainda na aba principal “Geomecânica”, a segunda aba é a de “Propriedades” (Figura 35). Nesta aba, três quadros na porção superior, permitem a definição das propriedades geomecânica para cada um dos domínios principais: “Orezone”, “Hangingwall” e “Footwall”. Na porção inferior mais três quadros de preenchimento opcional são fornecidos, sendo eles denominados: “Subdomínio 1”, “Subdomínio 2” e “Subdomínio 3”. Estes subdomínios podem ser utilizados a critério do usuário, caso queiram definir subdomínios geomecânicos com propriedades diferentes, dentro dos domínios principais já definidos. Para sua utilização, basta o usuário marcar a caixa de seleção “Usar subdomínio”, que automaticamente os campos para preenchimento serão liberados, sendo necessário que usuário indique a localização do arquivo de *wireframes* que representa a geometria do subdomínio. Por limitação do tamanho da janela, o número foi limitado a três subdomínios, porém, mais subdomínios poderiam ser permitidos, com pequenas alterações no código do programa, caso necessário.

As propriedades geomecânicas a serem inseridas nos domínios e subdomínios, quando utilizados, são definidas pelo critério de ruptura de Mohr-Coulomb, ou seja, coesão

(MPa) e ângulo de atrito ( $^{\circ}$ ), além da densidade  $t/m^3$  e as propriedades elásticas módulo *Bulk* (MPa) e módulo *Shear* (MPa).

Figura 35 - *Layout* da aba principal “Geomecânica” e da aba secundária “Propriedades”.

The screenshot shows the 'OP&RA - Otimizador de Pilares & Realces Abertos' window. The 'Geomecânica' tab is selected, and the 'Propriedades' sub-tab is active. The interface is organized into a grid of input fields for material properties.

Zone	Densidade (t/m3)	K - Bulk (GPa)	G - Shear (GPa)	Coesão (MPa)	Ângulo de atrito ( $^{\circ}$ )
Orezone	2.78	19.9	13.13	1.35	46.9
Hangingwall	2.75	13.8	8.9	1.1	40
Footwall	2.63	25.7	18.1	2	47

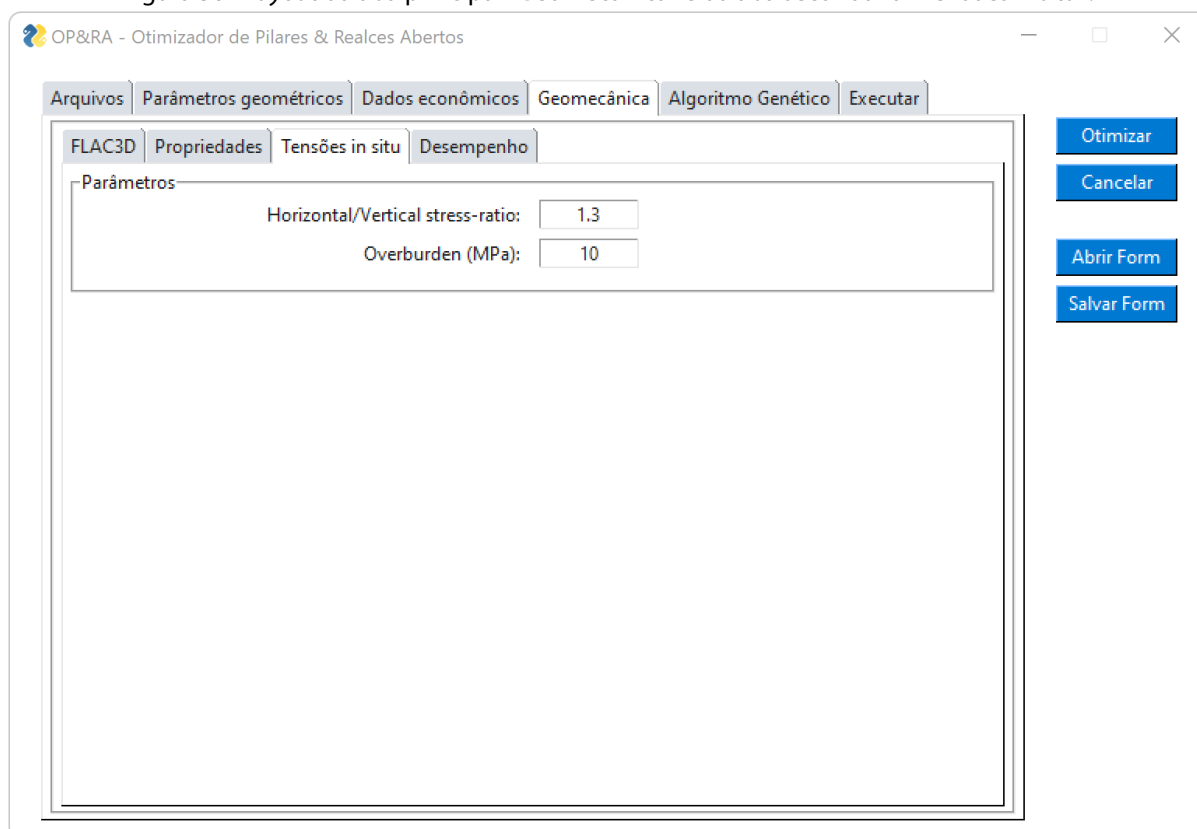
  

Subdomínio	Usar subdomínio	Wireframe (.dxf)	Nome (sigla)	Densidade (t/m3)	K - Bulk (GPa)	G - Shear (GPa)	Coesão (MPa)	Ângulo de atrito ( $^{\circ}$ )
Subdomínio 1	<input checked="" type="checkbox"/>	C:/DR	dom1	2.5	12	8.5	1	37
Subdomínio 2	<input type="checkbox"/>							
Subdomínio 3	<input type="checkbox"/>							

Fonte: Autoria própria.

Ainda na aba principal “Geomecânica”, a terceira aba é a de “Tensões *in situ*” (Figura 36). Nesta aba deve ser inserida a relação entre a tensão horizontal e vertical ( $k$ ) e a tensão equivalente do *overburden* (MPa). O *overburden* representa a tensão vertical, produto da cobertura de rocha acima da porção superior da *Bounding Box*, que foi previamente definida.

Figura 36 - Layout da aba principal “Geomecânica” e da aba secundária “Tensões *in situ*”.



Fonte: Autoria própria.

Ainda na aba principal “Geomecânica”, a última aba é a de “Desempenho”. Nesta aba temos dois quadros um relacionado ao *hangingwall* e o outro aos pilares (Figura 37). Com relação ao *hangingwall* inicialmente é solicitado o critério de diluição definido, sendo necessário informar a partir de que valor inferior de  $\sigma_3$  é que o elemento do *hangingwall* deve ser considerado como em situação de ruptura e conseqüentemente passível de ocasionar diluição. Também neste mesmo quadro é solicitada a diluição esperada para o método de lavra, o custo de suporte secundário do *hangingwall*, em US\$/m<sup>3</sup> de diluição e a taxa de eficiência deste suporte secundário em porcentagem.

Por fim, o quadro relacionado aos pilares solicita o critério de falha dos pilares a ser utilizado, a porcentagem máxima aceitável de falha em pilares (PAF) e o custo de reforço em pilares com falha em US\$/m<sup>3</sup> de elemento de pilar com falha. Ainda que haja uma caixa de seleção disponível, o único critério de falha de pilares implementado é o do SSR (*Strength-Stress Ratio*) disponível no FLAC3D, e descrito no item 2.5.4.1. Outros critérios de falha de pilares podem ser implementados ao programa em versões futuras, caso desejado.

Figura 37 - Layout da aba principal “Geomecânica” e da aba secundária “Desempenho”.

The screenshot displays the OP&RA software interface. The main window has a title bar 'OP&RA - Otimizador de Pilares & Realces Abertos'. Below the title bar is a menu bar with options: 'Arquivos', 'Parâmetros geométricos', 'Dados econômicos', 'Geomecânica', 'Algoritmo Genético', and 'Executar'. The 'Geomecânica' tab is active, and within it, the 'Desempenho' sub-tab is selected. The interface is divided into two main sections: 'Hanging wall' and 'Pilares'. The 'Hanging wall' section contains four input fields: 'Critério de diluição, s3 menor que (MPa):' with value 0.1, 'Diluição esperada (%):' with value 10, 'Custo de suporte (US\$/m3 de diluição):' with value 1, and 'Eficiência do suporte do HW (%):' with value 90. The 'Pilares' section contains three input fields: 'Critério de falha:' with a dropdown menu showing 'SSR<1.01', 'Porcentagem máx. aceitável de falha em pilares (%):' with value 5, and 'Custo de suporte (US\$/m3 de pilar com falha):' with value 0.9. On the right side of the window, there are four blue buttons: 'Otimizar', 'Cancelar', 'Abrir Form', and 'Salvar Form'.

Fonte: Autoria própria.

A quinta aba é a do “Algoritmo Genético”, nesta aba as informações solicitadas estão todas contidas no mesmo quadro denominado “Parâmetros” (Figura 38). O primeiro campo solicitado ao usuário é a quantidade de indivíduos a serem gerados randomicamente para a população inicial. O número máximo de iterações precisa ser informado pelo usuário, além da “Paciência”, que define o número de iterações aceitáveis sem melhora no resultado, conforme o fator de melhora mínimo definido. A probabilidade de mutações acontecerem também deve ser informada.

Figura 38 - Layout da aba "Algoritmo Genético".

OP&RA - Otimizador de Pilares & Realces Abertos

Arquivos | Parâmetros geométricos | Dados econômicos | Geomecânica | Algoritmo Genético | Executar

Parâmetros

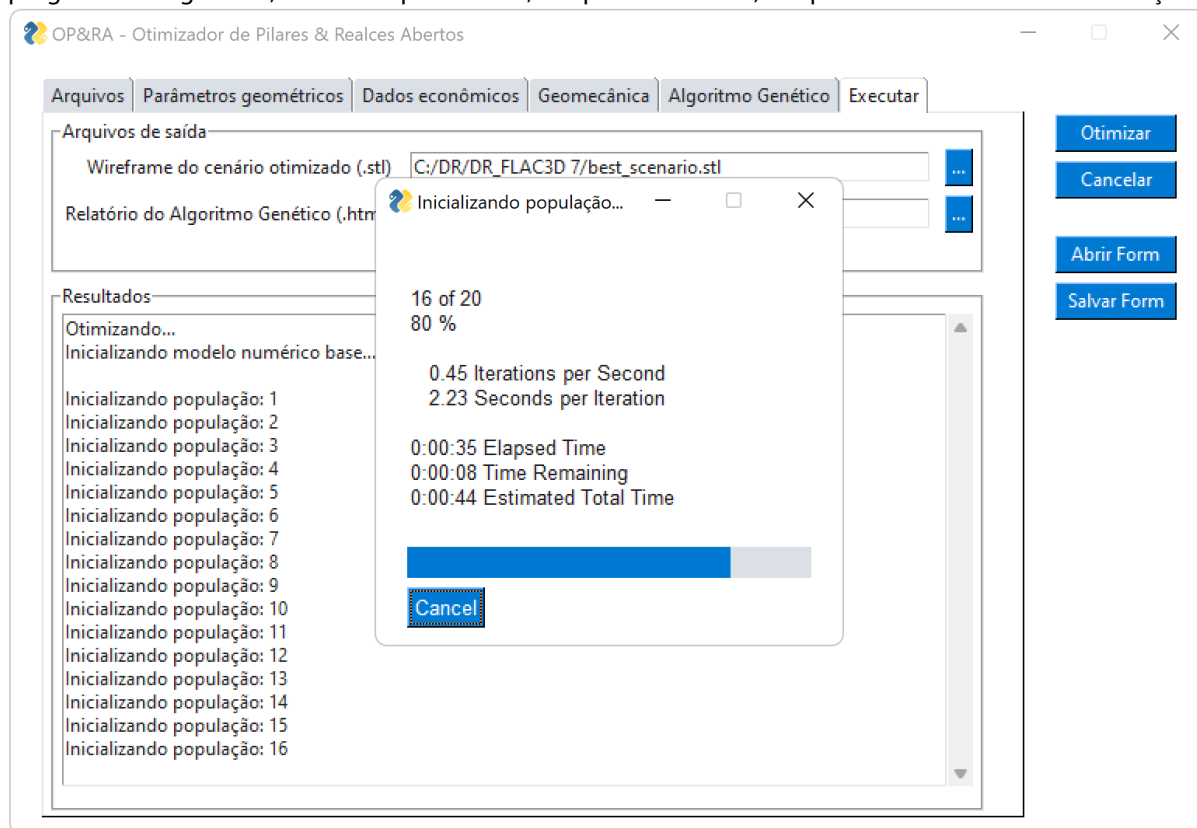
População inicial randômica (0-200):	<input type="text" value="20"/>
Número de iterações máximas (0-1000):	<input type="text" value="200"/>
Paciência (0-100):	<input type="text" value="40"/>
Probabilidade de mutação (0-1):	<input type="text" value="1"/>
Fator de melhora (0.1):	<input type="text" value="0.1"/>

Otimizar  
Cancelar  
Abrir Form  
Salvar Form

Fonte: Autoria própria.

A sexta aba é a “Executar”, ela possui dois quadros sendo o primeiro relacionado a identificação dos arquivos de saída da otimização, o *wireframes* do cenário ótimo em formato “.STL” e o relatório de saída da evolução do algoritmo genético, em formato “.html”. O segundo quadro por sua vez é o quadro que mostra em tempo real os resultados da execução do algoritmo, que são mostrados quando o botão “Otimizar” é pressionado, conforme mostra a Figura 39.

Figura 39 - *Layout* da aba "Executar", com a janela *popup* em primeiro plano mostrando os detalhes do progresso do algoritmo, com o tempo restante, tempo transcorrido, tempo estimado e número de iterações.



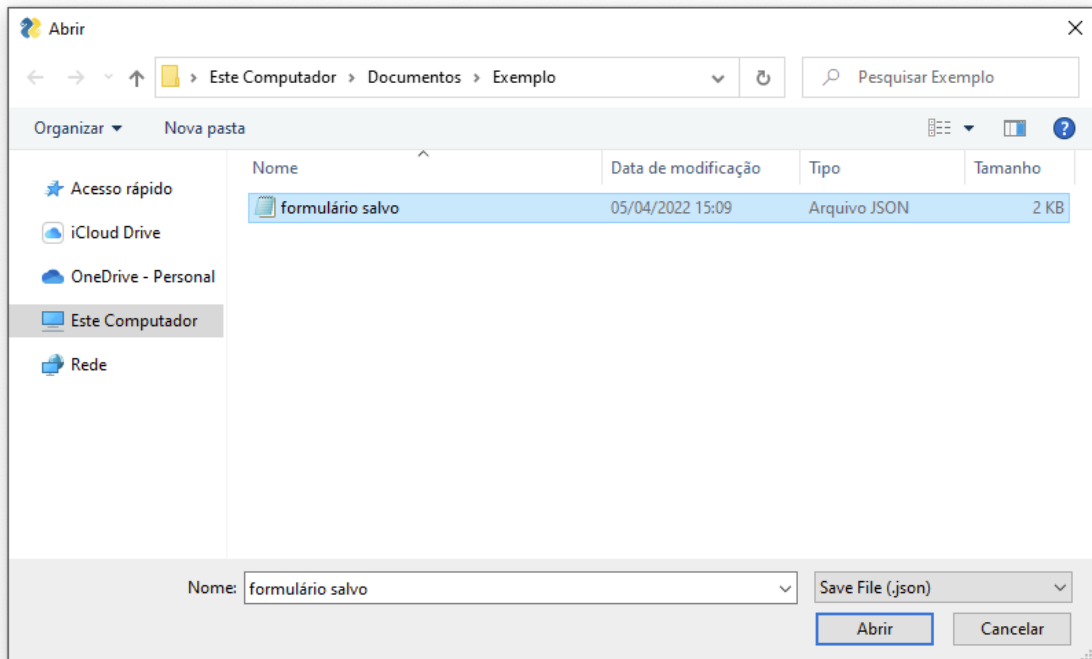
Fonte: Autoria própria.

### 3.2.6.3 Abrir e salvar formulários

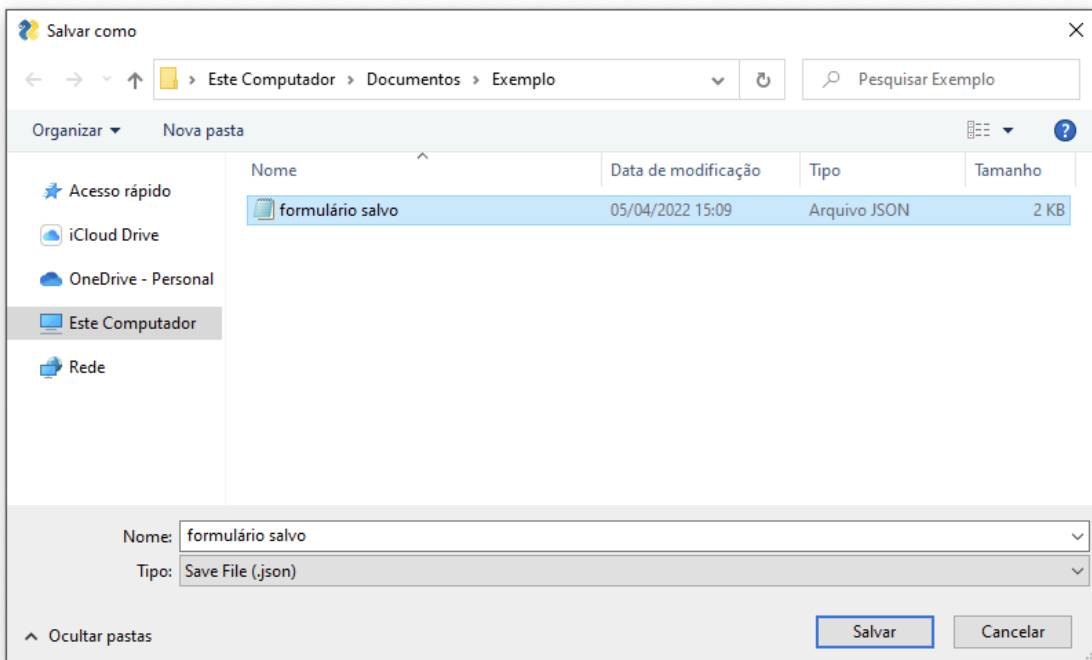
Devido a grande quantidade de campos e arquivos a serem preenchidos e selecionados para execução do algoritmo, se faz extremamente necessário a possibilidade de abrir e salvar arquivos contendo formulários preenchidos. Isso permite manter o histórico das decisões e configurações utilizadas para a otimização, além de facilitar que novas otimizações sejam realizadas com algumas variações rápidas nos parâmetros.

Para a implementação dessa função no programa foi utilizado o próprio pacote PySimpleGUI onde este possui uma função integrada para abrir e salvar os arquivos de configuração dos campos preenchidos em formato *JavaScript Object Notation (JSON)*. O JSON é um formato compacto, independente, e de padrão aberto para troca de dados simples, usando texto legível por humanos para armazenar e transmitir objetos de dados que consistem em pares de valor de atributo e matrizes.

Na Figura 40 é possível visualizar o *layout* da janela para abertura de um arquivo de formulário salvo em formato JSON. De forma análoga é possível visualizar na Figura 41 abaixo o *layout* da janela de salvamento de arquivo de configuração.

Figura 40 – *Layout* da janela para abrir arquivo salvo com formulário preenchido, em formato JSON.

Fonte: Autoria própria.

Figura 41 - *Layout* da janela para salvar arquivo com formulário preenchido, em formato JSON.

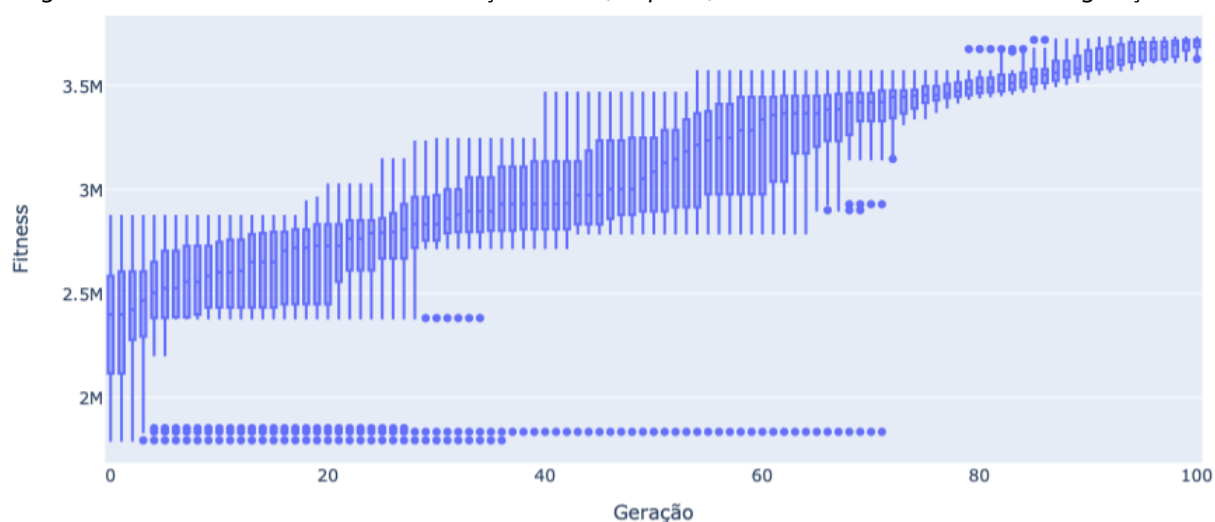
Fonte: Autoria própria.



### 3.2.7 Resultados e arquivos de saída

Após a finalização do algoritmo de otimização alguns arquivos são gerados. Os arquivos se resumem a relatórios, mapas com vistas isométricas (*plots*) e arquivos de *wireframes* com as geometrias otimizadas. O primeiro arquivo de saída que é mostrado na tela para usuário logo após a finalização da otimização é o relatório evolutivo (*evolution report*) que mostra a evolução do algoritmo genético a cada geração. O relatório evolutivo gerado é um gráfico do tipo diagrama de caixa (*boxplot*) contendo a mediana, os quartis inferiores e superiores, além dos valores *outliers* de todos os indivíduos da população a cada nova geração. Este gráfico foi produzido pela biblioteca Plotly para *Python* (PLOTLY, 2022). Um exemplo de relatório evolutivo pode ser visualizado na Figura 42.

Figura 42 – Exemplo de gráfico *boxplot* mostrando o relatório evolutivo a cada nova geração do algoritmo genético. O eixo vertical é o valor da função *fitness* (*net profit*) e o eixo horizontal o número de gerações.



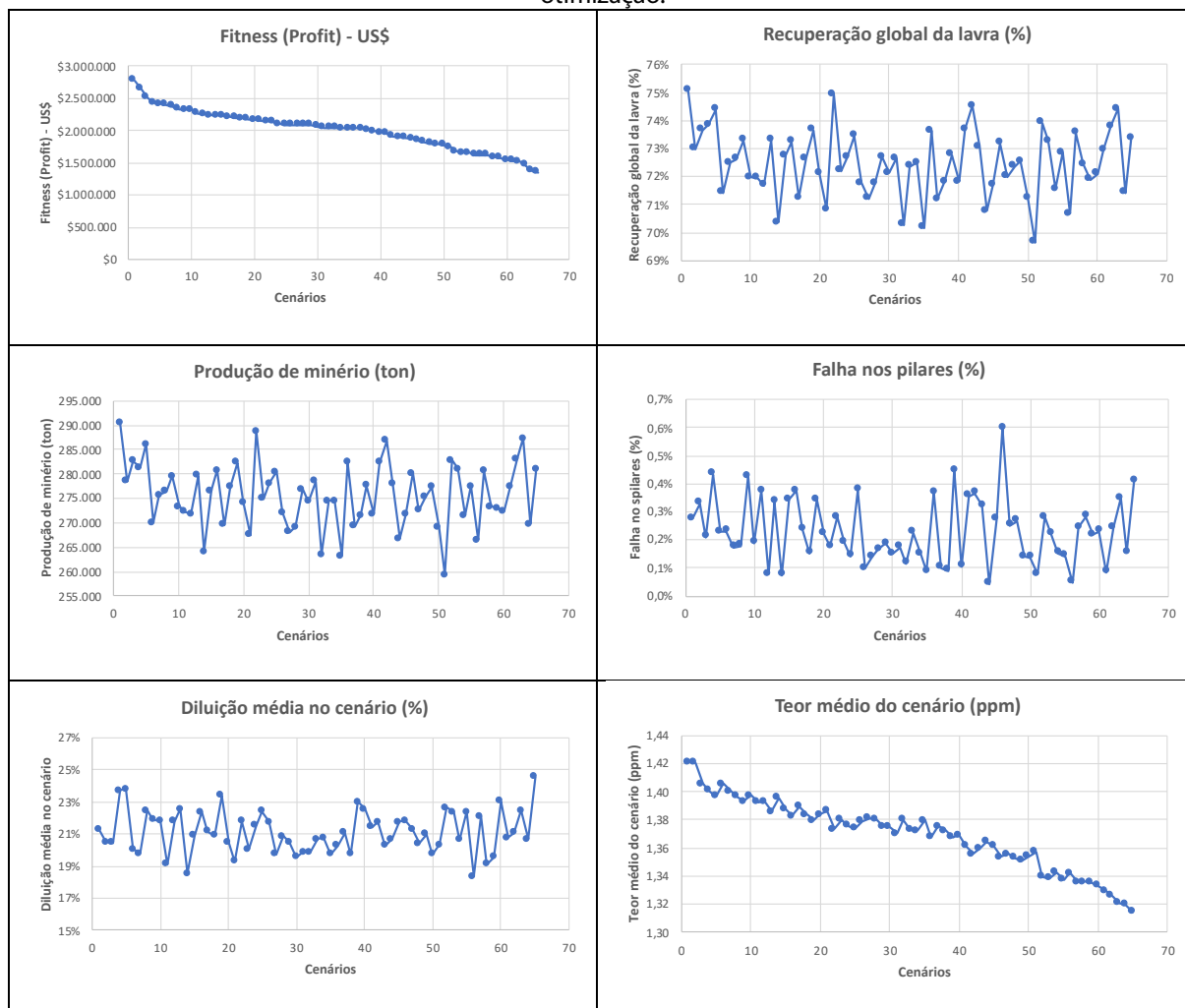
Fonte: Autoria própria.

Outro relatório gerado após a finalização do algoritmo é um arquivo em formato de planilha, contendo os atributos detalhados de cada um dos cenários avaliados. Essa planilha inclui os cenários inicialmente gerados randomicamente, além dos cenários gerados pelas iterações do algoritmo genético, incluindo o cenário tido como ótimo.

Os dados armazenados para cada cenário são: produção de minério, teor médio, volume de falha nos pilares, volume de diluição, porcentagem de falha nos pilares, porcentagem de diluição, recuperação global da lavra e o retorno econômico (*fitness*). A Figura 43 a seguir reúne estas informações em seis gráficos de exemplo produzidos a partir

da planilha com os atributos dos cenários produzidos a cada nova geração. Nos gráficos mostrados os cenários foram ordenados do melhor para o pior *fitness* obtido.

Figura 43 – Exemplo de gráficos dos atributos dos cenários produzidos a cada geração após fim da otimização.



Fonte: Autoria própria.

Os resultados de saída do FLAC3D® são gerados automaticamente após a execução do modelo. Os resultados incluem um gráfico do histórico do somatório de forças desbalanceadas (*unbalanced forces*) ao longo das etapas de escavação do modelo. Além desta, uma plotagem isométrica contendo a geometria analisada, outra plotagem isométrica mostrando o desempenho dos pilares, por meio das isosuperfícies de SSR, e uma última vista isométrica representando o potencial de diluição, por meio da isosuperfície de tensão principal menor, conforme valor definido.

### 3.3 Modelagem numérica automatizada

O *software* FLAC3D® é bastante utilizado na modelagem numérica de diversos problemas geotécnicos, permitindo verificar o comportamento esperado do maciços rochosos, após a escavação. O FLAC3D é um programa tridimensional, explícito, de diferenças finitas. O material é introduzido como elemento de zona caracterizado por um *grid* tridimensional que é construído para atender às necessidades do usuário. A lei linear e não linear de tensão/deformação é dada para permitir que cada elemento se comporte respondendo à condição de contorno (ITASCA, 2005).

Para que o algoritmo proposto seja capaz de encontrar uma solução ótima para o arranjo de realces e pilares, a partir de vários cenários avaliados e otimizados pelo algoritmo genético, se faz necessário que os modelos numéricos sejam executados de forma automatizada, sem necessidade de interferência humana. Sendo assim, códigos previamente escritos em linguagem de comandos e linguagem *FISH* do FLAC3D® são acionados sequencialmente, a partir da conexão *socket* com o console *Python*.

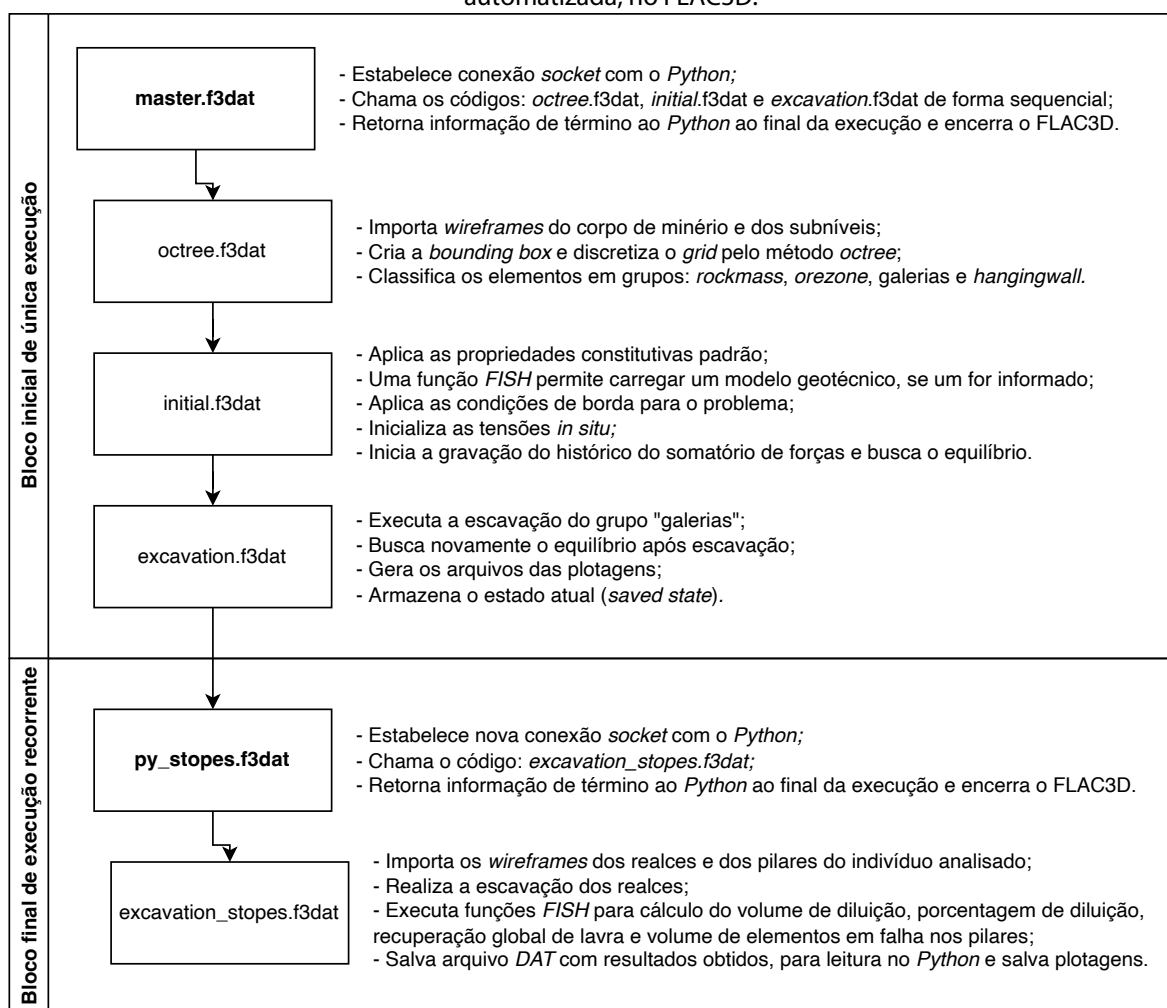
Considerando a lógica de análise do problema, arquivos separados de códigos foram escritos em formato *“.f3dat”*. Estes arquivos separados podem ser agrupados em dois blocos distintos, onde o primeiro bloco é executado uma única vez ao longo de todo algoritmo e o segundo bloco é executado de forma recorrente a cada avaliação de indivíduo do algoritmo genético.

Dentro do bloco inicial, a primeira etapa do problema consiste na importação dos arquivos de geometrias e discretização do *grid* de elementos dentro do corpo de minério (*orezone*), conforme definido pelo usuário, o que foi condensado em um mesmo *data file*. Para a etapa seguinte outro *data file* foi escrito, e neste é onde as propriedades constitutivas padrão são definidas, as condições de borda e as tensões *in situ* são atribuídas e o modelo é resolvido elasticamente para que ocorra o equilíbrio de tensões (sem nenhuma escavação). O terceiro arquivo de códigos é onde se realiza a escavação dos subníveis e um novo equilíbrio de forças é obtido. Este estado do FLAC3D® é então salvo (*saved state*) sendo posteriormente utilizado como ponto de partida quando os cenários gerados pelo algoritmo genético precisam ser avaliados.

Por sua vez o bloco final será invocado de forma recorrente ao longo da execução da otimização, carregando o *saved state* anterior e escavando o cenário de realces que aquele indivíduo representa. Após a escavação busca-se o equilíbrio de forças e funções *FISH* são

executadas automaticamente para avaliar o indivíduo quanto ao desempenho dos pilares, diluição do *hangingwall* e gerando as plotagens previamente definidas. A Figura 44 a seguir demonstra a conexão entre os blocos descritos e os diversos arquivos de códigos.

Figura 44 – Diagrama de blocos de códigos executados sequencialmente, para modelagem numérica automatizada, no FLAC3D.

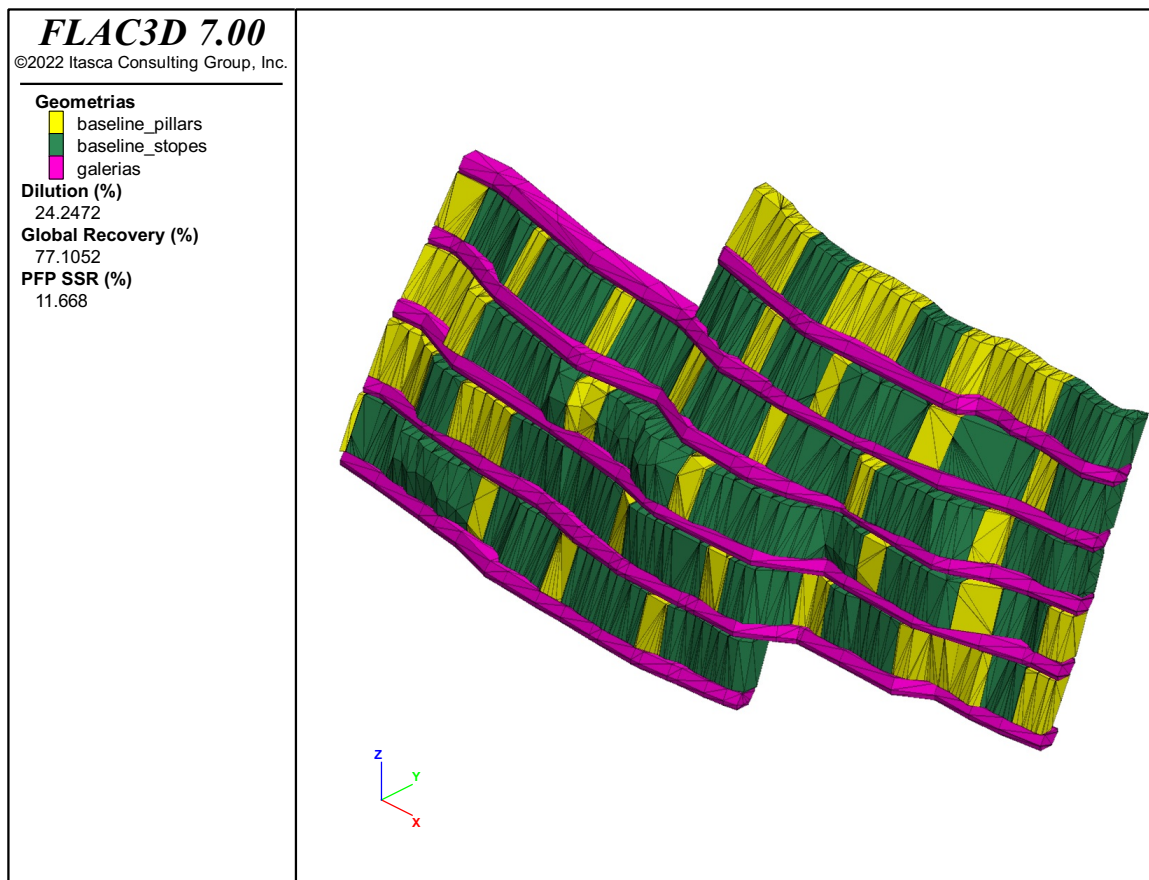


Fonte: Autoria própria

### 3.3.1 Importação de arquivos

A primeira parte do código automatizado que será executado no FLAC3D® está contida no arquivo *octree.f3dat*. Para cada cenário analisado, inicia pela importação dos arquivos de geometria preparados anteriormente. Os arquivos a serem importados são: zona de otimização dos painéis (.dxf), *Bounding Box* do *hangingwall* (.dxf), *Bounding Box* do *footwall* (.dxf) e os subníveis (.dxf) ao longo do corpo de minério. O arranjo dos realces e dos pilares, a ser analisado, também são importados, em formato STL. A Figura 45 a seguir mostra um cenário de exemplo com os arquivos de geometria importados.

Figura 45 - Exemplo de gráfico gerado automaticamente com a geometria do arranjo de realces e pilares otimizado, além dos subníveis fornecidos pelo usuário.

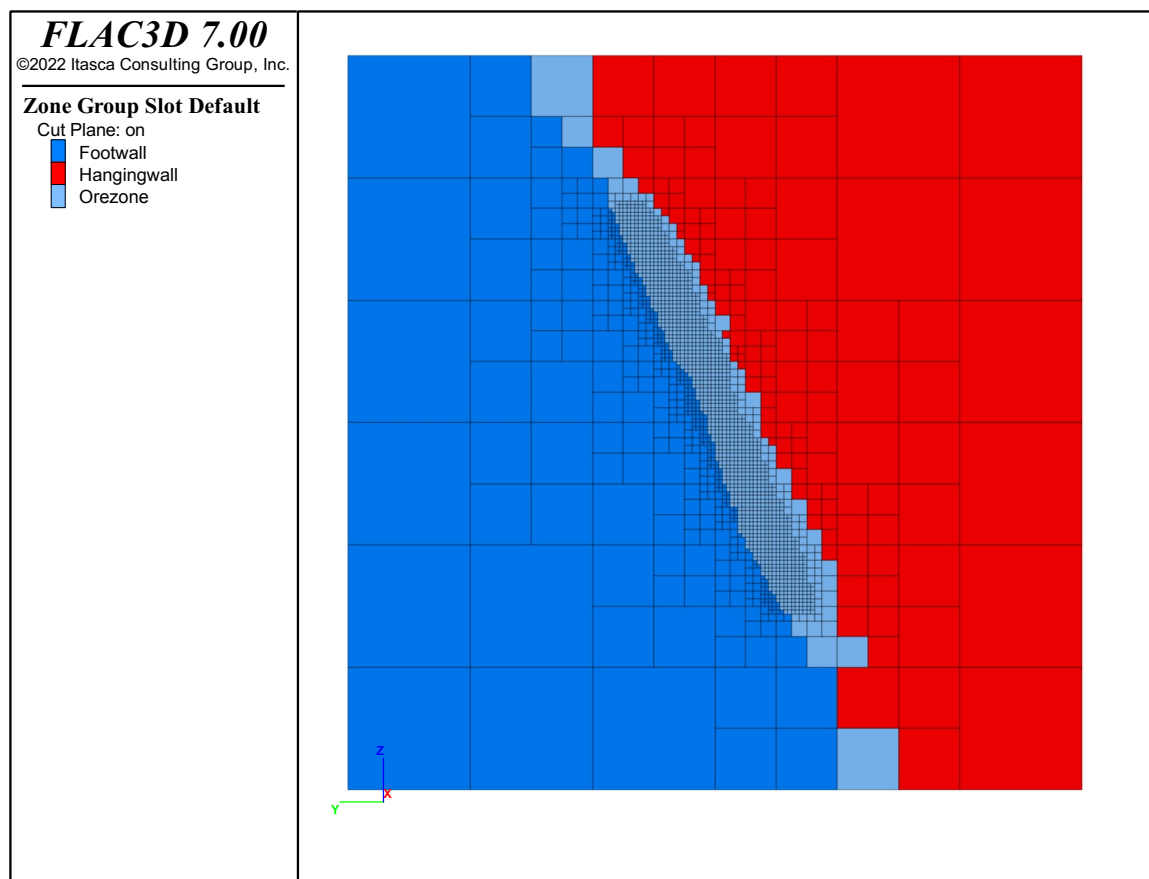


Fonte: Autoria própria.

### 3.3.2 Discretização pelo método *Octree*

Ainda no arquivo *octree.f3dat*, mas após a importação das geometrias, ocorre um adensamento da discretização nos locais de interesse, ou seja, dentro do volume das galerias, realces, pilares e próximo do contato entre essas superfícies, onde os elementos cúbicos gerados anteriormente são divididos sempre pela metade até atingirem a dimensão mínima definida, o que pode ser observado na Figura 46.

Figura 46 - Seção do modelo numérico, onde observa-se a discretização gradativa dos elementos nos locais de interesse.

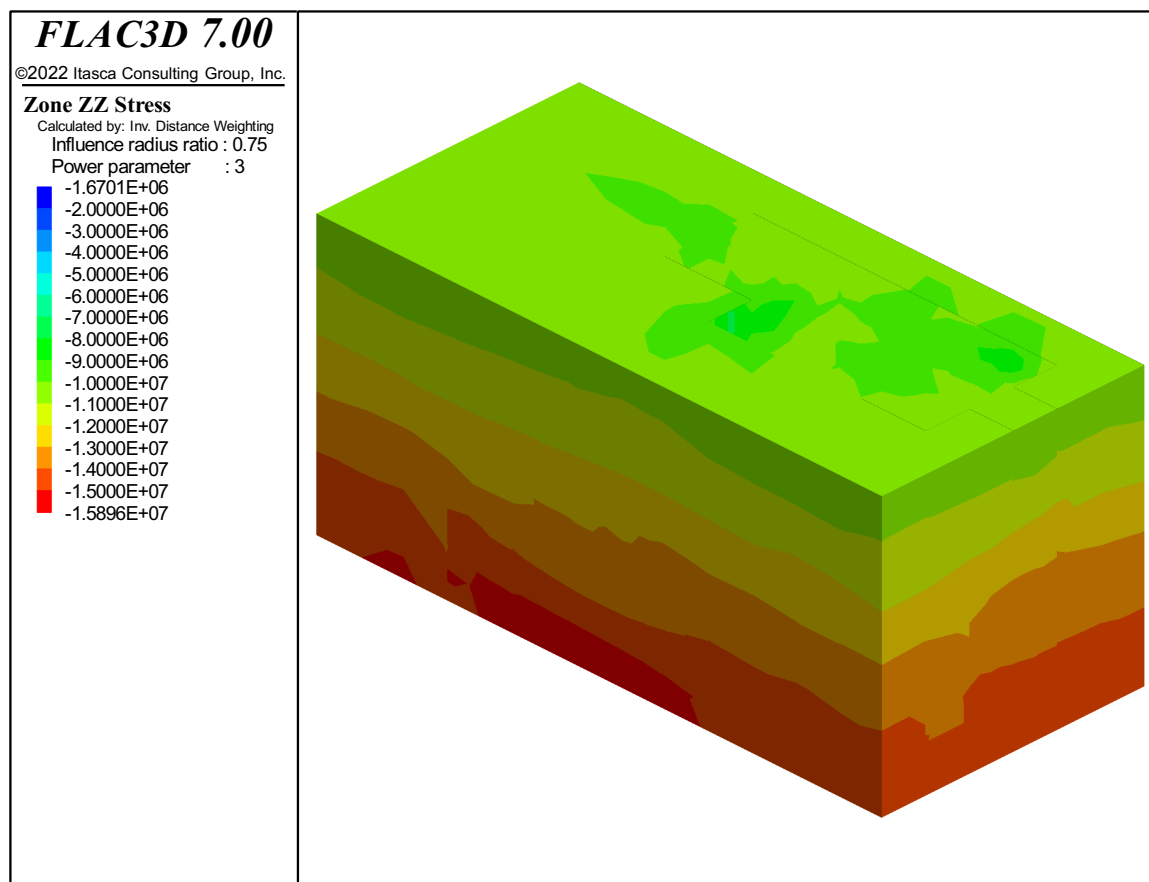


Fonte: Autoria própria.

### 3.3.3 Parâmetros de inicialização do modelo

A etapa seguinte do modelo é a inicialização dos parâmetros iniciais, como propriedades, modelo constitutivo, condições de borda e tensões *in situ*. O modelo é então executado, para que as forças atuantes encontrem o equilíbrio inicial, antes da realização de qualquer escavação. Por padrão do FLAC3D, o equilíbrio é atingido quando o somatório das forças internas (*unbalanced forces*) atinge o valor de  $1 \times 10^{-5}$ . Após atingir o equilíbrio, o estado do modelo (*saved state*) é salvo e a distribuição das tensões ao longo dos pontos do *grid* podem ser verificadas. Na Figura 47 há uma vista isométrica que mostra a distribuição das tensões verticais do modelo inicial.

Figura 47 – Vista isométrica do modelo numérico inicial, com a visualização das tensões verticais.

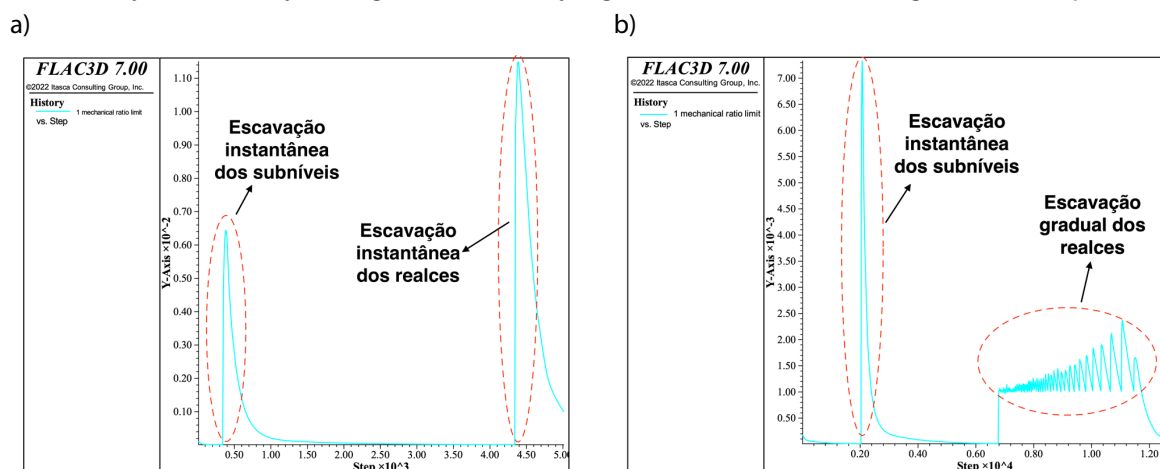


Fonte: Autoria própria.

O próximo passo do modelo é a realização da escavação das galerias de desenvolvimento e posteriormente dos realces. Considerando que o algoritmo proposto prevê o uso do critério de Mohr-Coloumb e que no FLAC3D, por padrão, se ocorrer falha de tração em uma zona, o valor atribuído para a resistência à tração nesta zona permanece constante, o uso do comando *flag-brittle* permite definir a resistência à tração como zero para essa zona, quando ocorrer falha de tração. Esta consideração permite simular o efeito de amolecimento instantâneo por tração e pode ser utilizada a critério do usuário quando ativado na aba Geomecânica>FLAC3D.

Outra configuração do programa permite realizar a escavação de forma gradual (*relax excavation*) ou de forma instantânea. Com uso da escavação gradual, os realces vão sendo escavados por etapas, simulando uma situação mais próxima da realidade. Porém, o número de ciclos, e conseqüentemente o tempo, para atingir o novo equilíbrio de forças é maior em comparação com o modo de escavação instantânea (Figura 48).

Figura 48 - Gráficos do somatório das forças internas do modelo: a) somatório de forças após a realização da escavação instantânea das galerias e dos realces, até atingir um novo equilíbrio; b) somatório de forças após a realização da escavação das galerias e escavação gradual dos realces, até atingir um novo equilíbrio.

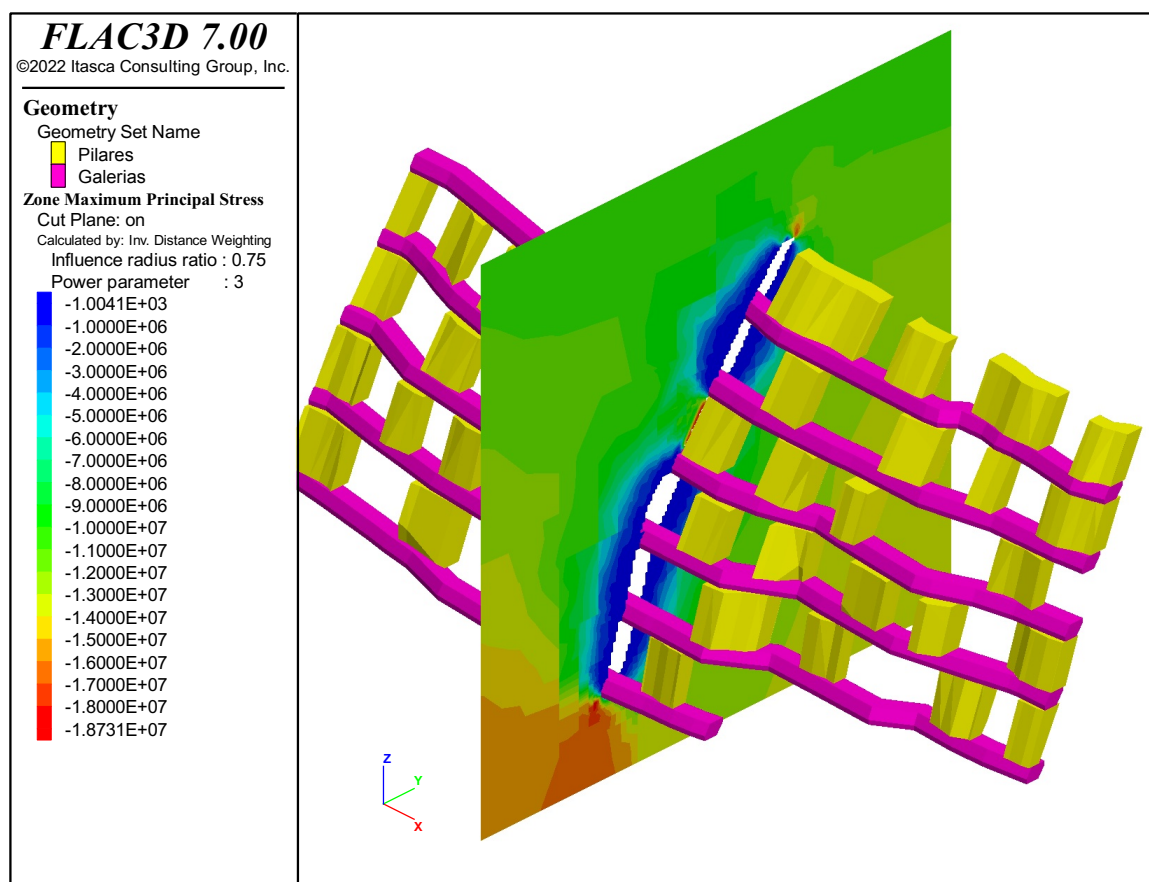


Fonte: Autoria própria.

### 3.3.4 Critério de falha do *hangingwall*

Com a obtenção do novo equilíbrio, após a escavação, é possível verificar a redistribuição de tensões no *hangingwall* dos realces, observando o volume da zona de relaxamento em cada realce, pelos isovalores da tensão  $\sigma_3$ . Quanto maior a zona de relaxamento, maior será a instabilidade esperada no *hangingwall*, devido a perda de confinamento, aumentando a possibilidade de diluição do minério. A Figura 49 a seguir mostra uma seção em uma vista isométrica, dos valores da tensão  $\sigma_3$  após a escavação dos realces.

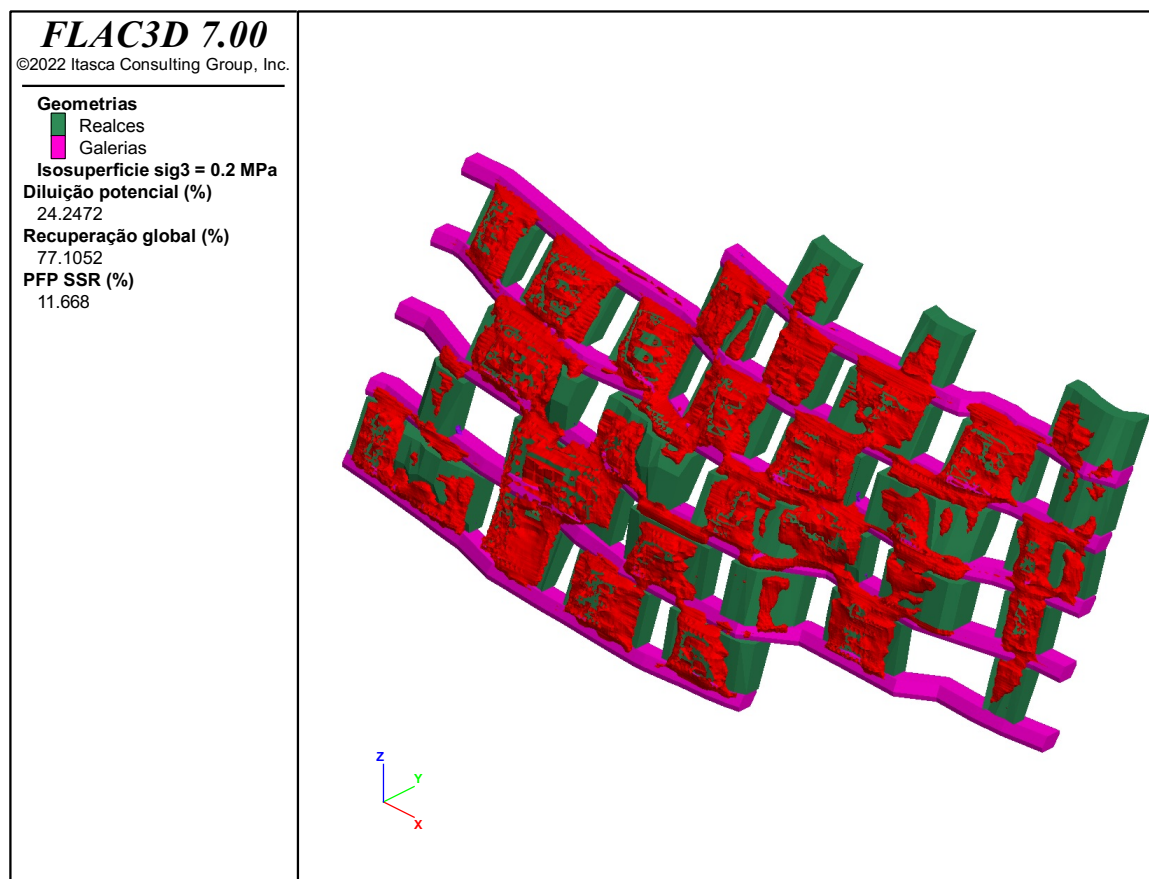


Figura 49 - Figura isométrica com uma seção 3D dos isovalores da tensão  $\sigma_3$ , após a escavação.

Fonte: Autoria própria.

A Figura 50 a seguir mostra uma vista isométrica, com a isosuperfícies de valores da tensão  $\sigma_3$  de 0,2 MPa após a escavação dos realces.

Figura 50 - Exemplo de plotagem gerada automaticamente mostrando as isosuperfícies de elementos sob relaxamento de tensões no *hangingwall*.



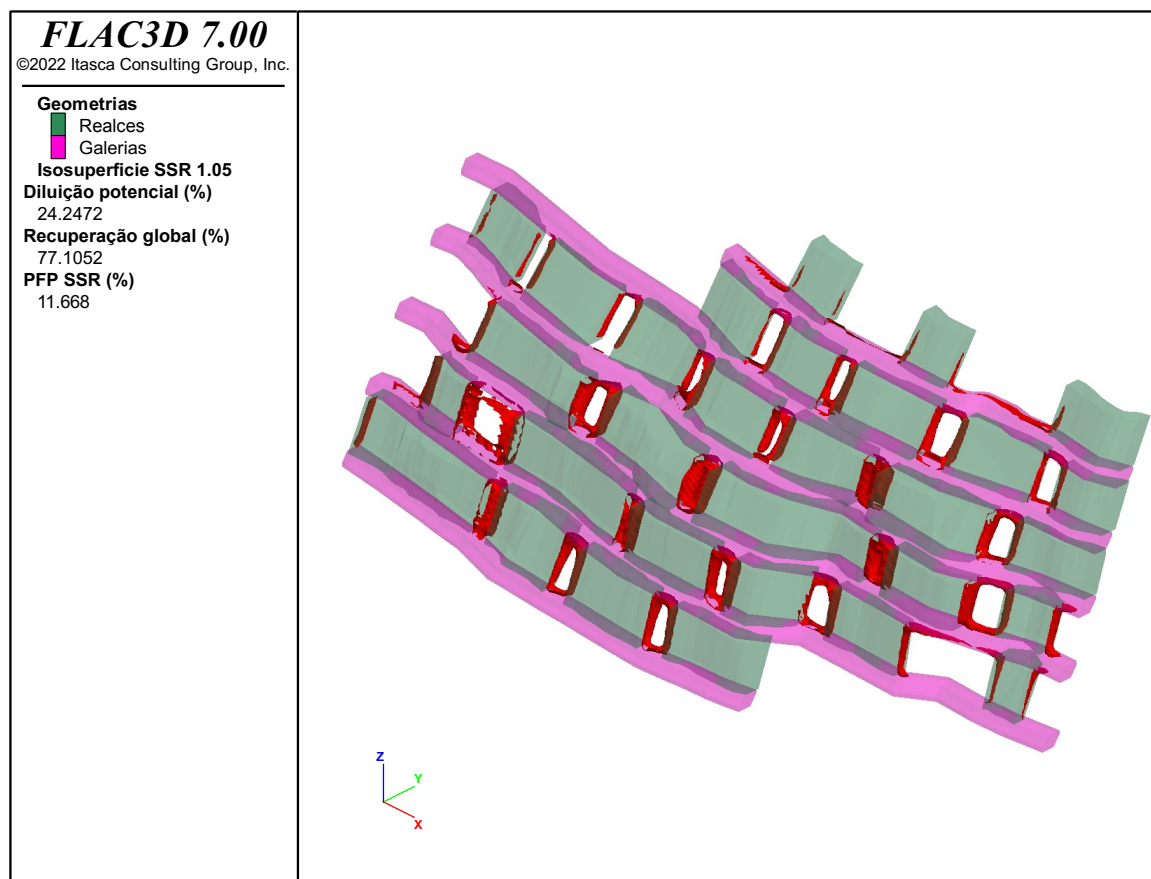
Fonte: Autoria própria.

### 3.3.5 Critério de falha dos pilares

O desempenho dos pilares será analisado pelo SSR obtido. De maneira geral o núcleo dos pilares apresentará um valor de SSR elevado, enquanto nas bordas do pilar, o SSR normalmente é próximo a um. A Figura 51 a seguir mostra as zonas na borda dos pilares com SSR menores do que 1.05.

O volume dos elementos dos pilares com falha, juntamente com o volume da zona de relaxamento de tensões no *hangingwall* dos realces, serão os parâmetros que irão avaliar o desempenho geomecânico dos cenários analisados. O código/ algoritmo, para realização desse processo de avaliação, está sendo escrito em linguagem FISH, linguagem de programação interna do FLAC3D, e o resultado para cada cenário será armazenado em um banco de dados para ser utilizado no algoritmo genético em *Python*.

Figura 51 - Exemplo de gráfico gerado automaticamente mostrando as isosuperfícies de elementos sob falha nos pilares.



Fonte: Autoria própria.

# Capítulo 4

## 4 Aplicação prática da metodologia proposta

Neste capítulo o algoritmo proposto será executado utilizando-se dados de um estudo de caso selecionado. Comparações e variações nos parâmetros de entrada serão consideradas com objetivo de verificar a flexibilidade do algoritmo em obter a solução ótima em diferentes situações.

### 4.1 Estudo de caso

O estudo de caso considerado trata-se de um projeto de extensão de um painel de lavra de uma mina subterrânea de ouro, com teor de minério marginal e vida útil de aproximadamente 2 anos. Devido a sua curta vida útil e por critério econômico da empresa, o projeto definia a adoção do método de subníveis, com realces deixados abertos, lavrados no sentido descendente.

#### 4.1.1 Localização e geologia

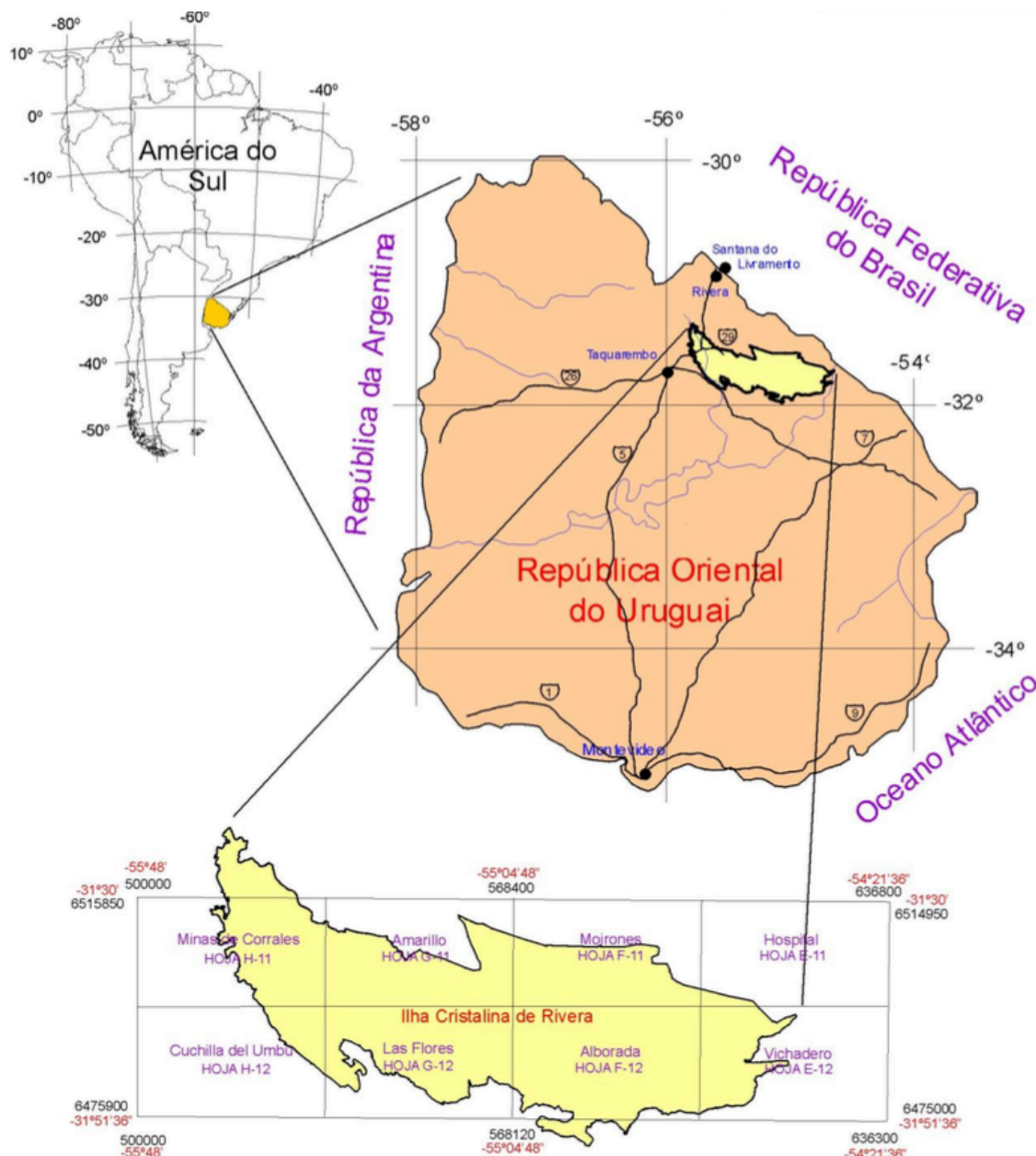
O depósito San Gregorio está situado na cidade de Minas de Corrales, no norte do Uruguai, departamento de Rivera, cerca de 70 km ao sul da fronteira internacional com o Brasil e 450 km ao norte da capital do Uruguai, Montevideu (Figura 52). O complexo constitui uma série de depósitos reconhecidos, alguns destes já lavrados por minas a céu aberto e por minas subterrâneas.

As rochas do embasamento, de idade Proterozoica, foram expostas a uma janela de erosão em uma sobreposição de sedimentos do paleozoico, formando um cinturão de 110 km de comprimento por 20-40 km de largura, denominado de Ilha Cristalina de Rivera. O cinturão é genericamente definido como um cinturão de granito e rochas verdes, entretanto, os graus de metamorfismo envolvidos variam de fácies xistos verdes a granulito. A mineralização anômala de ouro está presente ao longo dos 110 km de comprimento da Ilha Cristalina e está espacialmente associada com a falha de cisalhamento de Rivera, de escala regional. O depósito estudado está localizado na porção oeste da referida ilha.

A mineralização de ouro está associada a uma zona de falha rúptil-dúctil de tendência leste, mergulhando para o sul, denominada sistema de falhas de San Gregorio. O sistema de falhas de San Gregorio mergulha para o sul em um ângulo moderado a

acentuado, tem mais de 100 m de largura e é tipicamente bem limitado pelas estruturas do *hangingwall* e de *footwall*. As rochas hospedeiras são rochas graníticas, monzoníticas e dioríticas de idade proterozóica que foram metamorfoseadas em fácies anfíbolito.

Figura 52 - Mapa de localização da Ilha Cristalina de Rivera



Fonte: QUADROS, 2000

A mineralização do ouro está localizada ao longo de um filão central de quartzo-carbonato-pirita com quantidades variáveis de veios *stockwork* de sílica-pirita imediatamente adjacentes ao *hangingwall* e *footwall*. A zona de cisalhamento alterada tem até 100 m de largura e é geralmente composta por alteração de sílica-sericita-carbonato-

pirita e rochas fortemente deformadas, incluindo brechas abundantes na parede do *footwall*.

#### **4.1.2 Equipamentos e parâmetros operacionais**

Para o desenvolvimento horizontal de galerias o projeto considerava o uso de dois jumbos de dois braços, modelo SandVik DD-320. Para o carregamento da rocha desmontada, duas LHDs com capacidade de 5,4m<sup>3</sup> utilizadas no desenvolvimento e produção. Para o transporte do material, quatro caminhões rodoviários de 30t seriam utilizados.

Os equipamentos de perfuração radial disponíveis para o projeto são duas perfuratrizes *top-hammer* marca SandVik modelo DL-310, capazes de realizar perfurações com diâmetro entre 64-102mm e de no máximo 38m de comprimento. Em situações normais os desvios dos furos são muito maiores que o aceitável para comprimentos a partir dos 20m. Portanto, este foi um dos principais parâmetros limitantes para a separação máxima entre níveis, além de outros parâmetros como a variação de mergulho do corpo de minério.

Sendo assim, operacionalmente a separação entre níveis foi definida em 15m (de piso a teto). Por sua vez o comprimento mínimo admissível para os realces foi definido em 7m, por motivos econômicos e operacionais, pois para a escavação de cada realce é necessária a abertura da primeira face livre (*slot*) rompendo-se com o nível superior. Essa etapa de abertura de face livre agrega um grande custo econômico e de tempo, sendo necessário “diluir” estes custos ao longo da lavra de todo o realce.

#### **4.1.3 Propriedades e domínios geotécnicos**

As propriedades consideradas para o estudo de caso são descritas na Tabela 3 a seguir, e foram obtidas a partir da revisão geomecânica fornecida pela empresa. As propriedades do maciço rochoso foram definidas com base na combinação de descrições geomecânicas de 59 furos orientados totalizando 7.176 m de testemunhos de rocha, ensaios de resistência em laboratório (66 ensaios UCS, 26 ensaios brasileiros de tração, 36 ensaios de compressão triaxial e 24 ensaios de carga pontual).

Tabela 3 - Propriedades geomecânicas consideradas para o estudo de caso.

Domínio	Litologia	Peso específico (t/m <sup>3</sup> )	GSI	Young (GPa)	Poisson	B (GPa)	G (GPa)	Phi (°)	c (MPa)
<i>Hangingwall</i>	MNZ	2,75	55	21,94	0,23	13,78	8,88	31,21	2,00
	OZM	2,70	60	42,99	0,22	25,99	17,56	28,31	3,75
Minério	OZ	2,77	70	46,41	0,20	26,16	19,27	30,95	9,03
	GRN	2,63	65	44,07	0,21	25,70	18,15	33,64	7,54

Fonte: Autoria própria.

Para o domínio do *hangingwall* se decidiu pela utilização das mesmas propriedades ao longo de toda a zona, pois, a zona de falha do *hangingwall* não possui um contato tão definido. Caso contrário, poderia se utilizar de propriedades distintas para a zona de falha, na forma de um subdomínio, tal como o algoritmo já prevê a possibilidade do usuário incluí-la.

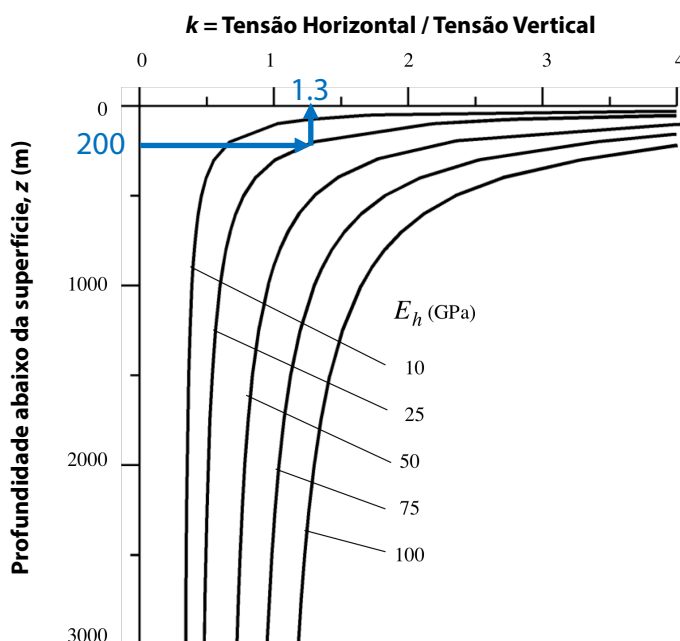
#### 4.1.4 Tensões *in situ*

Até a presente data não existem estudos publicados ou realizados nesta região do Uruguai sobre medidas de tensões *in situ*. Portanto, a magnitude das tensões foi assumida levando em conta que o ambiente geológico é relativamente estável e distante de influências tectônicas importantes.

Para a tensão vertical se considerou como base a equação linear obtida por Brown e Hoek (1978). De maneira mais conservadora, foi considerado o valor médio do peso específico de 0,028 MN/m<sup>3</sup> para o cálculo. Por sua vez a componente de tensão horizontal ( $\sigma_h$ ) pode ser calculada utilizando-se um fator ( $k$ ) multiplicado pela tensão vertical ( $\sigma_v$ ). Dessa forma, foi considerado que a tensão principal maior ( $\sigma_1$ ) e a tensão principal intermediária ( $\sigma_2$ ) são de mesma magnitude e horizontais. A tensão principal menor ( $\sigma_3$ ) é vertical.

As curvas da Figura 53 relacionando  $k$  com a profundidade abaixo da superfície  $z$ , segundo as equações de Sheorey (1994), são parecidas com as publicadas por Brown e Hoek (1978) para estimativas de tensões *in situ*. Dessa forma, considerando um módulo de elasticidade médio de 25 GPa e uma profundidade média de 200 metros foi considerado o valor de  $k$  igual a 1.3.

Figura 53 – Valores de  $k$  para diferentes módulos de deformação, variando com a profundidade.



Fonte: Modificado de Sheorey (1994).

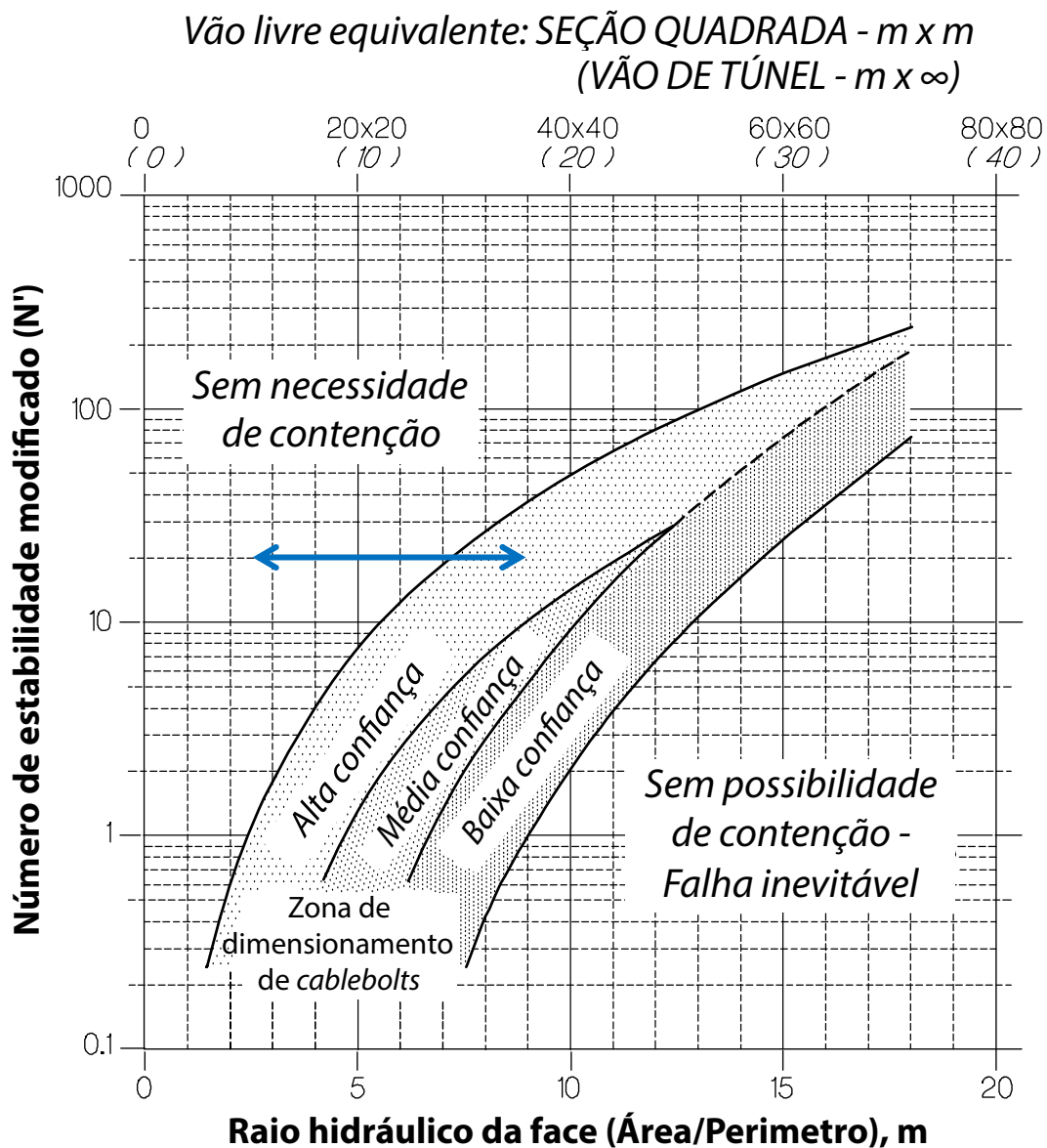
#### 4.1.5 Estabilidade dos realces e do *hangingwall*

Análises empíricas foram realizadas para avaliar a estabilidade dos realces. Também foi considerado o fator ELOS para se estimar a diluição no *hangingwall*. O número de estabilidade ( $N'$ ) foi considerado com valor médio 20. A largura considerada foi entre 7 e 25m, e a altura entre 15m e 60m (considerando até 4 níveis de realces conectados). O raio hidráulico mínimo obtido foi de 2,5m e o máximo de 8,8m.

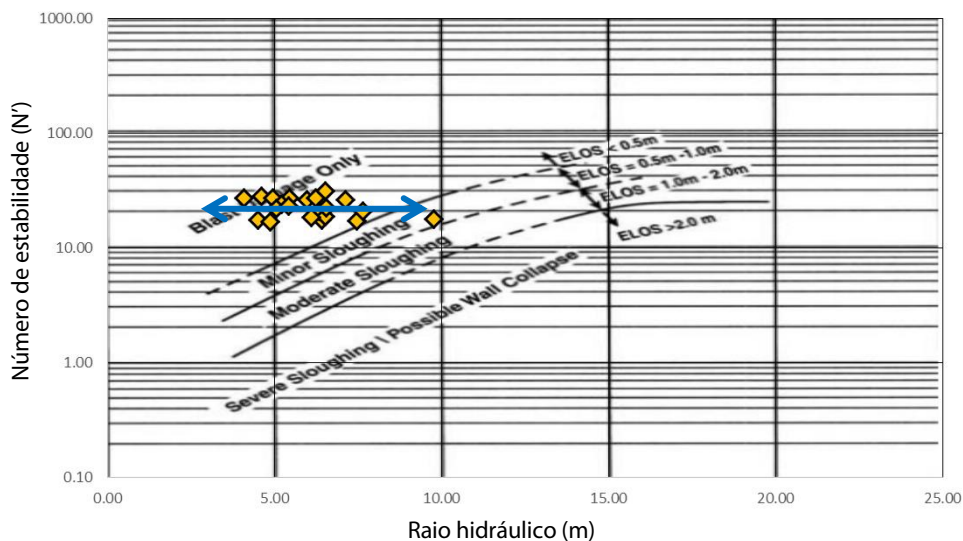
Sendo assim, os realces foram considerados estáveis, sem necessidade de suporte, para raios hidráulicos até 7m. Por sua vez, RH entre 7m e 13m são aceitos, considerando suporte com *cablebolts* (com média confiança), conforme demonstra a Figura 54.



Figura 54 - Gráfico de estabilidade, para as faces dos realces, considerando as variações estabelecidas representadas pela seta cor azul.



Para raios hidráulicos da ordem de 10m, um fator ELOS entre 0,5 e 1m poderia ser esperado para a diluição do *hangingwall* (Figura 56).

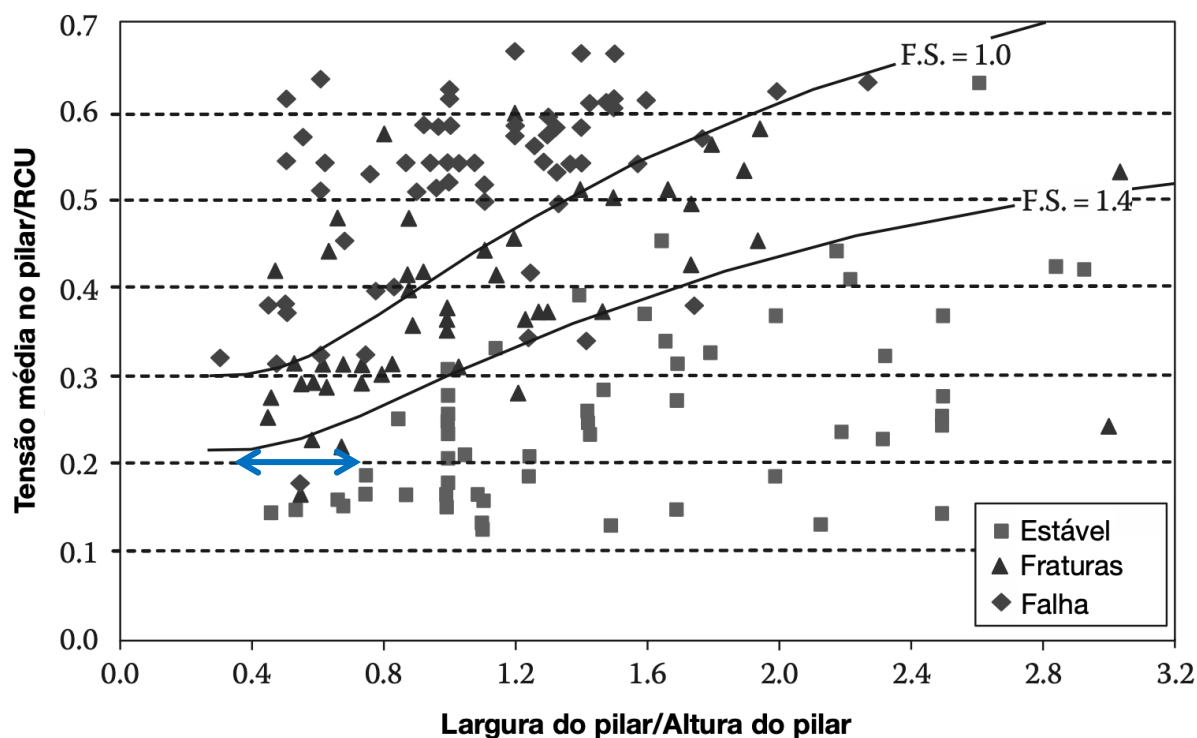
Figura 55 – Avaliação do fator ELOS para o *hangingwall*.

Fonte: Modificado de Villaescusa (2014).

#### 4.1.6 Estabilidade dos pilares

A análise de estabilidade dos pilares foi realizada segundo a metodologia de Lunder e Pakalnis (1997). A altura dos pilares foi considerada fixa de 15m, e a largura variável entre 5 e 10m. A tensão média nos pilares foi considerada entre 15 e 20 MPa e a resistência a compressão uniaxial considerada foi de 90 MPa. Na Figura 56 a seguir pode se verificar a zona de estabilidade considerada para os pilares, dadas as variações consideradas. Os fatores de segurança são próximos a 1,4.

Figura 56 – Gráfico empírico de estabilidade de pilares, a seta em azul representa a zona para os pilares possíveis, considerando as variações dos parâmetros.



Fonte: Modificado de Lunder (1994).

#### 4.1.7 Parâmetros econômicos e operacionais

Os parâmetros econômicos considerados para o estudo de caso foram atualizados para o ano 2021. Por sua vez, os parâmetros operacionais os quais são julgados interessantes para a otimização também podem ser observados na Tabela 4 a seguir.

Tabela 4 – Parâmetros econômicos e operacionais considerados para o estudo de caso.

Parâmetro	Valor	Unidade
Preço de venda do ouro	55,0	US\$/g
Recuperação de lavra nos realces	90,0	%
Recuperação do processo de beneficiamento	93,0	%
Diluição máxima aceitável com o <i>hangingwall</i>	10,0	%
Porcentagem máxima aceitável de falha em pilares (PAF)	10,0	%
Custo de lavra	16,5	US\$/t
Custo de transporte	0,00014	US\$/t/m
Custo de processo	18,6	US\$/t
Custos gerais e administrativos	18,1	US\$/t
Custo de refino	0,5	US\$/g
Custo de <i>royalty</i>	0,015	% do <i>gross value</i>
Custo de suporte secundário no <i>hangingwall</i> (quando acima da máxima diluição aceitável)	1,0	US\$/m <sup>3</sup> de diluição no cenário)
Eficiência do suporte secundário no <i>hangingwall</i>	70	%
Custo de reforço de pilares (quando a PAF for acima da PAF)	0,9	US\$/m <sup>3</sup> de pilar com falha)

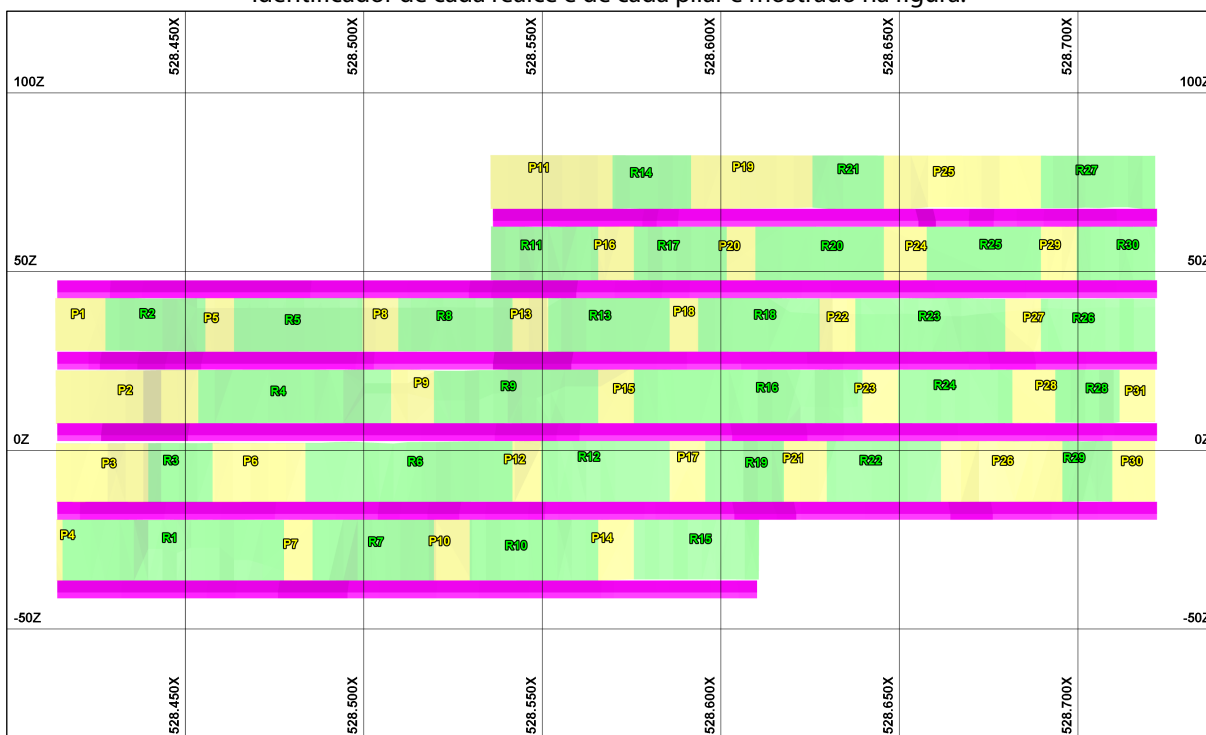
Fonte: Autoria própria.

#### 4.1.8 Cenário ótimo adotado pela metodologia tradicional

Considerando todas as particularidades do estudo de caso, o cenário ótimo adotado pela empresa para o painel de lavra, através da metodologia tradicional de planejamento e projeto de realces e pilares, previa uma produção de minério de 304.860,00 t, com um teor médio de 1,48 g/t, o que totaliza 14.505 onças contidas, com uma recuperação global do depósito de 77,1%. A obtenção desta otimização pelo método tradicional, envolveu o esforço do departamento de geologia, engenharia e de geomecânica. Ao todo pode se considerar um total aproximado de 40h de trabalho, entre os desenhos preliminares dos realces, a partir do modelo geológico e modelo de blocos, até findar com as otimizações geomecânicas por métodos de modelagem numérica.

A Figura 57 demonstra o arranjo ótimo de realces e pilares, obtido pela empresa, considerando a metodologia tradicional. De maneira geral se observa que o arranjo é relativamente regular, com realces de maior dimensão na porção central do painel de lavra. O número total de realces considerados é de 30 e o número de pilares 31.

Figura 57 – Vista para o norte, do arranjo ótimo dos realces e pilares, considerando a metodologia tradicional utilizada. Os realces são mostrados em verde, os pilares em amarelo e os subníveis em cor magenta. O código identificador de cada realce e de cada pilar é mostrado na figura.

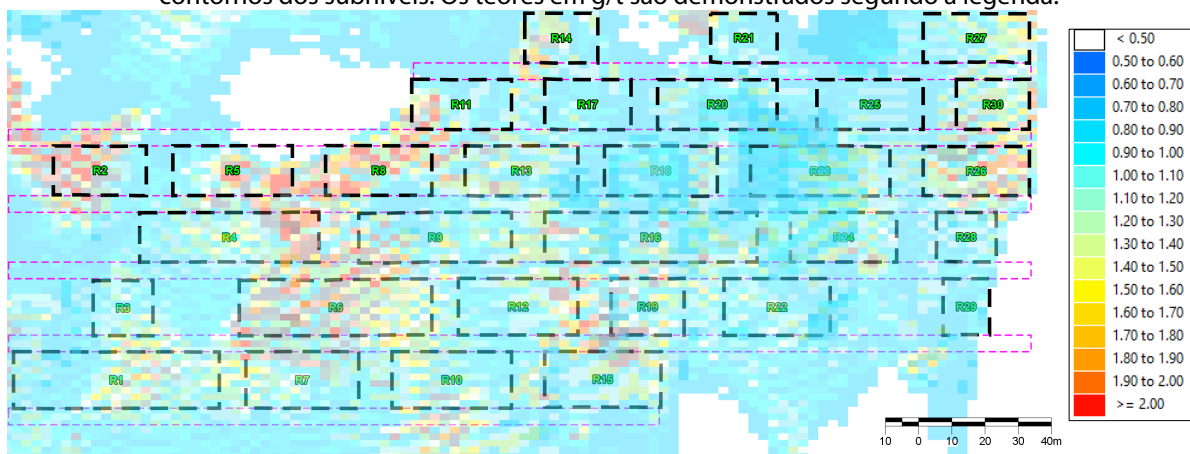


Fonte: Autoria própria.

A Figura 58 mostra uma vista olhando para o norte, contendo o contorno dos realces, representados com a distribuição dos teores dos blocos. As cores quentes representam os altos teores, acima de 2 g/t. A Tabela 5 traz individualmente todos os realces definidos com a sua tonelagem, teor e onças. De maneira geral se observa que os realces apresentam teores próximos a média, com desvio padrão de 0,27 g/t e todos os realces com teores maiores do que 1 g/t.

Da mesma forma a Tabela 6 lista individualmente todos os pilares definidos, com tonelagem, teor e onças contidas. Se observa que os pilares apresentam uma maior variação de teores sendo a média de 1,04 g/t e as toneladas totais de 129.406,0 t e onças totais de 4.320,6. Os pilares P15 e P22 elevam a média por estarem em zonas de alto teor.

Figura 58 - Vista para o norte, do contorno do arranjo ótimo dos realces em linha tracejada cor preta, considerando a metodologia tradicional utilizada. As linhas tracejadas em cor magenta representam os contornos dos subníveis. Os teores em g/t são demonstrados segundo a legenda.



Fonte: Autoria própria

Tabela 5 – Propriedades dos realces do cenário ótimo considerado pela empresa, utilizando a metodologia tradicional.

<b>Código do Realce</b>	<b>Volume (m³)</b>	<b>Densidade (t/m³)</b>	<b>Ton</b>	<b>Teor (g/t)</b>	<b>Onças contidas</b>
<b>R1</b>	9.015,9	2,77	24.974,0	1,52	1.220,4
<b>R2</b>	2.917,3	2,77	8.081,1	1,54	400,1
<b>R3</b>	2.182,6	2,77	6.045,7	1,21	235,2
<b>R4</b>	5.562,2	2,77	15.407,2	1,42	703,4
<b>R5</b>	3.511,7	2,77	9.727,5	1,03	322,1
<b>R6</b>	6.160,2	2,77	17.063,7	1,94	1.064,3
<b>R7</b>	3.947,4	2,77	10.934,2	1,60	562,5
<b>R8</b>	3.007,0	2,77	8.329,5	1,22	326,7
<b>R9</b>	6.428,1	2,77	17.805,7	1,63	933,1
<b>R10</b>	3.883,2	2,77	10.756,5	1,53	529,1
<b>R11</b>	2.301,1	2,77	6.374,0	1,23	252,1
<b>R12</b>	4.198,2	2,77	11.629,1	1,81	676,7
<b>R13</b>	3.186,5	2,77	8.826,7	1,67	473,9
<b>R14</b>	1.882,3	2,77	5.213,8	1,53	256,5
<b>R15</b>	3.492,4	2,77	9.673,9	1,35	419,9
<b>R16</b>	7.776,6	2,77	21.541,1	1,62	1.121,9
<b>R17</b>	2.278,4	2,77	6.311,1	1,40	284,1
<b>R18</b>	3.831,1	2,77	10.612,2	1,22	416,2
<b>R19</b>	2.701,4	2,77	7.482,9	2,13	512,4
<b>R20</b>	4.014,5	2,77	11.120,2	1,34	479,1
<b>R21</b>	1.654,0	2,77	4.581,7	1,06	156,1
<b>R22</b>	3.025,6	2,77	8.381,0	1,22	328,7
<b>R23</b>	6.105,8	2,77	16.913,2	1,68	913,5
<b>R24</b>	3.775,3	2,77	10.457,7	1,45	487,5
<b>R25</b>	3.005,6	2,77	8.325,4	1,20	321,2
<b>R26</b>	2.573,0	2,77	7.127,2	1,50	343,7
<b>R27</b>	2.586,7	2,77	7.165,3	1,03	237,3
<b>R28</b>	1.662,4	2,77	4.604,8	1,28	189,5
<b>R29</b>	1.521,8	2,77	4.215,4	1,15	155,9
<b>R30</b>	1.869,4	2,77	5.178,3	1,09	181,5
		<b>Total</b>	304.860,0	1,48	14.504,5
		<b>Min</b>	4.215,4	1,03	155,9
		<b>Max</b>	24.974,0	2,13	1.220,4
		<b>Desv. Pad.</b>	5142,7	0,27	297,4

Fonte: Autoria própria

Tabela 6 - Propriedades dos pilares do cenário ótimo considerado pela empresa, utilizando a metodologia tradicional.

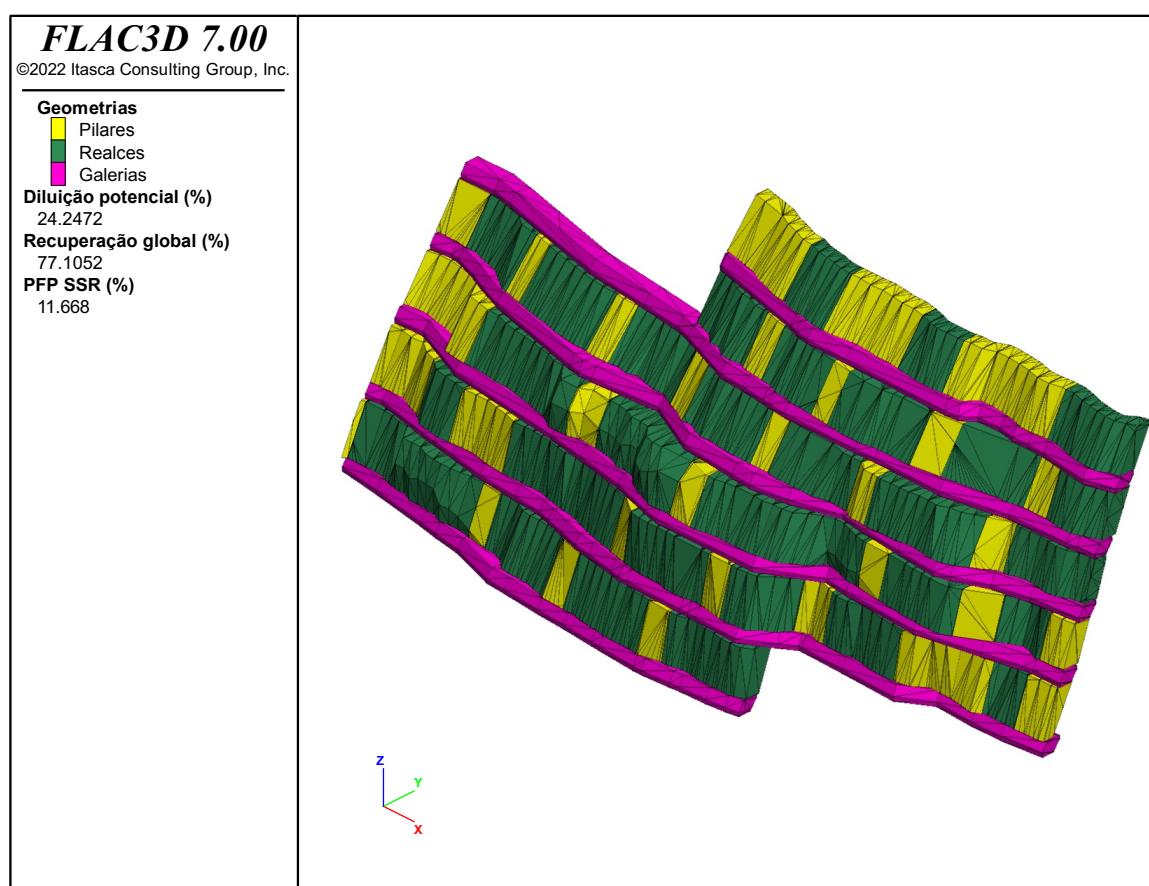
<b>Código do Pilar</b>	<b>Volume (m³)</b>	<b>Densidade (t/m³)</b>	<b>Ton</b>	<b>Teor (g/t)</b>	<b>Onças contidas</b>
<b>P1</b>	1.319,4	2,77	3.654,6	0,96	112,8
<b>P2</b>	3.853,7	2,77	10.674,8	0,76	260,8
<b>P3</b>	2.897,7	2,77	8.026,7	1,04	268,4
<b>P4</b>	226,5	2,77	627,3	1,07	21,6
<b>P5</b>	780,4	2,77	2.161,7	0,48	33,4
<b>P6</b>	2.361,5	2,77	6.541,3	0,77	161,9
<b>P7</b>	823,2	2,77	2.280,3	1,06	77,7
<b>P8</b>	975,5	2,77	2.702,1	1,90	165,1
<b>P9</b>	1.588,7	2,77	4.400,7	1,80	254,7
<b>P10</b>	1.203,3	2,77	3.333,3	1,30	139,3
<b>P11</b>	2.928,4	2,77	8.111,7	0,65	169,5
<b>P12</b>	939,7	2,77	2.603,0	1,43	119,7
<b>P13</b>	929,1	2,77	2.573,6	1,57	129,9
<b>P14</b>	1.060,6	2,77	2.937,9	1,27	120,0
<b>P15</b>	1.431,0	2,77	3.963,8	2,10	267,6
<b>P16</b>	841,0	2,77	2.329,4	1,87	140,0
<b>P17</b>	1.203,6	2,77	3.334,0	1,91	204,7
<b>P18</b>	798,3	2,77	2.211,2	0,88	62,6
<b>P19</b>	2.851,7	2,77	7.899,3	0,51	129,5
<b>P20</b>	818,1	2,77	2.266,3	0,87	63,4
<b>P21</b>	1.184,5	2,77	3.281,0	1,11	117,1
<b>P22</b>	1.428,1	2,77	3.955,9	2,06	262,0
<b>P23</b>	1.095,6	2,77	3.034,9	1,22	119,0
<b>P24</b>	1.306,8	2,77	3.619,7	0,93	108,2
<b>P25</b>	3.746,5	2,77	10.377,9	0,66	220,2
<b>P26</b>	3.036,1	2,77	8.409,9	0,92	248,7
<b>P27</b>	966,4	2,77	2.677,0	1,42	122,2
<b>P28</b>	1.308,1	2,77	3.623,3	0,96	111,8
<b>P29</b>	817,5	2,77	2.264,4	1,15	83,7
<b>P30</b>	1.074,5	2,77	2.976,3	0,03	2,9
<b>P31</b>	921,4	2,77	2.552,3	0,27	22,2
		<b>Total</b>	129.406,0	1,04	4.320,6
		<b>Min</b>	627,3	0,03	2,9
		<b>Max</b>	10.674,8	2,10	268,4
		<b>Desv. Pad.</b>	2600,2	0,52	77,8

Fonte: Autoria própria

Após reanalisar a geometria do cenário por modelagem numérica, utilizando o critério de *unbalanced forces* de  $1 \times 10^{-5}$ , foi obtido um potencial de diluição de 24,3%, o que considerando uma eficiência de suporte do *hangingwall* de 70%, levaria a uma diluição real de 14,3%.

A receita bruta foi calculada em US\$ 20.768.734,22 e os custos em US\$ 17.180.390,64 totalizando um lucro líquido de US\$ 3.588.343,58. Ao se desconsiderar o uso de suporte no *hangingwall*, ou seja, atribuindo-se a eficiência de suporte zero, a diluição real seria igual a diluição potencial máxima obtida pelo modelo numérico, em 24,3% levando a um lucro líquido de US\$ 2.120.361,81. A geometria do arranjo de pilares e realces deste cenário, analisada no software FLAC3D 7.0, pode ser observada na Figura 59 a seguir.

Figura 59 - Vista isométrica com o layout de realces e pilares tido como ótimo para a metodologia considerada pela empresa. Os resultados do modelo numérico, quanto a diluição potencial e a porcentagem de falha dos pilares (PFP) são mostrados.



Fonte: Autoria própria.

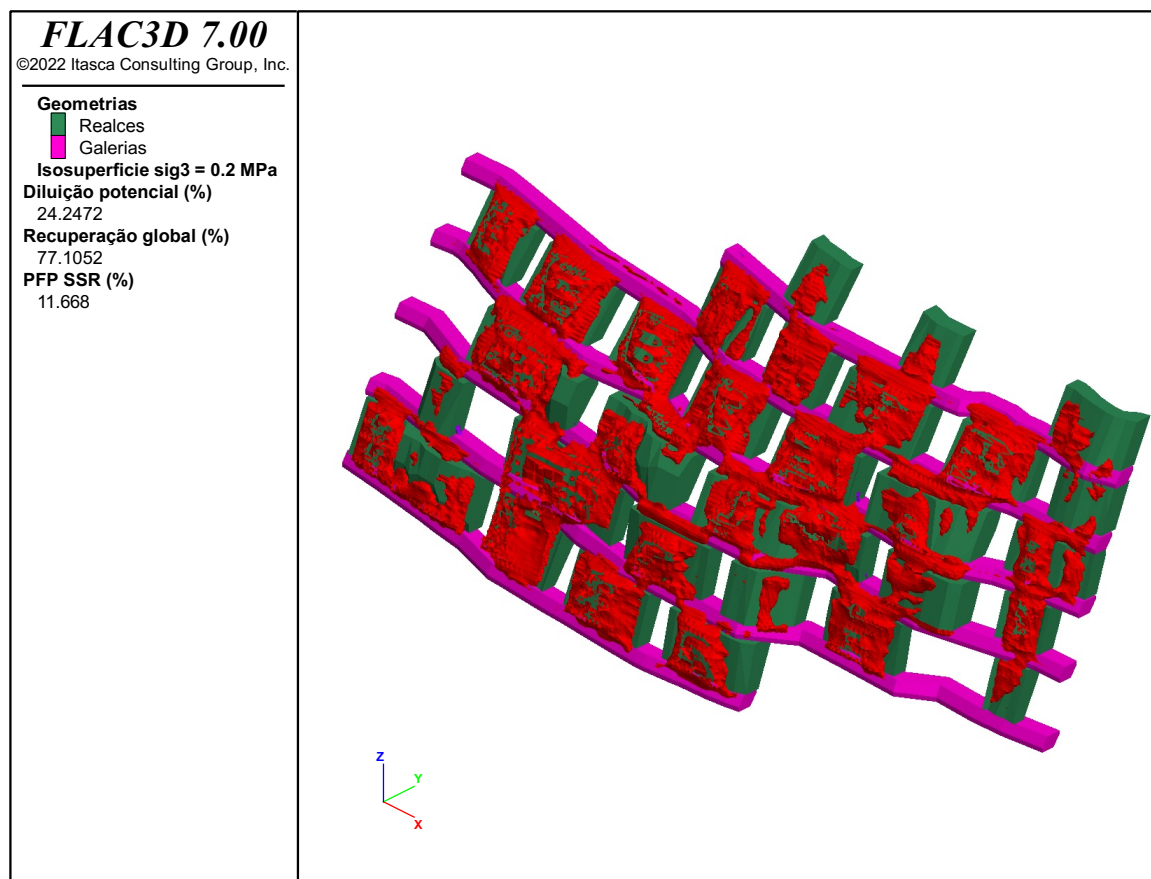
O desempenho geomecânico do cenário pode ser observado quanto a isosuperfície de tensão  $\sigma_3 < 0.2$  MPa, indicando o volume de diluição esperado pelo desprendimento do *hangingwall* (Figura 60). De maneira geral se observa que a diluição potencial é relativamente uniforme ao longo de todo *hangingwall*.

Da mesma forma o desempenho dos pilares também pode ser avaliado considerando os elementos do modelo numérico com valor de SSR menores do que 1.05, e



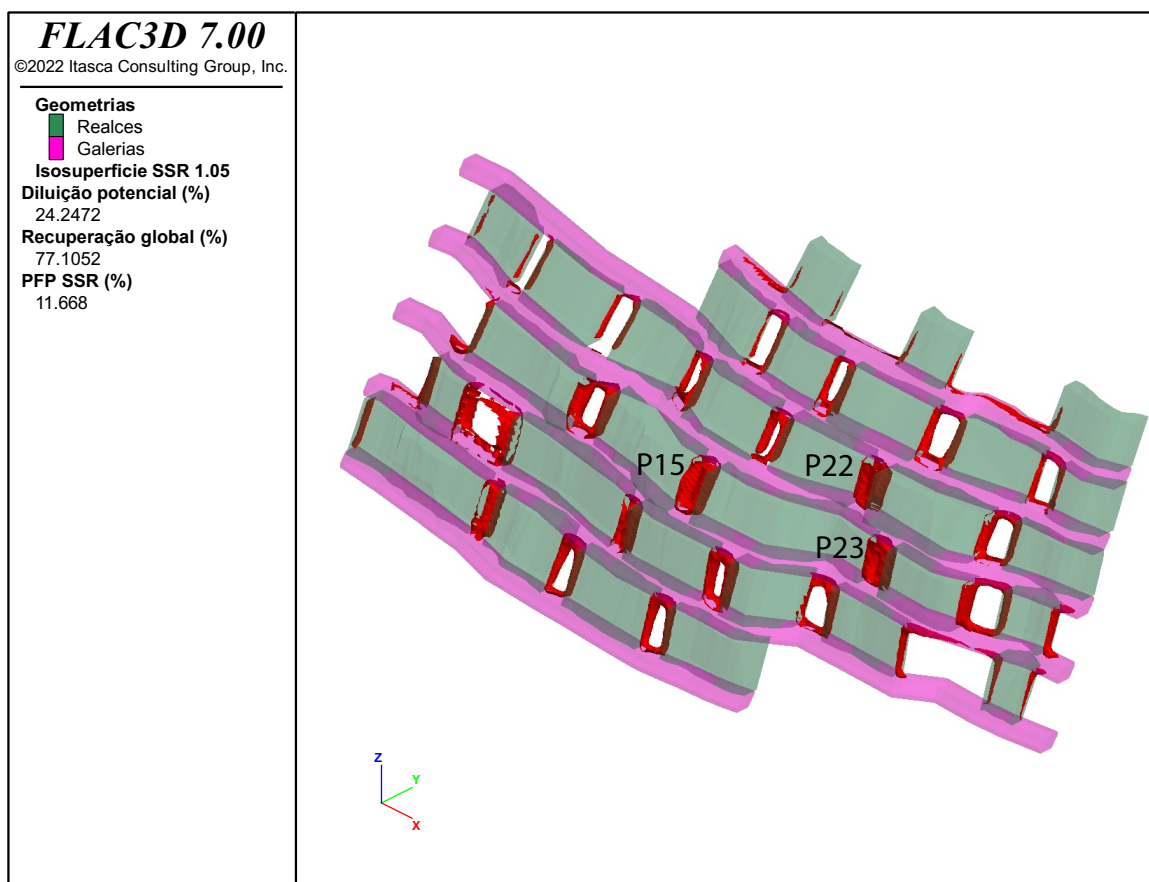
sua porcentagem de falha (PFP), conforme mostra a Figura 61. Os pilares P22, P23 e P15, da porção central, são observados com porcentagem de falha excessiva.

Figura 60 - Vista isométrica com zonas de falha no *hangingwall* em vermelho, para o cenário tido como ótimo para a metodologia considerada pela empresa.



Fonte: Autoria própria.

Figura 61 - Vista isométrica com zonas de falha nos pilares em vermelho, para o cenário tido como ótimo para a metodologia considerada pela empresa. Os pilares P22, P23 e P15, da porção central, são observados com porcentagem de falha excessiva.



Fonte: Autoria própria.

#### 4.2 Resultados e eficiências considerando a metodologia proposta

O resultado da otimização do estudo de caso, utilizando o algoritmo proposto, pode ser comparado com o resultado base do estudo de caso, obtido pela otimização manual da equipe de planejamento e geomecânica. A diferença entre os cenários permite quantificar o ganho econômico produzido e as diferenças do ponto de vista do desempenho geomecânico.

Também é importante avaliar o desempenho computacional do método, com relação ao tempo de execução da otimização. Para tanto, utilizou-se como parâmetro base a simulação estocástica de cenários possíveis, onde com o mesmo número de iterações é possível verificar a eficiência do algoritmo genético em otimizar o resultado.

#### 4.2.1 Parâmetros base considerados para o algoritmo genético

Para as primeiras otimizações realizadas os parâmetros do algoritmo genético foram testados. A populacional inicial de cenários foi testada com 20 e 30 indivíduos, sem obter uma mudança significativa no resultado ótimo. Por sua vez, a paciência testada inicialmente com valor 20 se mostrou muito pequena, o que ocasionava que o algoritmo finalizasse prematuramente. Ao adotar-se o valor de 30 o algoritmo tendeu a sempre atingir o número máximo de iterações, apresentando melhoras a cada iteração. Sendo assim, foram adotados os parâmetros para o cenário base, descritos na Tabela 7 a seguir.

Tabela 7 – Parâmetros do algoritmo genético considerados para o estudo de caso.

Parâmetro	Valor
População de cenários	20
Número de iterações máximas	200
Paciência	30
Probabilidade de mutação	1
Fator de melhora	0,1

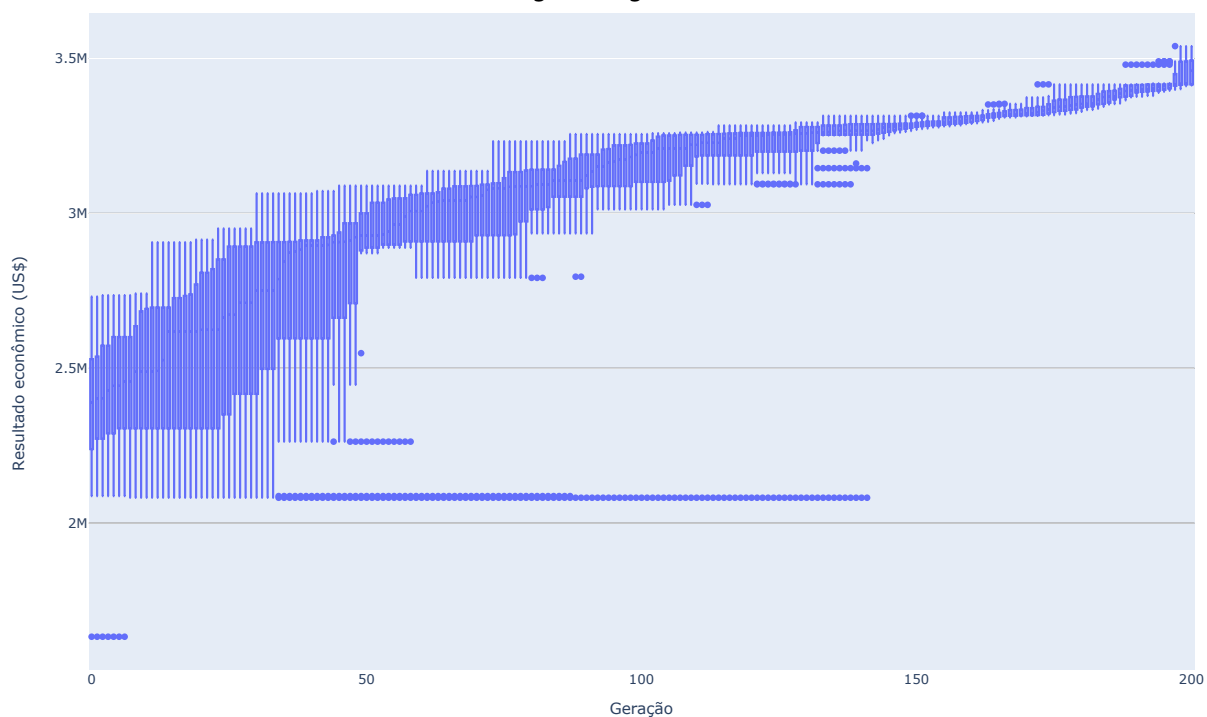
Fonte: Autoria própria.

#### 4.2.2 Resultados da otimização pelo algoritmo genético – cenário base

O resultado da otimização utilizando o algoritmo genético, com os parâmetros descritos nos tópicos anteriores, foi considerado como cenário base para futuras comparações. Com o objetivo de se obter um ganho no tempo computacional, o critério de *unbalanced forces* foi considerado em  $1 \times 10^{-3}$  para os modelos realizados pelo algoritmo. O cenário ótimo foi obtido após 200 gerações realizadas, sendo cada geração composta por dois cenários descendentes dos seus pais e igualmente analisados. Ao final da otimização, com o cenário ótimo obtido, um novo modelo numérico foi realizado desta vez com *unbalanced forces* definido em  $1 \times 10^{-5}$ .

A Figura 62 apresenta um diagrama de caixa com a evolução da população ao longo das novas gerações do algoritmo genético. Se observa que a população dos cenários realizados apresenta um acentuado crescimento inicial, dado pelo aumento da mediana do resultado econômico, tendendo a um patamar de menor taxa de crescimento após 100 gerações. Também é possível observar que ocorre uma diminuição considerável da dispersão dos resultados econômicos dos cenários após 150 gerações, ou seja, os indivíduos tendem a uma maior similitude quanto ao seu desempenho econômico. Porém, alguns cenários de menores valores extremos (*outliers*) persistiram dentro da população até próximo das 150 gerações.

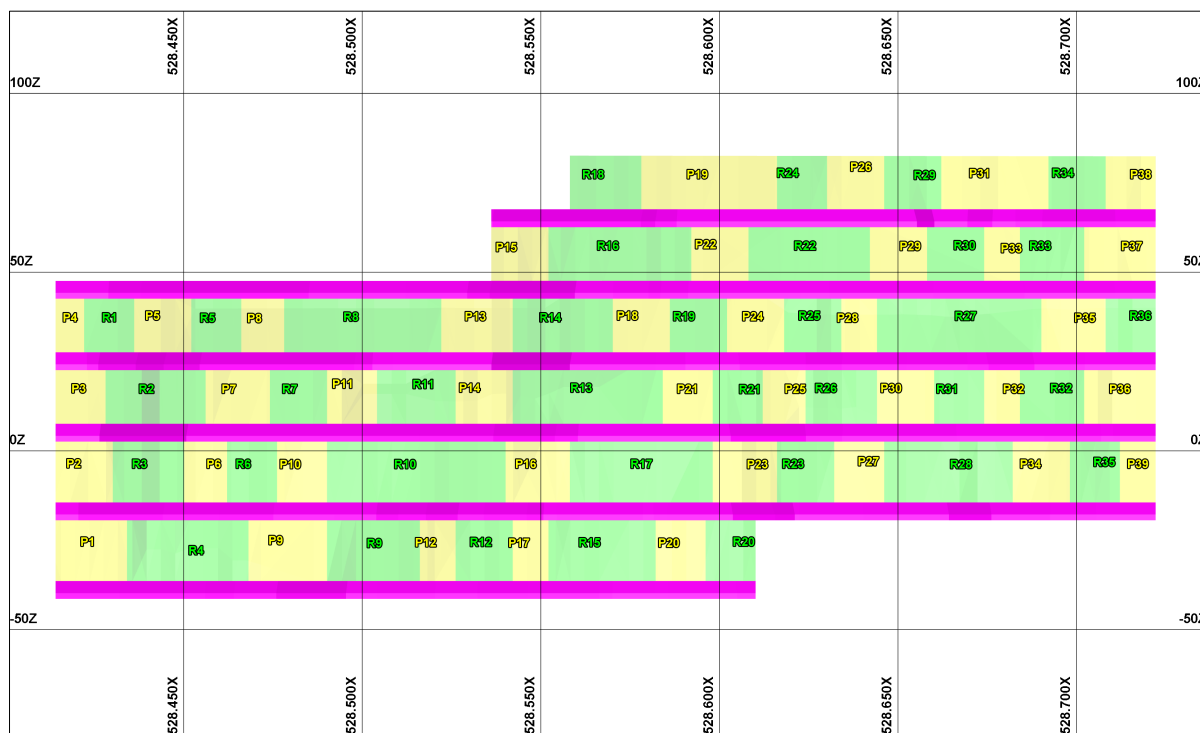
Figura 62 – Gráfico do tipo diagrama de caixas, mostrando a evolução da população ao longo das gerações do algoritmo genético.



Fonte: Autoria própria.

A Figura 63 demonstra o arranjo ótimo de realces e pilares obtido, e mostra a definição do número considerado para os pilares e realces. De maneira geral se observa que o arranjo obtido é irregular, com tamanhos variados de realces e pilares. Foram definidos um total de 36 realces e 39 pilares para o cenário ótimo, sendo o tamanho médio para os pilares de 7,9m e o tamanho médio dos realces de 11,8m.

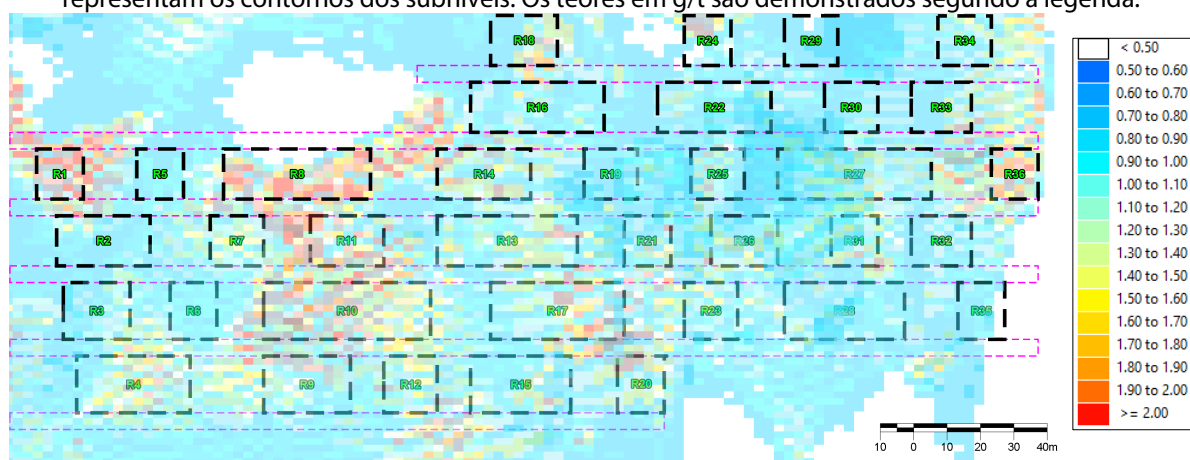
Figura 63 - Vista para o norte, do arranjo ótimo dos realces e pilares, considerando a nova metodologia proposta. Os realces são mostrados em verde, os pilares em amarelo e os subníveis em cor magenta. O código identificador de cada realce e de cada pilar é mostrado na figura.



Fonte: Autoria própria.

A Figura 64 apresenta uma vista para o norte, contendo o contorno dos realces, representados com a distribuição dos teores dos blocos. As cores quentes representam os altos teores, acima de 2 g/t.

Figura 64 - Vista lateral, olhando para o norte, do contorno do arranjo ótimo dos realces em linha tracejada cor preta, considerando a nova metodologia de otimização. As linhas tracejadas em cor magenta representam os contornos dos subníveis. Os teores em g/t são demonstrados segundo a legenda.



Fonte: Autoria própria.

A Tabela 8 traz individualmente todos os realces definidos com a sua tonelagem, teor e onças. De maneira geral se observa que os realces apresentam teores próximos a média, com desvio padrão de 0,40 g/t, com poucas exceções de realces com teor abaixo de 1 g/t.

Da mesma forma a Tabela 9 lista individualmente todos os pilares definidos, com tonelagem, teor e onças contidas. Se observa que os pilares apresentam uma maior variação de teores sendo a média de 1,18 g/t e as toneladas totais de 174.142,0 t e onças contidas de 6.612,2. O pilar P23 eleva a média por estar em zonas de alto teor.

Tabela 8 - Propriedades dos realces do cenário ótimo obtido, considerando a nova metodologia proposta.

<b>Código Realces</b>	<b>Volume</b>	<b>Densidade</b>	<b>Ton</b>	<b>Teor</b>	<b>Onças contidas</b>
<b>R1</b>	1.465,1	2,77	4.058,4	1,61	210,1
<b>R2</b>	2.908,5	2,77	8.056,5	0,90	233,1
<b>R3</b>	2.563,1	2,77	7.099,8	1,12	255,6
<b>R4</b>	5.539,6	2,77	15.344,7	1,65	814,0
<b>R5</b>	1.370,7	2,77	3.796,9	0,50	61,0
<b>R6</b>	1.210,6	2,77	3.353,4	0,73	78,7
<b>R7</b>	1.427,9	2,77	3.955,3	1,19	151,3
<b>R8</b>	4.257,6	2,77	11.793,5	1,55	587,7
<b>R9</b>	3.089,7	2,77	8.558,3	1,67	459,5
<b>R10</b>	5.366,7	2,77	14.865,7	2,01	960,6
<b>R11</b>	2.819,5	2,77	7.810,0	1,77	444,4
<b>R12</b>	1.748,5	2,77	4.843,3	1,51	235,1
<b>R13</b>	6.341,7	2,77	17.566,6	1,82	1027,9
<b>R14</b>	2.589,2	2,77	7.171,9	1,96	451,9
<b>R15</b>	3.174,0	2,77	8.792,1	1,33	375,9
<b>R16</b>	3.321,3	2,77	9.200,1	1,58	467,3
<b>R17</b>	4.603,5	2,77	12.751,8	1,93	791,2
<b>R18</b>	1.713,9	2,77	4.747,5	1,21	184,7
<b>R19</b>	1.610,3	2,77	4.460,6	0,93	133,4
<b>R20</b>	1.524,6	2,77	4.223,2	1,56	211,8
<b>R21</b>	1.872,2	2,77	5.185,9	1,80	300,1
<b>R22</b>	3.789,0	2,77	10.495,6	1,34	452,2
<b>R23</b>	1.585,3	2,77	4.391,3	1,13	159,5
<b>R24</b>	1.193,2	2,77	3.305,3	0,90	95,6
<b>R25</b>	2.140,6	2,77	5.929,3	1,76	335,5
<b>R26</b>	2.148,1	2,77	5.950,2	1,57	300,3
<b>R27</b>	6.251,5	2,77	17.316,5	1,60	890,8
<b>R28</b>	3.215,3	2,77	8.906,3	1,10	315,0
<b>R29</b>	1.368,7	2,77	3.791,4	0,51	62,2
<b>R30</b>	1.594,3	2,77	4.416,3	1,19	169,0
<b>R31</b>	1.610,2	2,77	4.460,2	1,67	239,5
<b>R32</b>	1.791,6	2,77	4.962,6	1,05	167,5
<b>R33</b>	1.469,2	2,77	4.069,8	1,18	154,4
<b>R34</b>	1.287,9	2,77	3.567,5	1,06	121,6
<b>R35</b>	1.519,3	2,77	4.208,4	1,01	136,7
<b>R36</b>	1.151,6	2,77	3.189,8	1,31	134,3
		<b>Total</b>	256.596,0	1,48	12.209,3
		<b>Min</b>	3.189,8	0,50	61,0
		<b>Max</b>	17.566,6	2,01	1.027,9
		<b>Desv. Pad.</b>	4122,5	0,40	264,0

Fonte: Autoria própria.

Tabela 9 - Propriedades dos realces do cenário ótimo obtido, considerando a nova metodologia proposta.

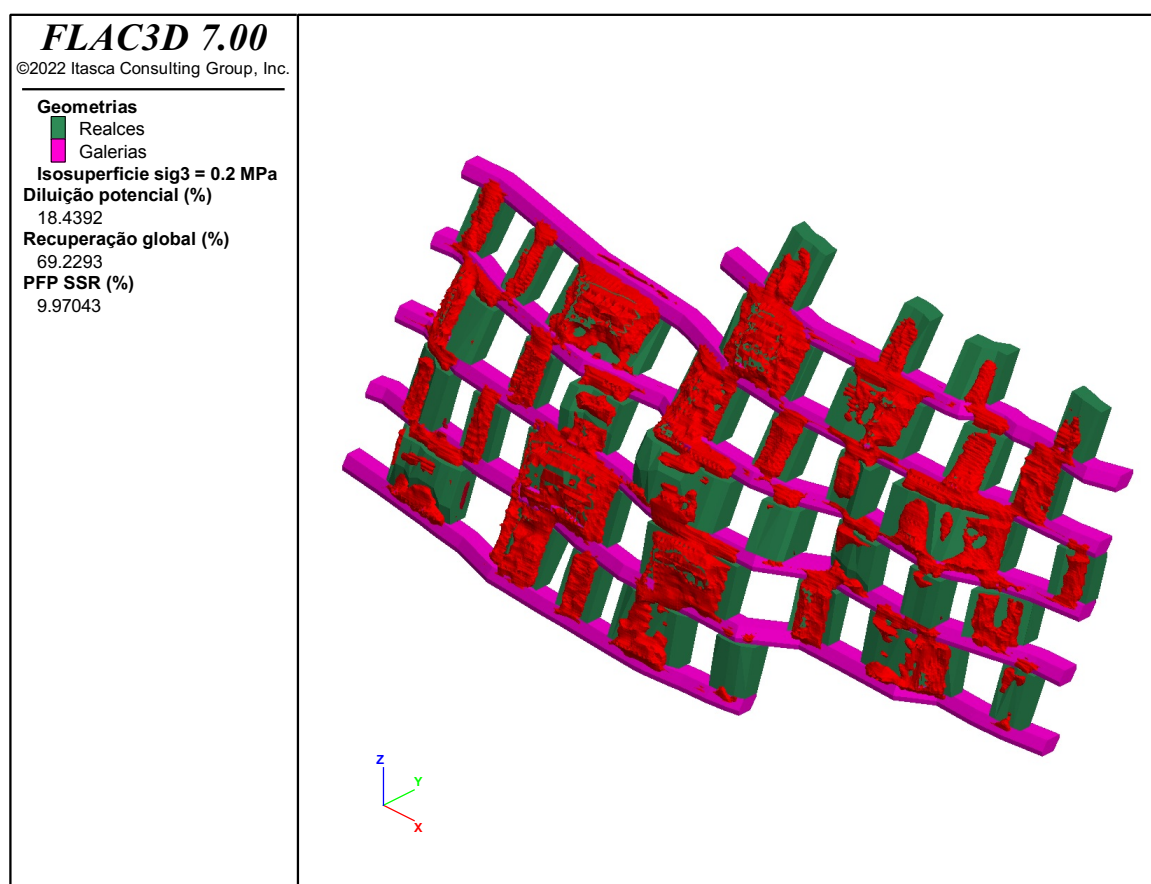
<b>Código Pilares</b>	<b>Volume (m³)</b>	<b>Densidade (t/m³)</b>	<b>Ton</b>	<b>Teor (g/t)</b>	<b>Onças contidas</b>
<b>P1</b>	2.480,4	2,77	6.870,7	1,50	331,3
<b>P2</b>	1.634,9	2,77	4.528,7	1,01	147,1
<b>P3</b>	1.207,4	2,77	3.344,6	0,53	57,0
<b>P4</b>	754,8	2,77	2.090,9	0,75	50,4
<b>P5</b>	1.714,0	2,77	4.747,8	1,58	241,2
<b>P6</b>	1.357,2	2,77	3.759,4	1,12	135,4
<b>P7</b>	1.754,0	2,77	4.858,5	1,19	185,9
<b>P8</b>	1.169,4	2,77	3.239,2	0,78	81,2
<b>P9</b>	2.414,0	2,77	6.686,8	1,01	217,1
<b>P10</b>	1.442,4	2,77	3.995,4	1,01	129,7
<b>P11</b>	1.672,2	2,77	4.632,1	1,77	263,6
<b>P12</b>	1.274,5	2,77	3.530,4	1,35	153,2
<b>P13</b>	1.848,2	2,77	5.119,5	0,90	148,1
<b>P14</b>	1.809,1	2,77	5.011,1	1,48	238,4
<b>P15</b>	1.199,0	2,77	3.321,3	1,08	115,3
<b>P16</b>	2.178,2	2,77	6.033,7	1,49	289,0
<b>P17</b>	1.062,6	2,77	2.943,3	1,58	149,5
<b>P18</b>	1.514,0	2,77	4.193,8	1,11	149,7
<b>P19</b>	3.231,7	2,77	8.951,8	0,78	224,5
<b>P20</b>	1.339,3	2,77	3.709,8	1,22	145,5
<b>P21</b>	1.604,4	2,77	4.444,3	1,42	202,9
<b>P22</b>	1.514,4	2,77	4.194,9	1,04	140,3
<b>P23</b>	2.239,1	2,77	6.202,4	2,17	432,7
<b>P24</b>	1.757,2	2,77	4.867,4	1,12	175,3
<b>P25</b>	1.583,5	2,77	4.386,3	1,37	193,2
<b>P26</b>	1.313,2	2,77	3.637,6	1,03	120,5
<b>P27</b>	1.354,7	2,77	3.752,6	1,16	140,0
<b>P28</b>	1.435,8	2,77	3.977,1	1,95	249,3
<b>P29</b>	1.753,2	2,77	4.856,3	0,98	153,0
<b>P30</b>	1.794,8	2,77	4.971,6	1,21	193,4
<b>P31</b>	2.607,0	2,77	7.221,5	0,76	176,5
<b>P32</b>	1.312,4	2,77	3.635,4	1,25	146,1
<b>P33</b>	895,4	2,77	2.480,2	1,24	98,9
<b>P34</b>	1.463,2	2,77	4.053,1	0,92	119,9
<b>P35</b>	1.481,3	2,77	4.103,2	1,61	212,4
<b>P36</b>	1.905,4	2,77	5.278,1	0,79	134,1
<b>P37</b>	1.724,3	2,77	4.776,2	1,07	164,3
<b>P38</b>	1.171,9	2,77	3.246,2	1,02	106,5
<b>P39</b>	898,6	2,77	2.489,2	0,00	0,0
		<b>Total</b>	174.142,4	1,18	6.612,2
		<b>Min</b>	2.090,9	0,00	0,0
		<b>Max</b>	8.951,8	2,17	432,7
		<b>Desv. Pad.</b>	1379,7	0,39	78,5

Fonte: Autoria própria.



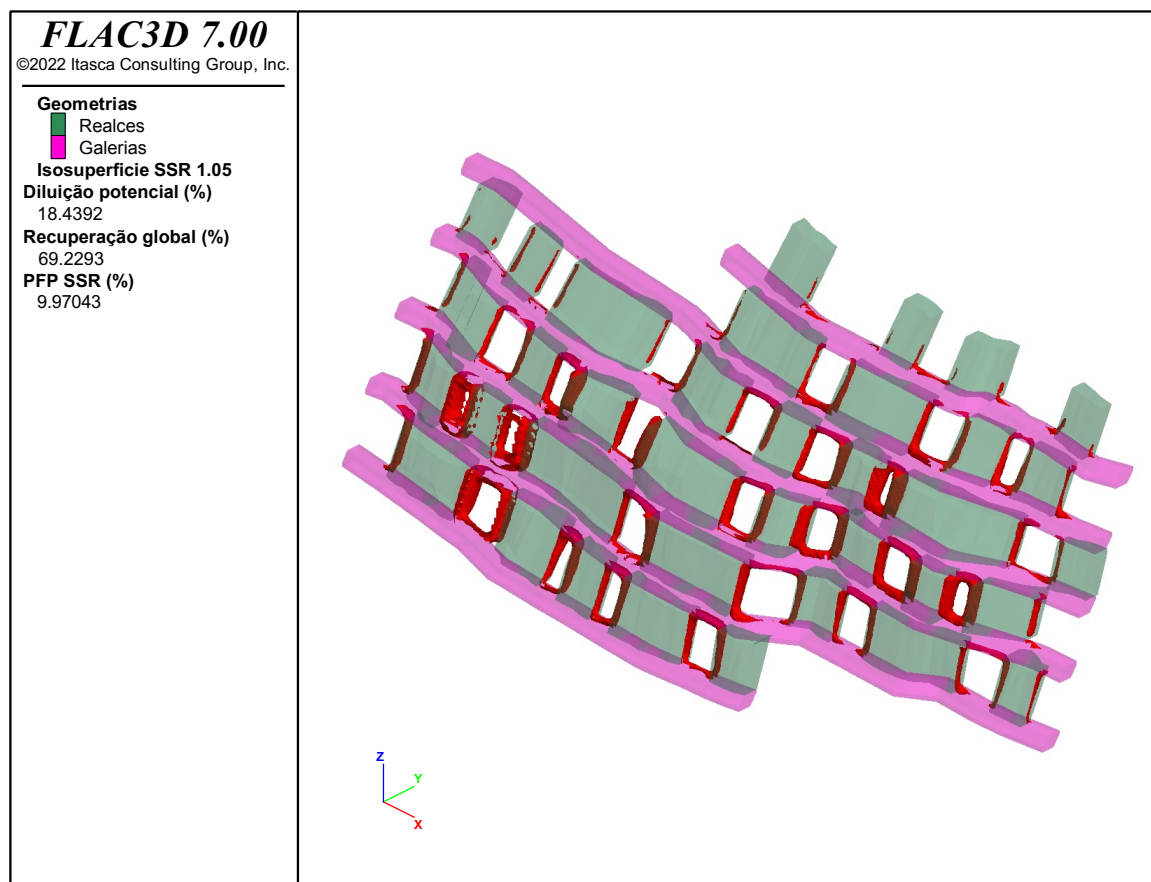
A diluição potencial obtida para o cenário foi de 18,4% (Figura 65), sendo a diluição real de 12,5%, considerando eficiência de 70% para o suporte. Por sua vez, a PFP foi de 9,97% (Figura 66). A recuperação global foi de 69,2%, sendo a tonelagem de minério 256.596,0 t com teor médio de 1,48 g/t, totalizando 12.209 onças contidas. O lucro líquido obtido foi de US\$ 3.249.039,58 e quando desconsiderado o uso de suporte no *hangingwall*, ou seja, atribuindo-se uma eficiência de suporte de 0%, o lucro líquido resulta em US\$ 2.517.288,91.

Figura 65 - Vista isométrica com zonas de falha no *hangingwall* em vermelho, para o cenário base otimizado.



Fonte: Autoria própria.

Figura 66 - Vista isométrica com zonas de falha nos pilares em vermelho, para o cenário base otimizado.



Fonte: Autoria própria.

De maneira geral o que se observa ao se comparar o cenário ótimo, obtido pela empresa considerando a metodologia tradicional, e o cenário ótimo obtido pela nova metodologia proposta é de que ainda que o lucro líquido seja menor no caso da metodologia proposta, a diluição potencial foi minimizada, assim como a PFP, ao mesmo tempo que preserva o mesmo teor médio. A Tabela 10 a seguir mostra a comparação direta entre os resultados das metodologias distintas.

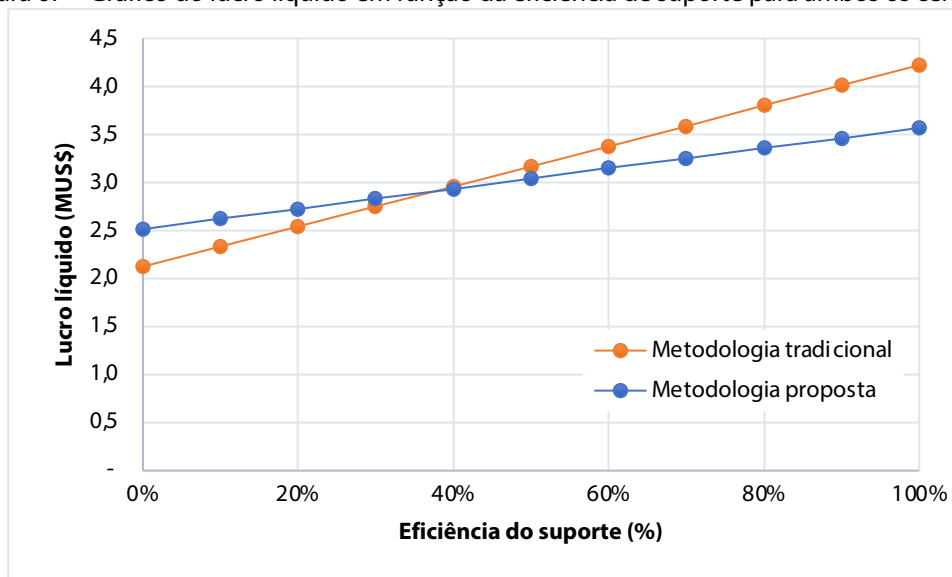
Tabela 10 – Comparação entre o resultado do cenário ótimo pela metodologia tradicional e do cenário ótimo obtido pela metodologia proposta.

Cenário ótimo	PPP (%)	Diluição potencial (%)	Rec. global (%)	Ton	Teor (g/t)	Onças contidas	Eficiência do suporte (%)	Diluição real (%)	Lucro líquido (US\$)
<b>Metodologia tradicional</b>	11,7	24,3	77,1	304.860,0	1,48	14.505	70	14,3	3.588.343,6
							0	24,3	2.120.361,8
<b>Metodologia proposta</b>	10,0	18,4	69,2	256.596,0	1,48	12.209	70	12,5	3.249.039,6
							0	18,4	2.517.288,9

Fonte: Autoria própria

Quando se considera uma diminuição progressiva na eficiência do suporte do *hangingwall*, ou ainda, quando se considera a não utilização de suporte, pode se observar que o cenário otimizado pela metodologia proposta tem melhor resultado econômico para eficiência de suporte abaixo de 40%. A Figura 67 a seguir mostra este efeito, observando se o lucro líquido em função da eficiência de suporte para ambos os cenários.

Figura 67 – Gráfico do lucro líquido em função da eficiência de suporte para ambos os cenários.



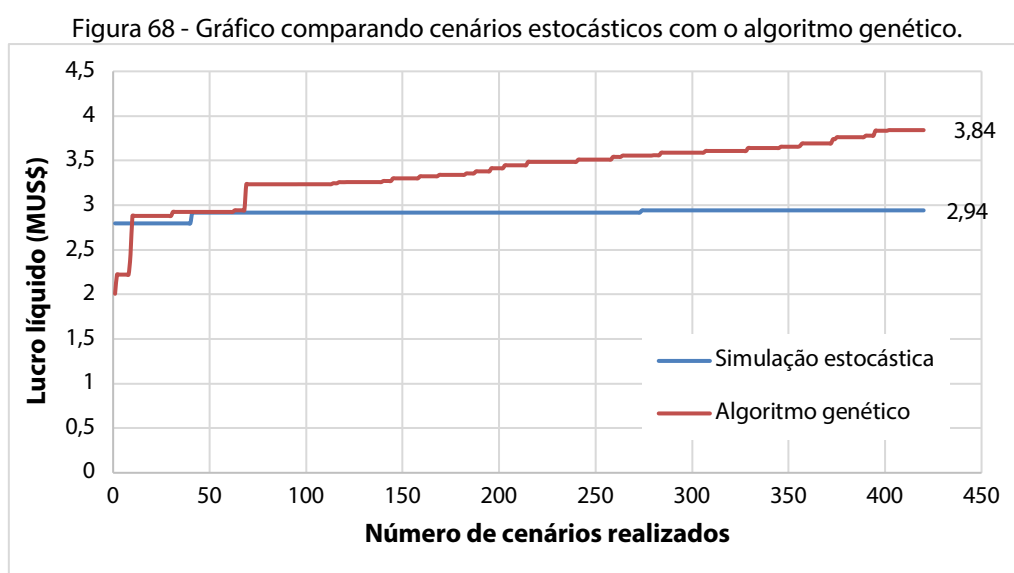
Fonte: Autoria própria.

### 4.2.3 Eficiência e tempos de otimização

A eficiência computacional do algoritmo genético de otimização, pode ser verificada comparando-se o resultado econômico a cada modelo numérico realizado ao longo do tempo, com cenários estocásticos válidos, isto é que respeitam os parâmetros limites mínimos e máximos das geometrias. Utilizando-se o próprio algoritmo foi possível realizar

os cenários estocásticos, sendo definido o valor de 420 para a população, e número de iterações 0 para que o algoritmo finalizasse sua operação sem inicializar o algoritmo genético, mas que dessa forma fossem produzidos 420 cenários aleatórios. No algoritmo genético, para que sejam produzidos 420 cenários, são gerados inicialmente 20 cenários aleatórios e 400 cenários correspondendo aos descendentes do cruzamento entre os pais, ou seja, 200 novas gerações.

No gráfico da Figura 68 é possível verificar que o lucro líquido para o melhor cenário obtido pelo algoritmo genético é 31% maior que o das simulações estocásticas de cenários, realizadas no mesmo intervalo de tempo.



Fonte: Autoria própria.

Os tempos de otimização são influenciados pelos parâmetros adotados na simulação numérica, principalmente quanto ao tamanho da geometria dos domínios e quanto ao nível de subdivisão pelo método *octree* que definido. Outro parâmetro que influencia no tempo de otimização é o valor adotado para o *unbalanced forces*. O valor de *unbalanced forces* representa o somatório das forças resultantes do modelo, que por padrão do FLAC3D é atribuído em  $1 \times 10^{-5}$ , ou seja, quando atingido o valor definido o modelo pode ser considerado em equilíbrio.

Testes realizados mostraram que um valor de *unbalanced forces* de  $1 \times 10^{-3}$  é capaz de representar bem a situação de equilíbrio, sem grandes diferenças numéricas quanto ao SSR dos pilares, mas com uma diferença um mais significativa no volume de diluição verificado.

Isto ao longo de vários modelos numéricos realizados, representa um grande ganho de tempo computacional. A Tabela 11 a seguir mostra os tempos computacionais obtidos para diferentes valores de *unbalanced forces* para um mesmo modelo comparado, utilizando como base um computador com processador Intel Core i7 (i7-4870HQ) 2.5 GHz. A diferença do tempo nos modelos chega a ser de 559%, enquanto a diferença na diluição obtida é de 49% e na identificação do volume de falha nos pilares é de apenas 17%, ou seja, a variação na diluição passa a ser aceitável em comparação com o ganho de tempo computacional.

Tabela 11 – Diferença de tempos computacionais conforme valor de *unbalanced forces* adotado.

<i>unbalanced forces</i>	Tempo de execução para cada modelo (min)	Tempo total de execução para 200 modelos (min)	Varição no tempo (%)	Varição no volume de diluição (%)	Varição no volume de falha de pilares (%)
$1 \times 10^{-3}$	4,2	834,0	-	-	-
$1 \times 10^{-4}$	13,0	2.604,0	212%	45%	16%
$1 \times 10^{-5}$	27,5	5.496,0	559%	49%	17%

Fonte: Autoria própria

Da mesma forma também se avaliou a influência da forma de escavação, se instantânea ou gradual, no tempo computacional e no resultado obtido. Utilizando se o modo de escavação gradual (*relax excavation*) o tempo de execução de cada modelo aumentou para 36,5 min, considerando *unbalanced forces* de  $1 \times 10^{-4}$ , em comparação com o tempo de 13 min para escavação instantânea. Por sua vez, os valores de diluição e de PFP para o cenário com escavação instantânea, apresentou acréscimo de 6,7% quando comparado com o cenário de escavação gradual.

#### 4.3 Sensibilidade do algoritmo

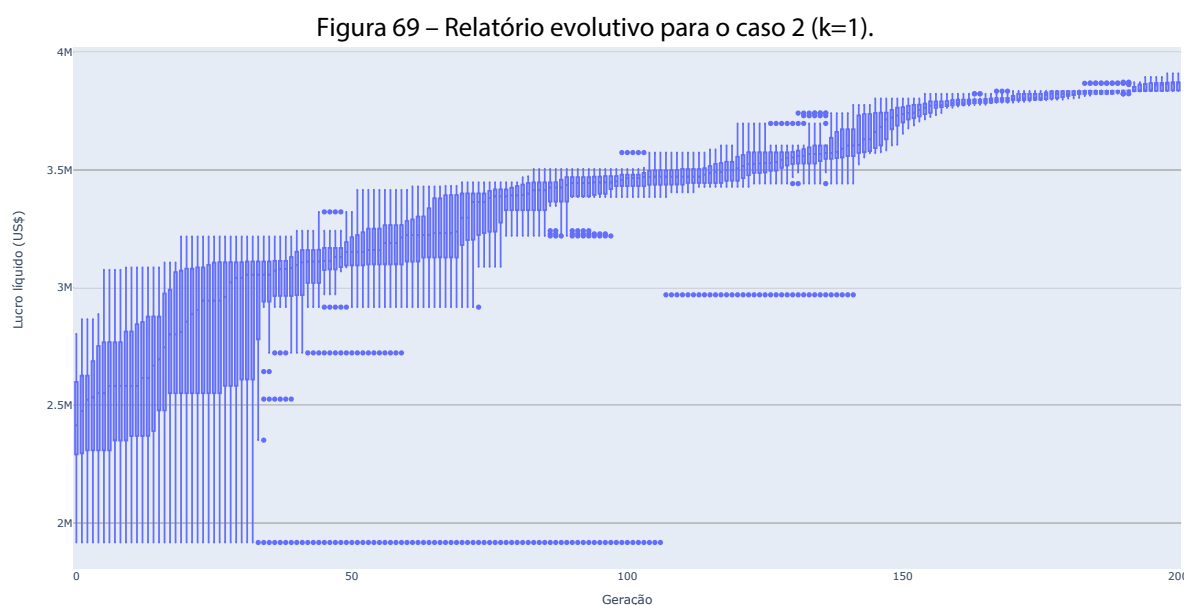
A sensibilidade do algoritmo proposto pode ser verificada fazendo-se modificações por exemplo, nos parâmetros de tensões *in situ*, simulando situações de estudos de caso com profundidades diferentes. A inclusão de subdomínios geotécnicos também pode ser realizada representando as variações das propriedades geomecânicas ao longo do depósito.

##### 4.3.1 Parâmetros de tensões *in situ*

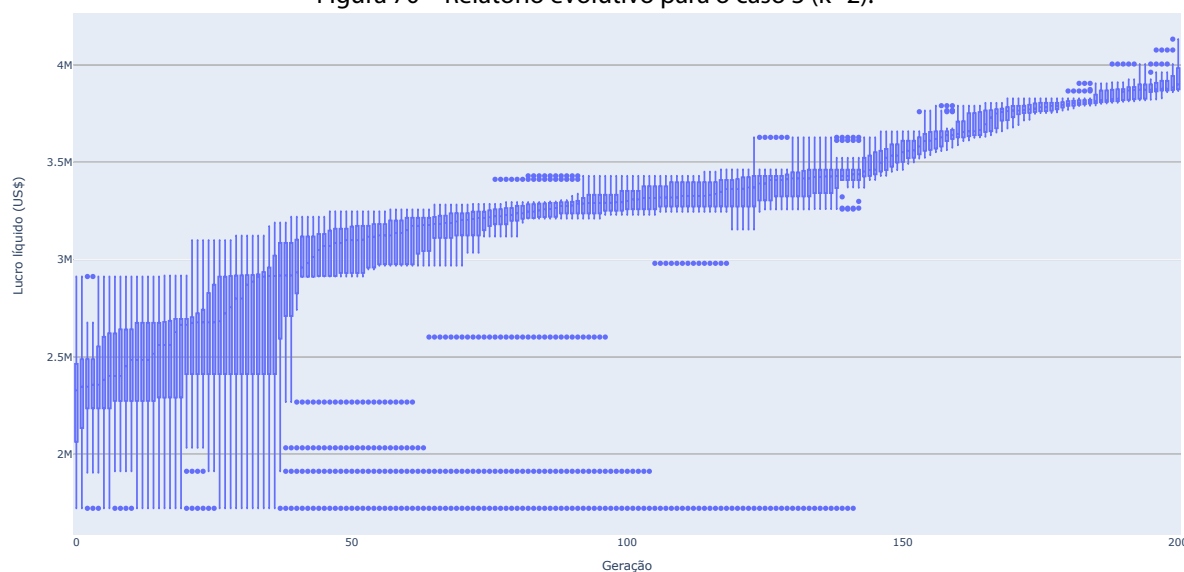
Com objetivo de verificar a sensibilidade dos modelos otimizados, três cenários distintos foram otimizados e comparados com o cenário 1 (base). Dois dos cenários

consideraram a variação do fator ( $k$ ), razão entre tensões horizontais ( $\sigma_h$ ) e verticais ( $\sigma_v$ ), para  $k = 1$  e para  $k = 2$ . O último cenário simulado considera a variação da cobertura (*overburden*) para uma tensão de 20 MPa, o que corresponde a uma cobertura média de 740 m, simulando uma situação em que o depósito está mais profundo e conseqüentemente sob maior magnitude de tensões verticais.

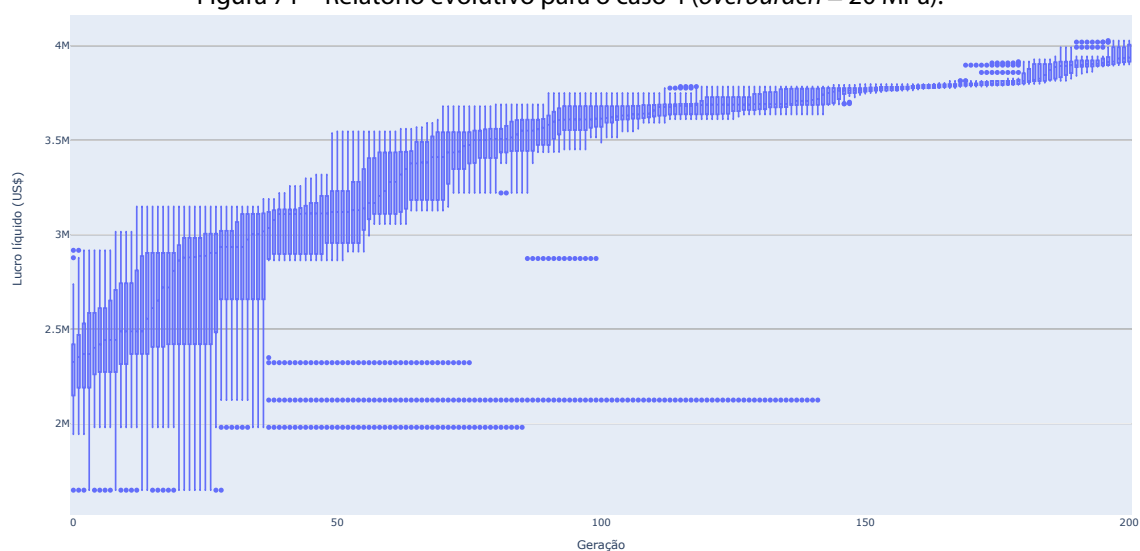
Os relatórios do processo evolutivo, em gráficos do tipo *boxplot*, podem ser visualizados nas figuras Figura 69, Figura 70 e Figura 71 a seguir. Importante salientar que os relatórios apresentam os valores de lucro líquido considerando uma eficiência de suporte de 90% e foram obtidos com o somatório de forças em  $1 \times 10^{-3}$ . Após obtenção do modelo ótimo, o arranjo obtido foi novamente analisado com somatório de forças de  $1 \times 10^{-5}$  e a eficiência de suporte considerada em 70%, resultado em um menor lucro líquido.



Fonte: Autoria própria.

Figura 70 – Relatório evolutivo para o caso 3 ( $k=2$ ).

Fonte: Autoria própria.

Figura 71 – Relatório evolutivo para o caso 4 ( $overburden = 20$  MPa).

Fonte: Autoria própria.

O resultado obtido quanto ao retorno econômico destes cenários, tamanho médio de pilares e de realces pode ser comparado com o cenário base a seguir, para os casos foi considerada uma eficiência de suporte de 70% (Tabela 12).

Tabela 12 – Comparativo de cenários otimizados, considerando variações das tensões *in situ*.

Caso	Melhor cenário otimizado				
	Lucro	Tamanho	Tamanho	Diluição	PFP (%)
	líquido (US\$)	médio dos pilares (m)	médio dos realces (m)	potencial (%)	
<b>Caso 1 (base) – (k = 1,3 e <i>overburden</i> = 10 MPa)</b>	3.249.039,6	7,9	11,8	18,4	9,9
<b>Caso 2 (k = 1 e <i>overburden</i> = 10 MPa)</b>	3.497.879,1	6,9	16,1	25,2	11,8
<b>Caso 3 (k = 2 e <i>overburden</i> = 10 MPa)</b>	3.619.381,4	7,7	11,4	16,9	10,2
<b>Caso 4 (k = 1,3 e <i>overburden</i> = 20 MPa)</b>	3.795.446,3	8,5	14,2	21,0	8,7

Fonte: Autoria própria.

Pode se verificar que o lucro líquido para os cenários analisados, tende a permanecer relativamente constante, ainda que se observe que o cenário com maior *overburden* apresenta maior lucro líquido, dentre os casos analisados. Isso ocorre pois, o algoritmo genético compensa as variações de tensões em cada caso, selecionando cenários com tamanho médio de realces e pilares maiores ou menores, e conseqüentemente variando o número total de pilares e realces obtidos, sempre em função de um melhor resultado econômico.

Para o caso 4, com maior *overburden*, se observa que o tamanho médio dos pilares teve um acréscimo de 7,5% em relação ao caso base. O tamanho médio dos realces, por sua vez, teve um acréscimo de 20%, levando a um aumento de 14% na diluição potencial do cenário. O aumento do tamanho dos pilares está relacionado com o aumento da tensão vertical atuantes nos mesmos, sendo necessária uma maior área de pilar para que sejam capazes de suportar as solicitações, sem que ocorra aumento de elementos em situação de falha, como se observa pelo valor de PFP inferior ao do caso base. Conseqüentemente, o número de pilares diminuiu de 39 para 33 e o número de realces de 36 para 30, respectivamente em comparação com cenário base.

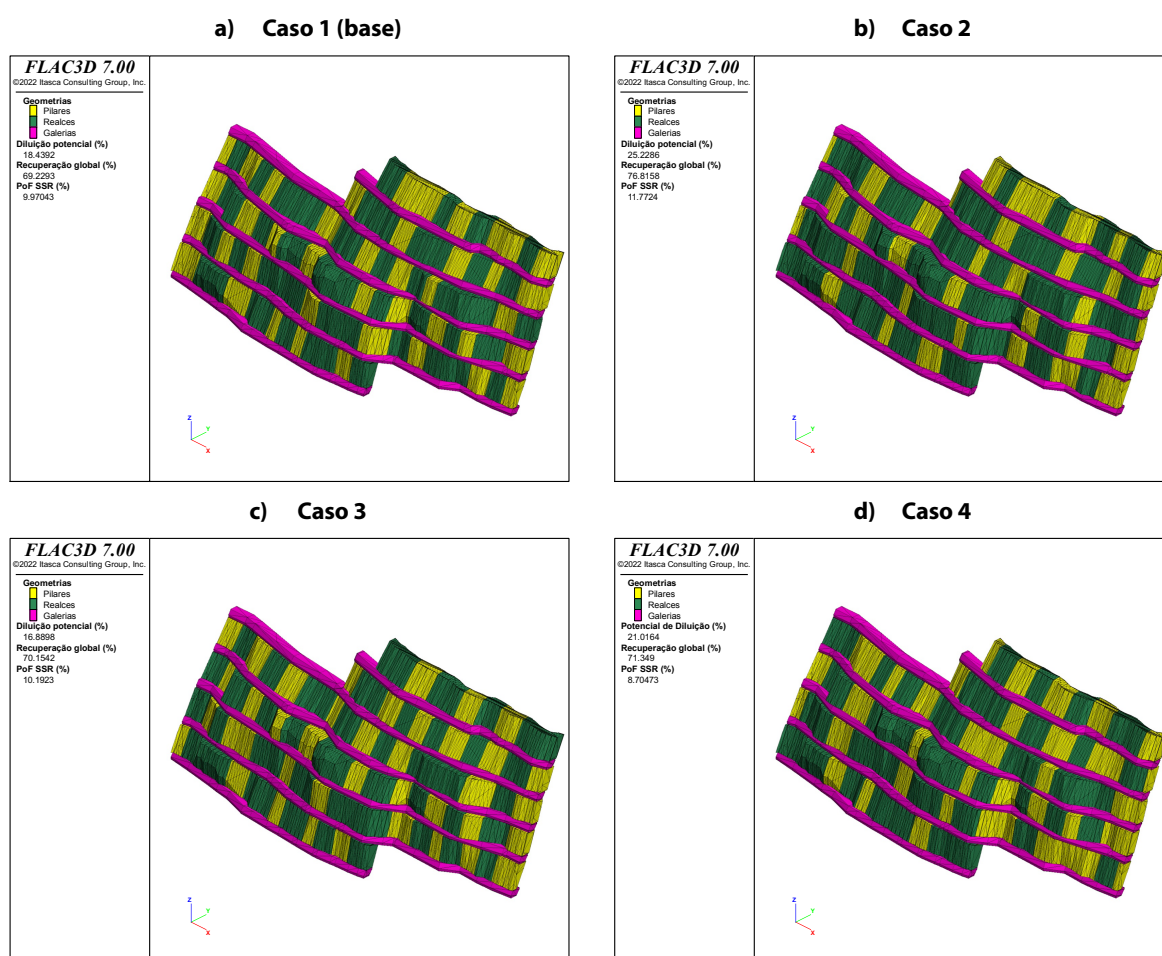
Para o caso 3, se observa que ocorre uma leve diminuição tanto no tamanho médio dos pilares, em 2,5%, quanto no tamanho médio dos realces, em 3,5%, ainda sim ocorre uma diminuição, de 8,2%, na diluição potencial, pois com o valor de  $k=2$  ocorre o aumento da magnitude das tensões horizontais, resultando em uma diminuição no alívio de tensões no *hangingwall*.

Por sua vez, para o caso 2 com valor de  $k=1$ , situação isotrópica de tensões, se observa que o tamanho médio de pilares no cenário ótimo diminuiu em 12,7% e o tamanho médio de realces diminuiu em 26,7% em comparação com o caso base. Já o número de



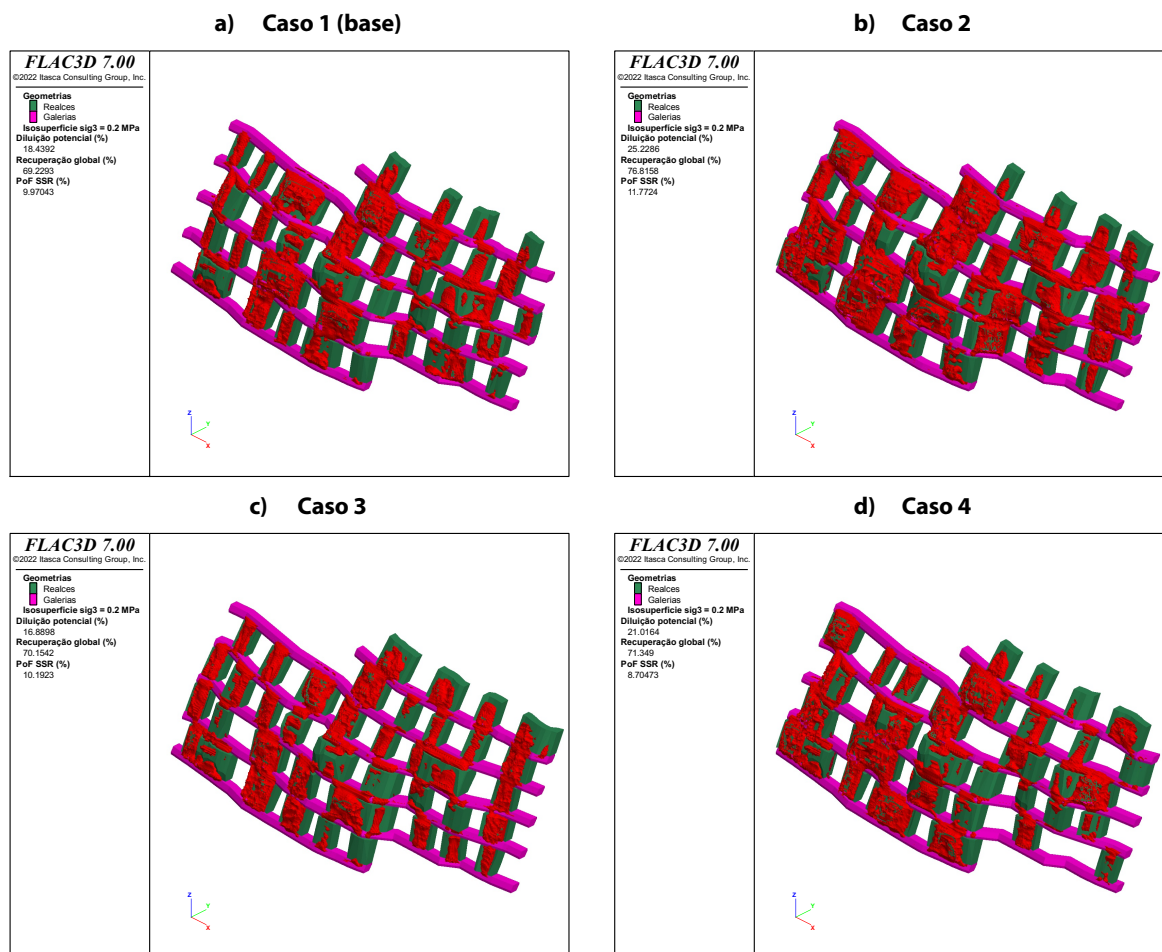
realces diminuí de 36 para 31 e o número de pilares de 39 para 34, respectivamente em comparação com o cenário base. Porém, se observa que mesmo com a diminuição do tamanho médio dos realces, a diluição potencial teve um acréscimo substancial de 37%, o que está relacionado com o aumento do alívio de tensões de confinamento no *hangingwall*, em função da diminuição do  $k$ , de 1,3 para 1 no cenário em questão. As Figura 72, Figura 73 e Figura 74, mostram as geometrias finais, as isosuperfícies do potencial de diluição e o desempenho dos pilares para os 4 casos analisados.

Figura 72 – Vista isométricas para o arranjo de realces e pilares otimizado para os 4 casos considerados.



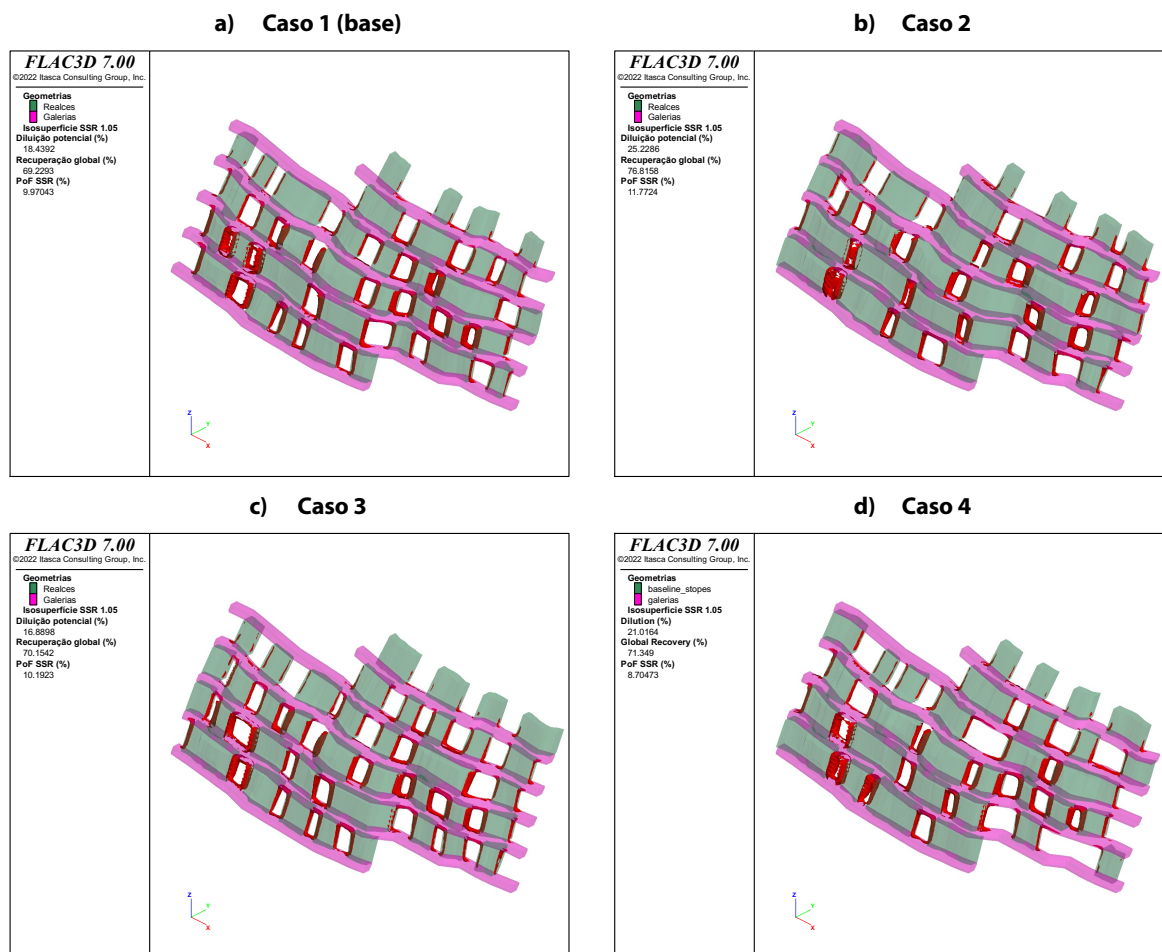
Fonte: Autoria própria.

Figura 73 – Vistas isométricas com as isosuperfícies do potencial de diluição no *hangingwall*, em vermelho, para os cenários ótimos dos 4 casos considerados.



Fonte: Autoria própria.

Figura 74 – Vistas isométricas com as isosuperfícies de zonas de falha nos pilares, em vermelho, para os cenários ótimos dos 4 casos considerados.



Fonte: Autoria própria.

### 4.3.2 Inclusão de subdomínios geotécnicos

Variações nas propriedades geomecânicas são comuns devido à complexa configuração geológica e eventos estruturais de uma determinada região. Além das variações nas propriedades do maciço rochoso, o modelo constitutivo considerado na análise numérica tem efeito nos resultados econômicos. Portanto, o uso de subdomínios geotécnicos pode ser necessário para capturar variações significativas nas propriedades do maciço rochoso.

Com objetivo de verificar a utilização de subdomínios geotécnicos, foi considerado um subdomínio dentro da zona mineralizada, definido como subdomínio 1 (OZM), que representa uma zona de menor resistência dentro da zona mineralizada. A propriedades consideradas para o subdomínio 1 são mostradas, em conjunto das propriedades da zona mineralizada, na Tabela 13 a seguir.

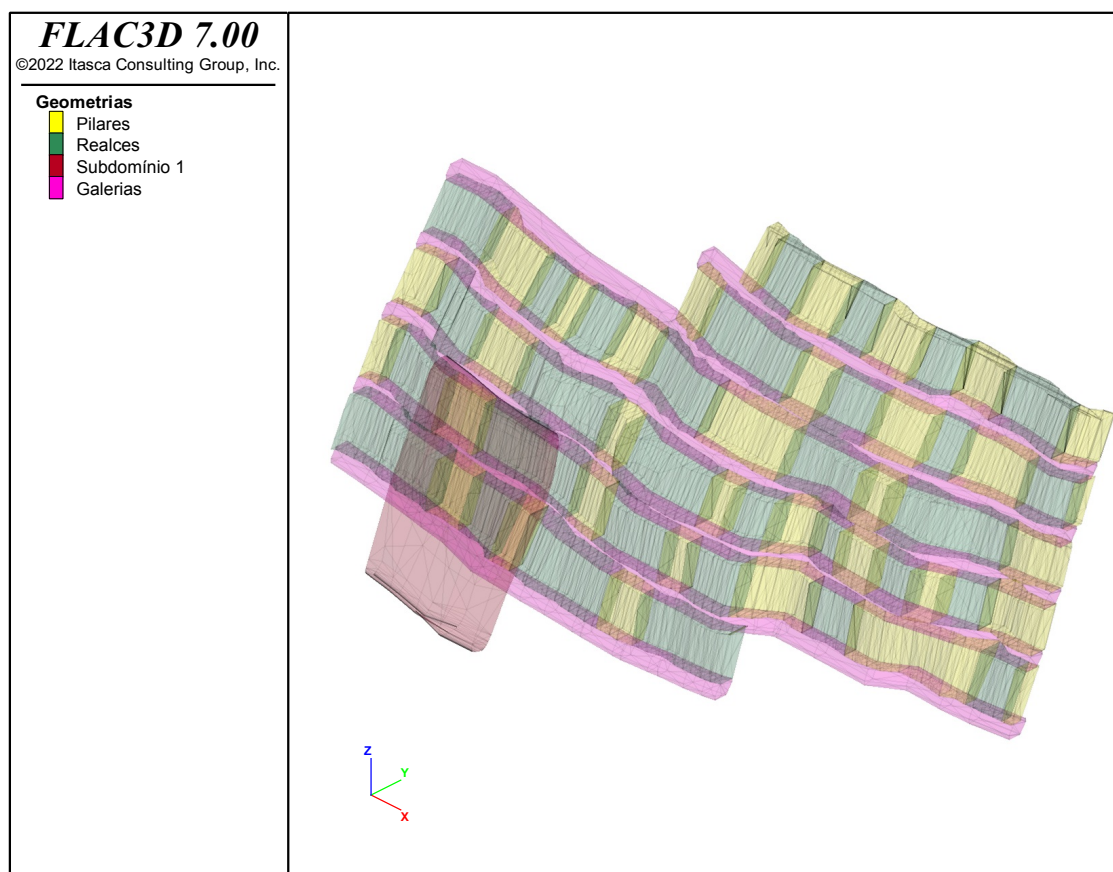
Tabela 13 - Propriedades geomecânicas consideradas para o estudo de caso.

Subdomínio	Peso específico (t/m <sup>3</sup> )	GSI	Young (GPa)	Poisson	B (GPa)	G (GPa)	Phi (°)	c (MPa)
Orezone	2,77	70	46,41	0,20	26,16	19,27	30,95	9,03
1 (OZM)	2,70	60	42,99	0,22	25,99	17,56	28,31	3,75

Fonte: Autoria própria.

A geometria considerada para o subdomínio também pode ser verificada na Figura 75 a seguir.

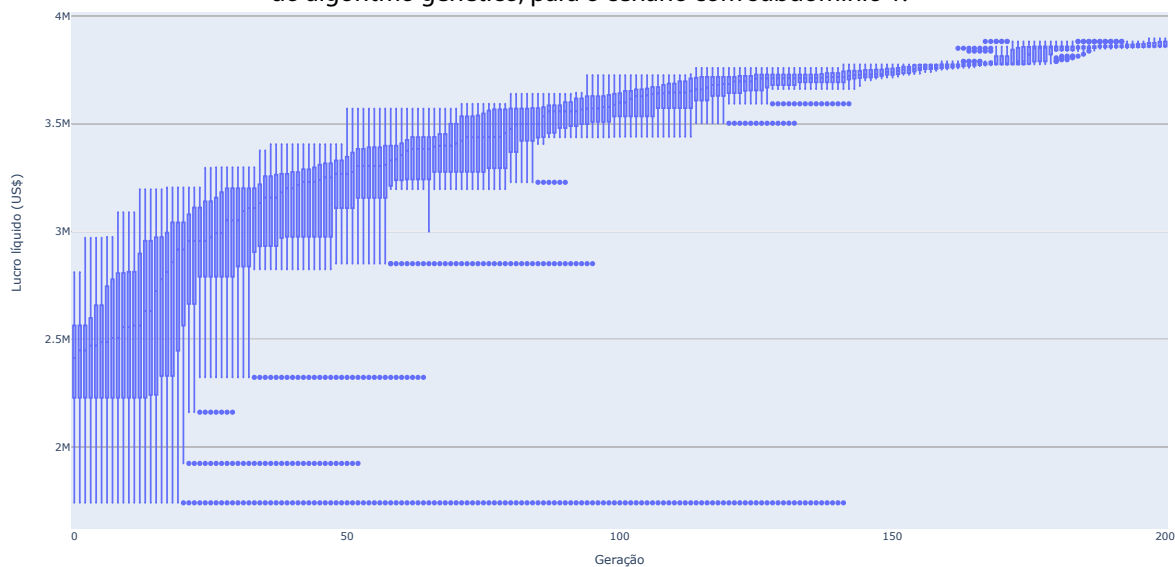
Figura 75 – Vista isométrica mostrando a zona do subdomínio 1 considerado.



Fonte: Autoria própria.

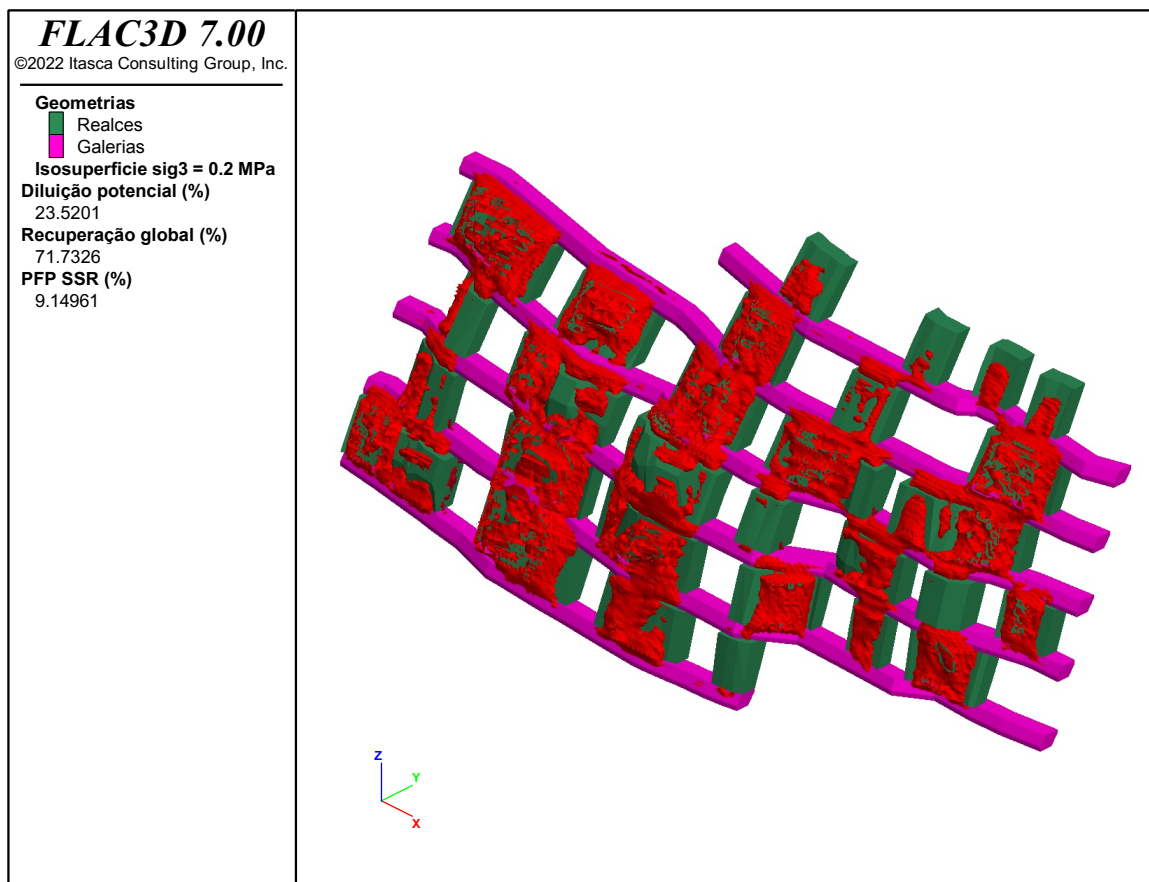
Após 200 gerações o cenário ótimo foi obtido. A Figura 76 a seguir demonstra o relatório evolutivo. A diluição potencial obtida para o cenário foi de 23,5%, sendo a diluição real de 14,1%, considerando eficiência de 70% para o suporte. A Figura 77 mostra uma vista isométrica contendo as superfícies de potencial de diluição no *hangingwall*.

Figura 76 - Gráfico do tipo diagrama de caixas, mostrando a evolução da população ao longo das gerações do algoritmo genético, para o cenário com subdomínio 1.



Fonte: Autoria própria.

Figura 77 - Vista isométrica com zonas de falha no hangingwall em vermelho, para o cenário otimizado com o subdomínio 1 na zona mineralizada.



Fonte: Autoria própria.

Por sua vez, a PFP foi de 9,15% (Figura 78). A recuperação global foi de 71,7%, sendo a tonelagem de minério 270.921,7t com teor médio de 1,51 g/t, totalizando 13.144,7 onças contidas.

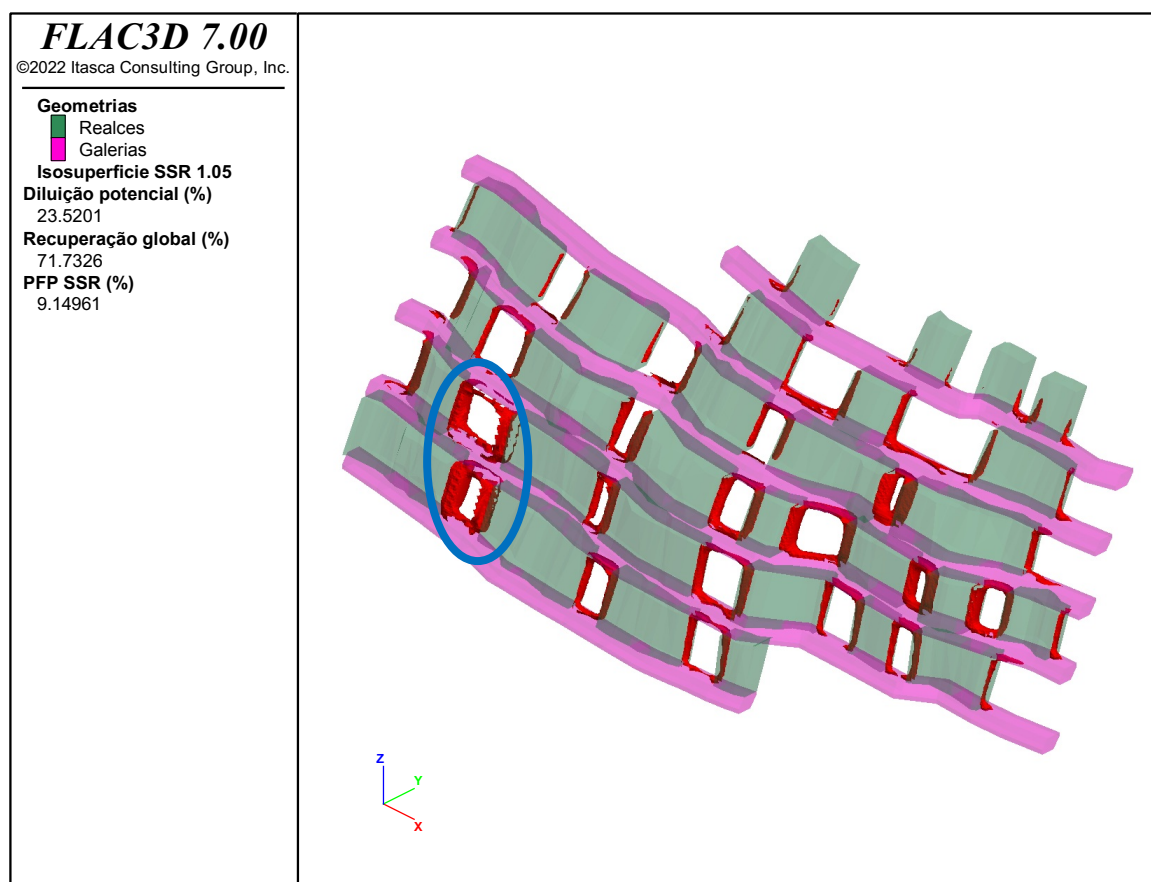
O lucro líquido obtido foi de US\$ 3.577.648,2 e quando desconsiderado o uso de suporte no *hangingwall*, ou seja, atribuindo-se uma eficiência de suporte de 0%, o lucro líquido resulta em US\$ 2.341.673,6. Se observa que o cenário obtido, ainda que considerando um subdomínio de menor qualidade do maciço rochoso, tem um lucro líquido bem próximo, mas que inclusive supera o cenário base, que não considera o subdomínio.

O arranjo ótimo de realces e pilares obtido, dados os parâmetros de entrada, pode ser observado na Figura 78. De maneira geral se observa que o arranjo obtido é irregular, com tamanhos variados de realces e pilares. Dentro da zona do subdomínio se observa que os pilares apresentam maior volume de elementos com falha, quando comparados com o cenário base sem a consideração do subdomínio.

Foram definidos um total de 29 realces e 33 pilares para o cenário ótimo, sendo o tamanho médio para os pilares de 8,8m e o tamanho médio dos realces de 15,3m. Se verifica que o tamanho médio de pilares é quase 1m maior em comparação com o cenário base, sem o subdomínio considerado, o que corresponde a um acréscimo de 11%.



Figura 78 - Vista isométrica com zonas de falha nos pilares em vermelho, para o caso com um subdomínio de menor resistência dentro da zona mineralizada. Os pilares com maior número de elementos com falha, dentro do subdomínio, estão circulados em azul.



Fonte: Autoria própria.

# Capítulo 5

## 5 Considerações finais

Neste capítulo final são apresentadas as contribuições obtidas na tese, são discutidas as limitações da metodologia proposta e são comentadas as variações possíveis do algoritmo/metodologia em trabalhos futuros.

### 5.1 Contribuições da tese

Considerando a meta proposta neste trabalho, foi desenvolvida uma metodologia integrada de otimização econômica do arranjo dos realces e dos pilares, com o desenvolvimento de um algoritmo evolutivo, incorporando os critérios estabelecidos, como tamanho mínimo dos realces, tamanho mínimo dos pilares, diluição máxima esperada e porcentagem máxima aceitável de falha nos pilares.

Considera-se que o trabalho conseguiu demonstrar com êxito a real necessidade de realização da otimização geomecânica integrada da otimização econômica. Isso fica evidente quando observado que o cenário otimizado pela metodologia proposta apresentou o mesmo teor médio do cenário pelo método tradicional, porém com menor potencial de diluição e porcentagem de falhas nos pilares.

A metodologia proposta também foi capaz de proporcionar uma interface direta entre os tradicionais pacotes de *software* de mineração e o software de modelagem numérica FLAC3D, utilizando linguagem *Python*.

A eficiência do algoritmo evolutivo foi analisada, ao compará-lo com simulações estocásticas de cenários e com o próprio cenário ótimo definido pela empresa. A sensibilidade do algoritmo também foi avaliada, a partir das variações dos parâmetros de entrada, como tensões *in situ* e inclusão de domínios geomecânicos, resultando em cenários otimizados com bom retorno econômico, variando-se naturalmente as dimensões ótimas dos pilares e realces e suas localizações.

### 5.2 Limitações da metodologia proposta

Uma das grandes limitações observadas no desenvolvimento da metodologia, foi o elevado tempo computacional devido ao grande número de modelos numéricos necessários para que um cenário ótimo fosse obtido. Após o cruzamento de dois indivíduos



(pais), dois indivíduos (filhos) são gerados, e estes indivíduos precisam ser analisados pelo modelo numérico. Na metodologia construída, estes modelos são analisados sequencialmente e não paralelamente.

A barreira técnica encontrada para a realização de modelos simultâneos, ocorre devido a impossibilidade de se estabelecer mais de uma conexão simultânea via *socket* entre o Python e o FLAC3D, o que poderia permitir que duas sessões separadas do FLAC3D fossem utilizadas, processando os dois modelos de maneira simultânea em um computador com elevado número de núcleos de processamento. Essa paralelização do cálculo poderia ser realizada desta forma, com duas sessões rodando no mesmo computador, ou ainda, utilizando-se de mais de um computador conectado em rede, com outra licença do software FLAC3D, para rodar o segundo modelo simultaneamente.

Outra limitação da metodologia proposta é a ausência da possibilidade de se considerar um teor de diluição. Considerando que é possível que o material de diluição tenha um teor maior do que zero, seria necessário que o FLAC3D possuísse o modelo de blocos com os teores, e a partir da geometria do potencial de diluição obtida, efetuasse o cálculo do teor médio da diluição. Outra forma seria que ao se obter a geometria de diluição, o algoritmo enviasse esta ao software de mineração, para que então seja realizado o cálculo do teor médio, uma vez que o FLAC3D não possui uma função como essa implementada de forma nativa.

Por fim, também se considera que o resultado ótimo obtido pode chegar a apresentar localmente realces e pilares com desempenho insatisfatório e abaixo da média com relação aos demais. Ainda que com um elevado número de gerações estabelecidas se espera que o algoritmo naturalmente consiga lidar com essas questões, mas que na prática não há nada implementado que faça essa verificação explicitamente.

### 5.3 Variações do algoritmo em trabalhos futuros

Considerando as etapas do ciclo de desenvolvimento dos pacotes de *software*, estas são divididas em três: *alfa*, *beta* e *gold*. Esta versão 1.0 do software pode ser classificada como atualmente no estágio *Beta*, onde o software já possui implementado as funcionalidades propostas e consegue ser executado de maneira relativamente estável. Alguns erros (*bugs*) e travamentos podem ainda ocorrer. Com a continuidade de testes, o

*software* passaria para um estágio *gold* onde então poderia ser disponibilizado ao usuário final, com as mesmas funcionalidades originais.

Para uma versão 2.0 do *software*, novas funcionalidades poderão ser implementadas, dentre elas:

- Adoção de modelos constitutivos diversos, comportamento plástico;
- Possibilidade de escolha de diferentes critérios de falha para os pilares e para o *hangingwall* dos realces;
- Otimização em depósitos poli metálicos;
- Otimização multiobjetivo, priorizando além do retorno econômico, quantidade de metal, recuperação global, etc;
- Otimização para recuperação de pilares com uso de *backfill*;
- Otimização da distância entre subníveis;
- Simulação de cenários, considerando as incertezas de entrada.

Para a implementação destas novas funcionalidades, o *software* retornaria a um estágio *Beta* até que pudesse novamente ser considerado em estágio *gold*, livre de *bugs* conhecidos, para liberação ao usuário final.

Outra possibilidade considerada para uma versão futura, seria a utilização de algoritmos de modelagem numérica diretamente no *Python*, sem a necessidade de utilização do FLAC3D. Dessa forma o algoritmo se tornaria independente de pacotes de *software* comerciais e os problemas relacionados a tempo computacional possivelmente seriam mitigados com a inclusão da paralelização de processos.

## 6 Referências bibliográficas

- ALFORD, C.; BRAZIL, M.; LEE, D. H. Optimisation in underground mining. Em: Weintraub, A.; Romero, C.; Bjorndal, T.; Epstein, R., **Handbook of Operations Research in Natural Resources**, 1º ed, Berlin: Springer, 2007, pp 561-578.
- ALFORD MINING SYSTEMS (Austrália). **Mineable Shape Optimiser**. 2016. Disponível em: [http://alfordminingsystems.com/?page\\_id=74](http://alfordminingsystems.com/?page_id=74). Acesso em: 20 out. 2020.
- ANDRADE, A. B.; FARIA, A. R. C.; RAMPAZZO, P. C. B. Economic optimization of rib pillars placement in underground mines. Em: APCOM, 39th, 2019, Wroclaw, Poland, **Proceedings [...]**, London: CRC Press, 2019, p. 292-299. <http://dx.doi.org/10.1201/9780429320774-34>.
- APPIANING, E. J. A.; BEN-AWUAH, E. **Underground Mining Stope Layout Optimization and Production Scheduling: A Review of Existing Solvers and Algorithms**. Canada: Mining Optimization Laboratory, 2018.
- ATAEE-POUR, M. **A Heuristic Algorithm To Optimize Stope Boundaries**, 2000. Tese (Doutorado). University of Wollongong, New South Wales, março de 2000.
- ATAEE-POUR, M. A critical survey of the existing stope layout optimization techniques. **J. Min. Sci.**, 41, 447–466. setembro de 2005. <https://doi.org/10.1007/s10913-006-0008-9>
- BAI, X.; MARCOTTE, D.; SIMON, R. Underground stope optimization with network flow method. **Computer and Geosciences**. 52, 361–371. 2013
- BARTON, N.; LEIN, R.; LUNDE, J. Engineering classification of rock masses for design of tunnel support. **Rock Mechanics**, volume 6, p. 189–236. dezembro de 1974 <https://doi.org/10.1007/BF01239496>
- BECK, D. A.; DUPLANCIC, P. Forecasting performance and achieving performance indicators in high stress and seismically active mining environments. Em: International Symposium on Rockburst and Seismicity in Mines, 6<sup>th</sup>. 2005, Perth, Western Australia. **Proceedings [...]** Australia: Australian Centre for Geomechanics, 2005, pp. 409–417.
- BECK, D. A.; KASSBOHM, S.; PUTZAR, G. Multi-scale simulation of ground support designs for extreme tunnel closure. Em: Symposium on Block and Sublevel Caving, 2<sup>nd</sup>. 2010, Perth, Western Australia, **Proceedings [...]**, Perth: ACG, 2010, pp. 441–453.
- BIENIAWSKI, Z. T. Engineering classification of jointed rock masses. **Transactions Of The South African Institution Of Civil Engineers**. Johannesburg, p. 335-344. 1973.
- BIENIAWSKI, Z. T. **Engineering rock mass classification**. New York: John Wiley And Sons, 1989. 251 p.
- BOBET, A.; FAKHIMI, A.; JOHNSON, S.; MORRIS, J. Numerical Models in Discontinuous Media: Review of Advances for Rock Mechanics Applications. **J. Geotech. Geoenviron. Eng.**, v. 135, n. 11, p.1547-1561, nov. 2009. American Society of Civil Engineers (ASCE). Disponível em: <[http://dx.doi.org/10.1061/\(asce\)gt.1943-5606.0000133](http://dx.doi.org/10.1061/(asce)gt.1943-5606.0000133)>. Acesso em: 20 out. 2020.
- BRADY, B. H. G.; BROWN, E. T. **Rock Mechanics for Underground Mining**. 3. ed. New York: Springer Science + Business Media Inc, 2005. 647 p.
- BROWN, E. T.; HOEK, E. Trends in relationships between measured in situ stresses and depth. **International Journal of Rock Mechanics and Mining Sciences**. p. 211-215. ago. 1978. Disponível em: <[http://dx.doi.org/10.1016/0148-9062\(78\)91227-5](http://dx.doi.org/10.1016/0148-9062(78)91227-5)>. Acesso em: 08 abr. 2020.

CASTRO, L. A. M.; BEWICK, R. P.; CARTER, T. G. An overview of numerical modelling applied to deep mining. In: SOUSA, L. R.; VARGAS J. E.; FERNANDES, M. M.; AZEVEDO, R. (ed.). **Innovative numerical modelling in geomechanics**. Londres: CRC Press, 2012. Cap. 21. p. 393-414.

CAWRSE, I. Multiple Pass Floating Stope Process. In: Biennial Strategic Mine Planning Conference, 4<sup>th</sup>. 2001, Perth. **Proceedings [...]** Melbourne: The Australasian Institute of Mining and Metallurgy, 2001, pp. 87-94.

CELADA, B.; TARDAGUILA, I; VARONA P.; RODRIGUEZ A.; BIENIAWSKI Z. T. Innovating tunnel design by an improved experience-based RMR system. Em: World Tunnel Congress – Tunnels for a Better Life. Brasil: Foz do Iguaçu, **Proceedings [...]** 2014, pp 1–9.

CEPURITIS, P. M. **An Integrated Approach to Span Design in Open Stope Mining**. 2010. 473 f. Tese (Doutorado) - Doctor Of Philosophy, Department Of Mining Engineering And Mine Surveying, Western Australian School Of Mines, Austrália, 2010.

CHEIMANOFF, N. M.; DELIAC, E. P.; MALLET, J. L. GEOCAD: an alternative CAD and artificial intelligence tool that helps moving from geological resources to mineable reserves. Em: APCOM, 21st, EUA, **Proceedings [...]** 1989.

CLARK, L.; PAKALNIS, R. An empirical design approach for estimating unplanned dilution from open stope hangingwalls and footwalls. Em: 99th CIM-AGM, Vancouver. **Proceedings...** Calgary: CIM, 1997.

COLEY, D. **An Introduction to Genetic Algorithms for Scientists and Engineers**. Singapura: World Scientific Publishing Co. Pte. Ltd., 1999. 223 p.

CORDOVA, D. P. **Análise da diluição no método câmaras transversais utilizando modelagem numérica: estudo de caso mina Arenal Deeps**. 2016. 286 f. Dissertação (Mestrado) - Mestrado em Tecnologia Mineral, PPGE3M, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.

DATAMINE. **About Us**: Datamine through the years. 2020. Disponível em: <https://www.dataminesoftware.com/about-us/>. Acesso em: 20 out. 2020.

DENG, J.; YUE, Z. Q.; THAM, L. G.; ZHU, H. H. Pillar design by combining finite element methods, neural networks and reliability: a case study of the feng huangshan copper mine, china. **International Journal of Rock Mechanics and Mining Sciences**, [S.L.], v. 40, n. 4, p. 585-599, jun. 2003. Elsevier BV. [http://dx.doi.org/10.1016/s1365-1609\(03\)00042-x](http://dx.doi.org/10.1016/s1365-1609(03)00042-x).

DERAISME, J.; DE FOUQUET, C.; FRAISSE, H. Geostatistical orebody model for computer optimization of profits from different underground mining methods. Em: APCOM, 18th. 1984, Londres. **Proceedings [...]**. Londres. pp. 583–590.

DHAR, B. B. Advances in rock mechanics and future trends. Em: RECENT ADVANCES IN ROCK ENGINEERING (RARE 2016), Bengaluru. **Proceedings [...]** Índia: Atlantis Press, 2016. p. 1-2. Disponível em: <https://www.atlantis-press.com/proceedings/rare-16/25866284>. Acesso em: 20 out. 2020.

DIEDERICHS, M. S. **Instability of hard rock masses: the role of tensile damage and relaxation**. 1999. 610 f. Tese (Doutorado) - Departamento de Engenharia Civil, Universidade de Waterloo, Ontario, 2000.

DIMITRAKOPOULOS, R.; GRIECO, N. Stope design and geological uncertainty: quantification of risk in conventional designs and a probabilistic alternative. **Journal Of Mining Science**,

v. 45, n. 2, p. 152-163, mar. 2009. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s10913-009-0020-y>.

ERDOGAN, G.; CIGLA, M.; TOPAL, E.; YAVUZ, M. Implementation and comparison of four stope boundary optimization algorithms in an existing underground mine. **International Journal of Mining, Reclamation and Environment**, 31:6, 389-403, 2017, DOI: 10.1080/17480930.2017.1331083

FIGUEIREDO, R. P.; CURI, A. Otimização de recuperação em lavra por câmaras e pilares utilizando programação não-linear. Em: Congresso Brasileiro de Túneis e Estruturas Subterrâneas – International South American Tunnelling, 1º. 2003, São Paulo. **Anais [...]** São Paulo, 2003 – Artigo n. 44, 6 p.

FIGUEIREDO, R. P.; CURI, A. Dimensionamento ótimo de painéis, câmaras e pilares com programação não-linear. Em: Simpósio Ibero Americano de Engenharia de Minas, 1º, 2003 São Paulo. **Anais [...]** São Paulo, pp 565-573.

FOROUGH, S.; HAMIDI, J. K.; MONJEZI, M.; NEHRING, M. The integrated optimization of underground stope layout designing and production scheduling incorporating a non-dominated sorting genetic algorithm (NSGA-II). **Resources Policy**, [S.L.], v. 63, p. 1-11, out. 2019. Elsevier BV. <http://dx.doi.org/10.1016/j.resourpol.2019.101408>.

FURTNEY, J. **Itasca**: Pacote Python. Versão 2020.7.12. [S. l.], 10 jul. 2020. Disponível em: <https://pypi.org/project/itasca/>. Acesso em: 20 out. 2020.

GOLDBERG, D. **Genetic Algorithms in Search optimization and machine learning**. EUA: Addison-Wesley Publishing Company, 1989.

GRIECO, N.; DIMITRAKOPOULOS, R. Managing Grade Risk in Stope Design Optimisation: probabilistic mathematical programming model and application in sublevel stoping. **Mining Technology**. v.116. p.49-57, 2007.

GRIMSTAD, E.; BARTON, N. Updating of the Q-System for NMT. Em: International Symposium on Sprayed Concrete, Fagernes, **Proceedings [...]** pp. 46-66, 1993.

GRIMSTAD, E.; KANKES, K.; BHASIN, R.; MAGNUSSEN, A.; KAYNIA, A. Rock Mass Quality Q Used in Designing Reinforced Ribs of Sprayed Concrete and Energy Absorption. Em: International Symposium on Sprayed Concrete, Davos, **Proceedings [...]** pp. 134-142, 2002.

HABIABDOLMAJID, V., KAISER, P., MARTIN, C. Modelling brittle failure of rock. **International Journal of Rock Mechanics and Mining Sciences** v. 39, p. 731-741. set. 2002. Disponível em: <[http://dx.doi.org/10.1016/S1365-1609\(02\)00051-5](http://dx.doi.org/10.1016/S1365-1609(02)00051-5)>. Acesso em: 08 abr. 2020.

HAMRIN, H. Underground mining methods and applications. Em: Hustrulid, W. A.; Bullock, R. L. **Underground Mining Methods: Engineering Fundamentals and International Case Studies**. Colorado: Littleton, 2001.

HARTMAN, H. L. **Introductory mining engineering**. Nova York: John Wiley & Sons. 1987.

HEDLEY, D. G. F.; GRANT, F. Stope-and-pillar design for the Elliot Lake Uranium Mines. **CIM Bulletin**, Canadá, vol. 65, 723, p. 1-8, 1972.

HEIDARZADEH, S.; SAEIDI, A.; ROULEAU, A. Evaluation of the effect of geometrical parameters on stope probability of failure in the open stoping method using numerical modeling. **International Journal of Mining Science and Technology**, v. 29, n. 3, p. 399-408, maio 2019. Elsevier BV. <http://dx.doi.org/10.1016/j.ijmst.2018.05.011>.

HENNING, J. G.; MITRI, H. S. Numerical modelling of ore dilution in blasthole stoping. **International Journal of Rock Mechanics and Mining Sciences**, v. 44, n. 5, p.692-703, jul. 2007. Disponível em: <<http://dx.doi.org/10.1016/j.ijrmms.2006.11.002>>. Acesso em: 20 out. 2020.

HOEK, E. Strength of jointed rock masses. **Géotechnique**, [S.L.], v. 33, n. 3, p. 187-223, set. 1983. Thomas Telford Ltd.. <http://dx.doi.org/10.1680/geot.1983.33.3.187>.

HOEK, E. **Practical Rock Engineering**, RocScience. 2007. Disponível em: <<https://www.rocscience.com/learning/hoeks-corner/course-notes-books>> Acesso em: 10 out. 2020.

HOEK, E.; BROWN, E. T. Practical estimation of rock mass strength. **International Journal of Rock Mechanics and Mining Sciences**, vol 34, p. 1165-1186. Set. 1997.

HOEK, E.; BROWN, E. T. The Hoek–Brown failure criterion and GSI – 2018 edition. **Journal of Rock Mechanics and Geotechnical Engineering**, v. 11, n. 3, p. 445-463, jun. 2019. Elsevier BV. <http://dx.doi.org/10.1016/j.jrmge.2018.08.001>.

HUDYMA, M. R. **Development of Empirical Rib Pillar Design Criterion for Open Stope Mining**. 1988. 202 f. Dissertação (Mestrado) - Master Of Applied Science, Department Of Mining And Mineral Process Engineering, The University Of British Columbia, Vancouver, 1986. Disponível em: <https://open.library.ubc.ca/cIRcle/collections/ubctheses/831/items/1.0081054#downloadfiles>. Acesso em: 10 out. 2020.

HUTCHINSON, D. J.; DIEDERICHS, M. S. **Cablebolting in underground mines**. Vancouver: Bitech Publishers Ltd, 1996. 416 p.

ITASCA CONSULTING GROUP INC. **User's Manual FLAC Version 5.0**. Minneapolis: Itasca, 2005.

JALALI, S.; ATAEE-POUR, M. A 2D dynamic programming algorithm to optimize stope boundaries. In: Symposium on Mine Planning and Equipment Selection, 13<sup>th</sup>, **Proceedings [...]**, 2004, pp. 45-52.

JALALI, S. E.; ATAEE-POUR, M.; SHAHRIYAR, K.; ELAHI-ZEYNI, E. A Computer Program to Optimize Stope Boundaries Using Probable Stope Algorithm, **Iranian Journal of Mining Engineering (IRJME)**, Vol. 2, No. 3, pp. 7–14, 2007.

JALALI, S. E.; ATAEE-POUR, M.; SHAHRIAR, K.; ELAHI-ZEYNI, E.; NIKBIN, V.; Computer based optimisation of underground mining area, **Journal of Civil & Environmental Engineering and Science Technology**, vol. 48, no. 4, pp. 475–489, 2016.

JING, L. A review of techniques, advances and outstanding issues in numerical modelling for rock mechanics and rock engineering. **International Journal of Rock Mechanics and Mining Sciences**, v. 40, n. 3, p.283-353, abr. 2003. Disponível em: <[http://dx.doi.org/10.1016/s1365-1609\(03\)00013-3](http://dx.doi.org/10.1016/s1365-1609(03)00013-3)>. Acesso em: 08 abr. 2020.

JING, L.; HUDSON, J.A. Numerical methods in rock mechanics. **International Journal of Rock Mechanics and Mining Sciences**, v. 39, n. 4, p. 409-427, jun. 2002. Elsevier BV. [http://dx.doi.org/10.1016/s1365-1609\(02\)00065-5](http://dx.doi.org/10.1016/s1365-1609(02)00065-5).

JOHNSON, T.; SHARP, R. **Three Dimensional Dynamic Programming Method for Optimal Ultimate pit**: Report Investigation (7553). EUA: US Bureau of Mines, 1971, p. 25.

LITTLE, J.; KNIGHTS, P.; TOPAL, E. Integrated optimization of underground mine design and scheduling. **Journal of the Southern African Institute of Mining and Metallurgy**. África do Sul, p. 775-785. 2013.

LUNDER, P. J. **Hard Rock Pillar Strength Estimation an Applied Empirical Approach**. 1994. Dissertação (Mestrado). University of British Columbia. Vancouver. 1994. Disponível em: <<https://open.library.ubc.ca/collections/ubctheses/831/items/1.0087520>>. Acesso em: 20 out. 2020

LUNDER, P.J.; PAKALNIS, R. C.. Determination of the strength of hard rock mine pillar. **CIM Bulletin**, Canadá, vol 90, nº 1014, pp. 51-55, 1997.

MANCHUK, J. **A Framework for Stope Sequence Optimization**. Alberta: Centre for Computational Geostatistics (CCG), 2006.

MANCHUK, J.; DEUTSCH, C. Optimizing stope designs and sequences in underground mines. **SME Transactions**, EUA, 324, pp. 67-75, 2008.

MARTIN, C. D.; KAISER, P. K.; TANNANT, D. D.; YAZICI, S. Stress path and instability around mine openings. Em: ISRM CONGRESS ON ROCK MECHANICS, 9th, Paris. **Proceedings [...]** Rotterdam: Balkema, 1999. p. 311 - 315. Disponível em: <<http://masters.dgtu.donetsk.ua/2011/igg/stelmakh/library/article9.pdf>>. Acesso em: 20 out. 2020.

MATHEWS, K. E.; HOEK, E.; WYLIE, D. C.; STEWART, S. B. V. Prediction of stable excavation spans for mining at depths below 1,000 m in hard rock mines. **Canmet Report DSS**, Serial No. OSQ80-00081, 1981.

MICROMINE (Austrália). **MICROMINE celebrates 30 years of Intuitive Mining Solutions**. 2016. Disponível em: <https://www.micromine.com/micromine-celebrates-30-years-intuitive-mining-solutions/>. Acesso em: 20 out. 2020.

MICROMINE (Austrália). **Offline Help**. Perth: Micromine, 2021.

MELANIE, M. **An Introduction to Genetic Algorithms**. Londres: Massachusetts Institute of Technology, 1996. 158 p.

MYERS, A. **Lifting the lid on Vulcan**. 2014. Disponível em: <https://web.archive.org/web/20140512223117/http://www.maptek.com/blogs/lifting-the-lid-on-vulcan/>. Acesso em: 21 out. 2020.

NAPA-GARCÍA, G. F.; CÂMARA, T. R.; TORRES, V. F. N. Optimization of room-and-pillar dimensions using automated numerical models. **International Journal Of Mining Science And Technology**, v. 29, n. 5, p. 797-801, set. 2019. Elsevier BV. <http://dx.doi.org/10.1016/j.ijmst.2019.02.003>.

NHLEKO, A. S.; THOLANA, T.; NEINGO, P. N. A review of underground stope boundary algorithms. **Resources Policy**. Volume 56, p. 59-69, junho 2018, <https://doi.org/10.1016/j.resourpol.2017.12.004>.

NICKSON, S. D. **Cable support guidelines for underground hard rock mine operations**. 1992. 223 f. Dissertação (Mestrado) - Department Mining And Mineral Processing, University Of British Columbia, Vancouver. Disponível em: <<http://hdl.handle.net/2429/1924>>. Acesso em: 20 out. 2020.



OVANIC, J., YOUNG, D. Economic optimisation of stope geometry using separable programming with special branch and bound techniques. Em: Third Canadian Conference on Computer Applications in the Mineral Industry. 1995, Quebec. **Proceedings [...]** pp. 129-135.

OVANIC, J., YOUNG, D. Economic optimisation of open stope geometry. Em: 28th International APCOM Symposium. 1999, Colorado, **Proceedings [...]**, Colorado school of Mines, pp. 855-862.

PAN, X. D.; REED, M. B. A coupled distinct element—finite element method for large deformation analysis of rock masses. **International Journal Of Rock Mechanics And Mining Sciences & Geomechanics Abstracts**, v. 28, n. 1, p.93-99, jan. 1991. Disponível em: <[http://dx.doi.org/10.1016/0148-9062\(91\)93238-2](http://dx.doi.org/10.1016/0148-9062(91)93238-2)>. Acesso em: 08 abr. 2020.

PEARL, J. **Heuristics: intelligent search strategies for computer problem solving**. Canada: Addison-Wesley Publishing Company, 1984. 382 p. (Artificial intelligence series).

PLOTLY: The interactive graphing library for Python. The interactive graphing library for Python. 2022. Disponível em: <https://github.com/plotly/plotly.py>. Acesso em: 20 fev. 2022.

PORUMBEL, D. C. **Heuristic Algorithms And Learning Techniques: Applications to the Graph Coloring Problem**. 2009. 127 f. Tese (Doutorado) - Curso de Doutorado, Universidade de Angers, França, 2009.

POTVIN, Y., **Empirical open stope design in Canada**. 1988. 350 f. Tese (Doutorado) - Department Mining And Mineral Processing, University Of British Columbia, Vancouver. Disponível em: <<http://hdl.handle.net/2429/29264>>. Acesso em: 08 abr. 2020.

POTVIN, Y.; HUDYMA, M. Open stope mining in Canada. Em: MassMin2000, 2000, Brisbane, **Proceedings [...]**, Carlton, Victoria: Australasian Institute of Mining and Metallurgy, 2000. pp. 661-674.

PYSIMPLEGUI: Python GUIs for Humans. Python GUIs for Humans. 2022. Disponível em: <https://github.com/PySimpleGUI/PySimpleGUI>. Acesso em: 20 fev. 2022.

QUADROS, T. F. P. **Integração de Dados em ambiente SIG para Mapeamento de Favorabilidade Mineral de Ouro na Ilha Cristalina de Rivera (Uruguai)**. 2000. 285 f. Tese (Doutorado) - Curso de Engenharia de Minas, Programa de Pós-graduação em Engenharia de Minas, Metalúrgica e de Materiais, UFRGS, Porto Alegre. Disponível em: <<http://hdl.handle.net/10183/3593>>. Acesso em: 08 abr. 2020.

RIDDLE, J. M. A dynamic programming solution of a block-caving mine layout. Em: APCOM, 14th. 1976, Nova York. **Proceedings [...]** Colorado: Society for Mining, Metallurgy and Exploration, 1977. pp. 767–780.

SANDANAYAKE, D. S. S. **Stope boundary optimisation in underground mining based on a heuristic approach**. 2014. 94 f. Tese (Doutorado) - Curso de Doctor Of Philosophy, Western Australian School Of Mines, Curtin University, Perth, 2014. Disponível em: <http://hdl.handle.net/20.500.11937/1725>. Acesso em: 20 out. 2020.

SANDANAYAKE, D. S. S.; TOPAL, E.; ASAD, M. W. A. A heuristic approach to optimal design of an underground mine stope layout. **Applied Soft Computing**, v. 30, p. 595-603, maio 2015. Elsevier BV. <http://dx.doi.org/10.1016/j.asoc.2015.01.060>.



SARI, Y. A.; KUMRAL, M. A planning approach for polymetallic mines using a sublevel stoping technique with pillars and ultimate stope limits, **Engineering Optimization**, 52:6, 932-944, 2020, DOI: 10.1080/0305215X.2019.1624739

SHAHRIAR, A.P.K., ORAEE, K., BAKHTAVAR, P. S. E., 2007. A Study on the optimization Algorithms for Determining Open-Pit and Underground Mining Limits. Em: Presented at the VII-th International Scientific Conference, 2007, **Proceedings [...]**, SGEM 2007, SGEM.

SHEOREY, P. R. A theory for in-situ stresses in isotropic and transversely isotropic rock. **Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.** p. 23-34. fev. 1994. Disponível em: <[http://dx.doi.org/10.1016/0148-9062\(94\)92312-4](http://dx.doi.org/10.1016/0148-9062(94)92312-4)>. Acesso em: 08 abr. 2016.

SOTOUDEH, F.; KAKAIE, R; ATAEL, M. Development of a computer program for underground mine stope optimisation using a heuristic algorithm. Em: International Conference On Underground Mining Technology, 1st, Sudbury, p. 689-700, 2017. **Proceedings [...]** Perth: Australian Centre for Geomechanics. [http://dx.doi.org/10.36487/acg\\_rep/1710\\_58\\_soutoudeh](http://dx.doi.org/10.36487/acg_rep/1710_58_soutoudeh).

SOUZA, F. B. B. **Dimensionamento ótimo de realces abertos/câmaras inclinadas e pilares via programação matemática não-linear**. 2011, Dissertação (Mestrado), UFOP-NUGEO, Ouro Preto/MG, dezembro de 2011.

TERZAGHI, R.; VOIGHT, B. 1979. Karl Terzaghi on rockslides: the perspective of a half-century. Em: **Rockslides and Avalanches** (ed. B. Voight), Part 2, 111-131. New York: Elsevier.

TKINTER: Python interface to Tcl/Tk. Python interface to Tcl/Tk. 2022. Disponível em: <https://docs.python.org/3/library/tkinter.html>. Acesso em: 20 fev. 2022.

TOPAL, E. Early start and late start algorithms to improve the solution time for long term underground mine scheduling, **South African Institute of Mining and Metallurgy Journal SAIMM**, 108(2), 99-1079, 2008.

TOPAL, E., SENS, J. A new algorithm for stope boundary optimization. **Journal of Coal Science and Engineering** (China), p 113-119. 2010

VELLOSO, D. A.; SANTA MARIA, P. E. L. L.; LOPES, F. R. **Princípios e modelos básicos de análise**. 2º Ed. São Paulo: Pini, 1998.

VILLAESCUSA, E. Global Extraction Sequences In Sublevel Stopping. Em: MPES 2003 CONFERENCE, 2003, Kalgoorlie. **Proceedings [...]** Kalgoorlie: Western Australian School Of Mines, 2003. Disponível em: <https://api.semanticscholar.org/CorpusID:368818>. Acesso em: 10 out. 2020.

VILLAESCUSA, E. **Geotechnical Design for Sublevel Open Stopping**. 1º Ed. Boca Raton: CRC Press, 2014. 541 p.

VON KIMMELMANN, M., HYDE, B., MADGWICK, R. The use of computer applications at BCL Limited in planning pillar extraction and the design of mining layouts. Em: ISRM Symposium: Design and Performance of Underground Excavation, **Proceedings [...]**, Brown E, Hudson JA, editors, London: British Geotechnical Society; 1984. p. 53-63.

WHITTLE, G. Enterprise Optimisation, In: International Symposium on Mine Planning and Equipment Selection, 19th, **Proceedings [...]** Melbourne: The Australasian Institute of Mining and Metallurgy, 2010, pp 105-117.

## APÊNDICE A – CÓDIGOS ESCRITOS EM LINGUAGEM PYTHON

Arquivo *functions.py*

```
# -*- coding: utf-8 -*-
"""
@author: diogocordova
"""

import numpy as np
import pandas as pd
from itasca import FLAC3D_Connection
import os, sys

class HiddenPrints:
    def __enter__(self):
        self._original_stdout = sys.stdout
        sys.stdout = open(os.devnull, 'w')

    def __exit__(self, exc_type, exc_val, exc_tb):
        sys.stdout.close()
        sys.stdout = self._original_stdout

def importascii(slice_name):
    slice_file = open(slice_name, "r")
    linha1 = slice_file.readline()
    size = linha1.split()
    num_vertices = int(size[0])
    num_triangulos = int(size[1])
    # Loop over lines and extract variables of interest
    vertices = []
    for i in np.arange(num_vertices):
        linha = slice_file.readline().split()
        tmp = []
        for element in linha:
            tmp.append(float(element))
        vertices.append(tmp)

    triangulos = []
    for i in np.arange(num_triangulos):
        linha = slice_file.readline().split()
        tmp = []
        for element in linha:
            tmp.append(int(element))
        triangulos.append(tmp)
    slice_file.close()
    return np.array(vertices), np.array(triangulos)

def triang_coord(vertices, triangulos):
    num_triangulos = triangulos.shape[0]
    coord = []
    for i in np.arange(num_triangulos):
        tmp = vertices[triangulos[i, :], :]
        #arredondar para 4 casas
        #tmp = np.round(tmp,4)
        coord.append(tmp)
    return coord

def normal_vector(triangulos):
    normal_vector = []
    for tri in triangulos:
        c1 = tri[0]
        c2 = tri[1]
        c3 = tri[2]
        PQ = c2 - c1
        PR = c3 - c1
        coord_norm = np.array([
            PQ[1]*PR[2]-PR[1]*PQ[2],
            PQ[2]*PR[0]-PR[2]*PQ[0],
            PQ[0]*PR[1]-PR[0]*PQ[1]])
        coord_norm = coord_norm/np.sqrt(np.sum(coord_norm**2))
        #arredondar para 4 casas
        tmp = np.round(coord_norm,6)
        normal_vector.append(tmp)
    return np.array(normal_vector)

def make_stl_entry(triangulo, normal, coord_min):
    s = "\n facet normal"
    for num in normal:
        s += " " + str(num)
```

Continuação do arquivo *functions.py*

```

        #zerar vetor normal
        #s += " " + str(0.0)
    s += "\n outer loop"
    for vec in triangulo:
        #transladar coordenadas
        vec = vec-coord_min
        #converter para precision 6
        vec = np.round(vec,6)
        s += "\n vertex"
        for num in vec:
            s += " " + str(num)
    s += "\n endloop\n endfacet"
    return s

def ascii_2_stl(files,output,coord_min):
    stl_txt = []
    coord = []
    for f in files:
        vertices, triangulos = importascii(f)
        tri = triang_coord(vertices, triangulos)
        coord.extend(tri)

    coord = retirar_duplicados(coord)
    normal = normal_vector(coord)

    for i in range(len(coord)):
        s = make_stl_entry(coord[i],normal[i],coord_min)
        stl_txt.append(s)

    # gravando arquivo
    stl = open(output + ".stl", "w")
    stl.write("solid " + output)
    for tri in stl_txt:
        stl.write(tri)
    stl.write("\nendsolid " + output)
    stl.close()

def retirar_duplicados(coord):
    coord_ord = []
    for tri in coord:
        tri = tri[tri[:,2].argsort()] # First sort doesn't need to be stable.
        tri = tri[tri[:,1].argsort(kind='mergesort')]
        tri = tri[tri[:,0].argsort(kind='mergesort')]
        coord_ord.append(np.expand_dims(tri,axis=0))
    coord = [np.expand_dims(mat, axis = 0) for mat in coord]

    #retirando duplicados
    coord = np.concatenate(coord,axis=0)
    coord_ord = np.concatenate(coord_ord,axis=0)
    coord_ord, coord_index, coord_counts = np.unique(coord_ord, axis=0,
                                                    return_index =True,
                                                    return_counts=True)

    coord_index = coord_index[coord_counts==1]
    coord = coord[coord_index,:,:]
    coord = np.split(coord, coord.shape[0])
    coord = [np.squeeze(mat) for mat in coord]
    return coord

def flac(filename): #octree, initial, excavate galleries
    from config import gui
    flac3d = FLAC3D_Connection()
    flac3d.executable_name = gui['-flac3d_exe_file-']

    with HiddenPrints():
        flac3d.start(filename)
        flac3d.connect()
        data = 'inicio'

    # bounding box
    x_elem = int(gui['x_elem'])
    y_elem = int(gui['y_elem'])
    z_elem = int(gui['z_elem'])
    x_min = float(gui['x_min'])
    y_min = float(gui['y_min'])
    z_min = float(gui['z_min'])
    x_max = float(gui['x_max'])

```

Continuação do arquivo *functions.py*

```

y_max = float(gui['y_max'])
z_max = float(gui['z_max'])

# octree
gap_limit = float(gui['gap_limit'])
min_segments = float(gui['min_segments'])
segments = int(gui['segments'])

# FLAC3D parameters
det = int(gui['det'])
mech_ratio = float(gui['ratio'])

# stress in situ
stress_ratio = float(gui['stress_ratio'])
overburden = float(gui['overburden'])*(-10**6)

# Propriedades Mohr-Coloumb
# orezone
ore_bulk = float(gui['ore_bulk'])*10**9
ore_shear = float(gui['ore_shear'])*10**9
ore_cohesion = float(gui['ore_cohesion'])*10**6
ore_density = float(gui['ore_density'])*10**3
ore_phi = float(gui['ore_phi'])

# hangingwall
hw_bulk = float(gui['hw_bulk'])*10**9
hw_shear = float(gui['hw_shear'])*10**9
hw_cohesion = float(gui['hw_cohesion'])*10**6
hw_density = float(gui['hw_density'])*10**3
hw_phi = float(gui['hw_phi'])

# footwall
fw_bulk = float(gui['fw_bulk'])*10**9
fw_shear = float(gui['fw_shear'])*10**9
fw_cohesion = float(gui['fw_cohesion'])*10**6
fw_density = float(gui['fw_density'])*10**3
fw_phi = float(gui['fw_phi'])

# checkbox subdominios
checkbox_dom1 = str(gui['-checkbox_dom1-'])
checkbox_dom2 = str(gui['-checkbox_dom2-'])
checkbox_dom3 = str(gui['-checkbox_dom3-'])

if checkbox_dom1 == 'False':
    checkbox_dom1 = 0
    dom1_file = str(gui['-dom1_file-'])
    dom1_bulk = 0
    dom1_shear = 0
    dom1_cohesion = 0
    dom1_density = 0
    dom1_phi = 0
else:
    checkbox_dom1 = 1
    dom1_file = str(gui['-dom1_file-'])
    dom1_bulk = float(gui['dom1_bulk'])*10**9
    dom1_shear = float(gui['dom1_shear'])*10**9
    dom1_cohesion = float(gui['dom1_cohesion'])*10**6
    dom1_density = float(gui['dom1_density'])*10**3
    dom1_phi = float(gui['dom1_phi'])

if checkbox_dom2 == 'False':
    checkbox_dom2 = 0
    dom2_file = str(gui['-dom2_file-'])
    dom2_bulk = 0
    dom2_shear = 0
    dom2_cohesion = 0
    dom2_density = 0
    dom2_phi = 0
else:
    checkbox_dom2 = 1
    dom2_file = str(gui['-dom2_file-'])
    dom2_bulk = float(gui['dom2_bulk'])*10**9
    dom2_shear = float(gui['dom2_shear'])*10**9
    dom2_cohesion = float(gui['dom2_cohesion'])*10**6
    dom2_density = float(gui['dom2_density'])*10**3
    dom2_phi = float(gui['dom2_phi'])

```

Continuação do arquivo *functions.py*

```

if checkbox_dom3 == 'False':
    checkbox_dom3 = 0
    dom3_file = str(gui['-dom3_file-'])
    dom3_bulk = 0
    dom3_shear = 0
    dom3_cohesion = 0
    dom3_density = 0
    dom3_phi = 0
else:
    checkbox_dom3 = 1
    dom3_file = str(gui['-dom3_file-'])
    dom3_bulk = float(gui['dom3_bulk'])*10**9
    dom3_shear = float(gui['dom3_shear'])*10**9
    dom3_cohesion = float(gui['dom3_cohesion'])*10**6
    dom3_density = float(gui['dom3_density'])*10**3
    dom3_phi = float(gui['dom3_phi'])

# wireframes
orezone = str(gui['-orezone_filename-'])
hangingwall = str(gui['-hw_filename-'])
footwall = str(gui['-fw_filename-'])
galerias = str(gui['-galerias_filename-'])

# failure criterion
s3_menor = float(gui['s3_menor'])*10**6
failure_type = 0
#failure_type = str(gui['-LIST-'])
# trocar para bsr,ssr,outro
#ssr = 1.01 #pilares ssr <1.01
#bsr = 0.7 #pilares bsr >0.7
#s3 = 1e5 #s3 hw < 1e5 Pa

# Bounding Box
flac3d.send(x_elem)
flac3d.send(y_elem)
flac3d.send(z_elem)
flac3d.send(x_min)
flac3d.send(y_min)
flac3d.send(z_min)
flac3d.send(x_max)
flac3d.send(y_max)
flac3d.send(z_max)

# Octree
flac3d.send(gap_limit)
flac3d.send(min_segments)
flac3d.send(segments)

# FLAC3D parameters
flac3d.send(det)
flac3d.send(mech_ratio)

# stress in situ
flac3d.send(stress_ratio)
flac3d.send(overburden)

# Propriedades
# Orezone
flac3d.send(ore_bulk)
flac3d.send(ore_shear)
flac3d.send(ore_cohesion)
flac3d.send(ore_density)
flac3d.send(ore_phi)

# Hangingwall
flac3d.send(hw_bulk)
flac3d.send(hw_shear)
flac3d.send(hw_cohesion)
flac3d.send(hw_density)
flac3d.send(hw_phi)

# Footwall
flac3d.send(fw_bulk)
flac3d.send(fw_shear)
flac3d.send(fw_cohesion)

```

Continuação do arquivo *functions.py*

```

flac3d.send(fw_density)
flac3d.send(fw_phi)

# dom1
flac3d.send(dom1_bulk)
flac3d.send(dom1_shear)
flac3d.send(dom1_cohesion)
flac3d.send(dom1_density)
flac3d.send(dom1_phi)

# dom2
flac3d.send(dom2_bulk)
flac3d.send(dom2_shear)
flac3d.send(dom2_cohesion)
flac3d.send(dom2_density)
flac3d.send(dom2_phi)

# dom3
flac3d.send(dom3_bulk)
flac3d.send(dom3_shear)
flac3d.send(dom3_cohesion)
flac3d.send(dom3_density)
flac3d.send(dom3_phi)

# wireframes
#flac3d.send(orezone)
#flac3d.send(hangingwall)
#flac3d.send(footwall)
#flac3d.send(galerias)

# checkbox domains
flac3d.send(checkbox_dom1)
flac3d.send(checkbox_dom2)
flac3d.send(checkbox_dom3)

# failure criterion
flac3d.send(s3_menor)
flac3d.send(failure_type)

#51 until here

#strings
flac3d.send(gui['-orezone_filename-'])
flac3d.send(gui['-fw_filename-'])

#flac3d.send(gui['-hw_filename-'])
#flac3d.send(gui['-galerias_filename-'])
#flac3d.send(gui['-dom1_file-'])
#flac3d.send(gui['-dom2_file-'])
#flac3d.send(gui['-dom3_file-'])

#58 until here

with HiddenPrints():
    while data !=-666:
        data = flac3d.receive() #aguardar retorno do FISH
        print(f"got {data} from FLAC3D")
        flac3d.end()
        flac3d.shutdown()

# Criar arquivo "orezone.stl"
def orezone():
    # Juntar Slices
    slice_list = open(".\\SLICES\\orezone.txt", "r")
    slices = slice_list.read().split("\n")
    slice_list.close()
    slices = ['.'\\SLICES\\' + sl + ".txt" for sl in slices]

    # Translada os solidos devido precisao do STL
    coord_min = np.array([528400,6506100,0])

    # Criar "orezone.stl"
    ascii_2_stl(slices,"orezone",coord_min)

```

Arquivo *design.py*

```

# -*- coding: utf-8 -*-
"""
@author: diogocordova
"""

import numpy as np
import pandas as pd
import copy
import PySimpleGUI as sg
import shutil

def flatten_list(outer_list):
    new_list = []
    for element in outer_list:
        if isinstance(element, list):
            new_list.extend(element)
        else:
            new_list.append(element)
    return new_list

def make_design(levels=(), min_pillar=0, max_pillar=0, min_stope=0,
               max_stope=0):
    class Level:
        def __init__(self, length):
            self.length = length
            self.segments = []
            self.segment_sizes = []

            next_segment = np.random.choice(["pillar", "stope"])
            total = 0
            while total < length:
                rest = length - total
                if next_segment == "stope":
                    if rest < min_stope:
                        # bota o resto no último pilar
                        self.segment_sizes[-1] += rest
                        total += rest
                    else:
                        n = np.random.randint(min_stope,
                                              np.minimum(rest, max_stope) + 1)
                        self.segments.append(next_segment)
                        self.segment_sizes.append(n)
                        total += n
                        next_segment = "pillar"
                else:
                    if total == 0:
                        # primeiro pilar pode ter tamanho < min_pillar
                        n = np.random.randint(1,
                                              np.minimum(rest, max_pillar) + 1)
                    else:
                        # último pilar pode ter tamanho < min_pillar
                        n = np.random.randint(np.minimum(rest, min_pillar),
                                              np.minimum(rest, max_pillar) + 1)
                    self.segments.append(next_segment)
                    self.segment_sizes.append(n)
                    total += n
                    next_segment = "stope"

        def encode_as_binary(self):
            num_code = [np.ones([s]) if val == "stope" else np.zeros([s])
                       for val, s in zip(self.segments, self.segment_sizes)]
            return np.concatenate(num_code)

    class Design:
        level_lengths = levels

        def __init__(self):
            self.levels = [Level(length) for length in Design.level_lengths]
            self.fitness = None

        def __repr__(self):
            s = ""
            for level in self.levels:
                for val, size in zip(level.segments, level.segment_sizes):
                    s += val[0] + ":" + str(size) + " "
            s += "\n"

```

Continuação do arquivo *design.py*

```

return s

def encode_as_binary(self):
    return np.concatenate(
        [level.encode_as_binary() for level in self.levels])

def mutation(self):
    methods = [self.dig_mutation,
               self.fill_mutation,
               self.merge_mutation,
               self.split_mutation,
               ]
    chosen_method = np.random.choice(methods)
    chosen_method()

def crossover(self, other):
    return self.level_crossover(other)

def dig_mutation(self):
    level = self.levels[np.random.randint(len(self.levels))]
    eligiblepillars = [i for i, (val, s) in enumerate(zip(
        level.segments, level.segment_sizes))
                       if ((val == "pillar") and (s > min_pillar))]
    if len(eligiblepillars) > 0:
        pillar_pos = np.random.choice(eligiblepillars)
        pillar_size = level.segment_sizes[pillar_pos]
        dig_size = np.random.randint(1, pillar_size - min_pillar + 1)
        side = np.random.choice(["left", "right"])
        if side == "left":
            if pillar_pos == 0:
                # new stope to the left
                if min_stope <= dig_size <= max_stope:
                    level.segment_sizes[pillar_pos] -= dig_size
                    level.segment_sizes.insert(pillar_pos, dig_size)
                    level.segments.insert(pillar_pos, "stope")
                else:
                    dig_size = np.minimum(
                        dig_size,
                        np.maximum(
                            max_stope - level.segment_sizes[pillar_pos -
                                                                1], 0))
                    if dig_size > 0:
                        level.segment_sizes[pillar_pos] -= dig_size
                        level.segment_sizes[pillar_pos - 1] += dig_size
            else:
                if pillar_pos == len(level.segments) - 1:
                    # new stope to the right
                    if min_stope <= dig_size <= max_stope:
                        level.segment_sizes[pillar_pos] -= dig_size
                        level.segment_sizes.append(dig_size)
                        level.segments.append("stope")
                    else:
                        dig_size = np.minimum(
                            dig_size,
                            np.maximum(
                                max_stope - level.segment_sizes[pillar_pos +
                                                                    1], 0))
                        if dig_size > 0:
                            level.segment_sizes[pillar_pos] -= dig_size
                            level.segment_sizes[pillar_pos + 1] += dig_size
        else:
            if pillar_pos == len(level.segments) - 1:
                # new stope to the right
                if min_stope <= dig_size <= max_stope:
                    level.segment_sizes[pillar_pos] -= dig_size
                    level.segment_sizes.append(dig_size)
                    level.segments.append("stope")
                else:
                    dig_size = np.minimum(
                        dig_size,
                        np.maximum(
                            max_stope - level.segment_sizes[pillar_pos +
                                                                1], 0))
                    if dig_size > 0:
                        level.segment_sizes[pillar_pos] -= dig_size
                        level.segment_sizes[pillar_pos + 1] += dig_size

def fill_mutation(self):
    level = self.levels[np.random.randint(len(self.levels))]
    eligiblestopes = [i for i, (val, s) in enumerate(zip(
        level.segments, level.segment_sizes))
                      if ((val == "stope") and (s > min_stope))]
    if len(eligiblestopes) > 0:
        stope_pos = np.random.choice(eligiblestopes)
        stope_size = level.segment_sizes[stope_pos]
        fill_size = np.random.randint(1, stope_size - min_stope + 1)
        side = np.random.choice(["left", "right"])
        if side == "left":
            if stope_pos == 0:
                # new pillar to the left
                if min_pillar <= fill_size <= max_pillar:
                    level.segment_sizes[stope_pos] -= fill_size

```



### Continuação do arquivo *design.py*

```

        level.segment_sizes.insert(stope_pos, fill_size)
        level.segments.insert(stope_pos, "pillar")
    else:
        fill_size = np.minimum(
            fill_size, np.maximum(
                max_stope - level.segment_sizes[stope_pos - 1],
                0))
        if fill_size > 0:
            level.segment_sizes[stope_pos] -= fill_size
            level.segment_sizes[stope_pos - 1] += fill_size
    else:
        if stope_pos == len(level.segments) - 1:
            # new pillar to the right
            if min_pillar <= fill_size <= max_pillar:
                level.segment_sizes[stope_pos] -= fill_size
                level.segment_sizes.append(fill_size)
                level.segments.append("pillar")
        else:
            fill_size = np.minimum(
                fill_size,
                np.maximum(max_stope - level.segment_sizes[stope_pos
                    + 1], 0))
            if fill_size > 0:
                level.segment_sizes[stope_pos] -= fill_size
                level.segment_sizes[stope_pos + 1] += fill_size

def merge_mutation(self):
    level = self.levels[np.random.randint(len(self.levels))]
    if len(level.segments) > 1:
        position = np.random.choice(len(level.segments))
        if position == 0:
            new_size = level.segment_sizes[1] + level.segment_sizes[0]
            if level.segments[position] == "stope" and \
                new_size <= max_pillar \
                or level.segments[position] == "pillar" and \
                new_size <= max_stope:
                level.segment_sizes[1] += level.segment_sizes[0]
                level.segment_sizes.pop(0)
                level.segments.pop(0)
        elif position == len(level.segments) - 1:
            new_size = level.segment_sizes[-2] + \
                level.segment_sizes[-1]
            if level.segments[position] == "stope" and \
                new_size <= max_pillar \
                or level.segments[position] == "pillar" and \
                new_size <= max_stope:
                level.segment_sizes[-2] += level.segment_sizes[-1]
                level.segment_sizes.pop(-1)
                level.segments.pop(-1)
        else:
            new_size = level.segment_sizes[position - 1] + \
                level.segment_sizes[position] \
                + level.segment_sizes[position + 1]
            if level.segments[position] == "stope" and \
                new_size <= max_pillar \
                or level.segments[position] == "pillar" and \
                new_size <= max_stope:
                level.segment_sizes[position - 1] += \
                    level.segment_sizes[position]
                level.segment_sizes[position - 1] += \
                    level.segment_sizes[position + 1]
                level.segment_sizes.pop(position)
                level.segment_sizes.pop(position)
                level.segments.pop(position)
                level.segments.pop(position)

def split_mutation(self):
    level = self.levels[np.random.randint(len(self.levels))]
    eligible_positions = []
    for p, (s, val) in enumerate(zip(level.segment_sizes,
                                    level.segments)):
        if (val == "pillar") & (s >= 2 * min_pillar + min_stope):
            eligible_positions.append(p)
        elif (val == "stope") & (s >= 2 * min_stope + min_pillar):
            eligible_positions.append(p)

```

Continuação do arquivo *design.py*

```

if len(eligible_positions) > 0:
    position = np.random.choice(eligible_positions)
    size = level.segment_sizes[position]
    val = level.segments[position]
    if val == "pillar":
        start_point = np.random.randint(min_pillar, size - \
                                         min_pillar - min_stope + 1)
        end_point = np.random.randint(start_point + min_stope, \
                                     size - min_pillar + 1)
        fill = ["pillar", "stope", "pillar"]
    else:
        start_point = np.random.randint(min_stope, size - \
                                         min_pillar - min_stope + 1)
        end_point = np.random.randint(start_point + min_pillar, \
                                     size - min_stope + 1)
        fill = ["stope", "pillar", "stope"]
    length_1 = start_point
    length_2 = end_point - start_point
    length_3 = size - end_point

    level.segments[position] = fill
    level.segment_sizes[position] = [length_1, length_2, length_3]

    level.segments = flatten_list(level.segments)
    level.segment_sizes = flatten_list(level.segment_sizes)

def shift_mutation(self):
    level = self.levels[np.random.randint(len(self.levels))]
    if len(level.segments) >= 4:
        segment_type = np.random.choice(["stope", "pillar"])
        if segment_type == "stope":
            eligible_positions = [i for i, (s, val) in
                                 enumerate(zip(level.segment_sizes,
                                              level.segments))
                                 if val == "stope" and s >= min_stope]
            if len(eligible_positions) > 1:
                chosen_positions = np.random.choice(eligible_positions,
                                                    2, replace=False)
                max_shift = np.minimum(
                    level.segment_sizes[chosen_positions[0]] - \
                    min_stope,
                    max_stope - level.segment_sizes[chosen_positions[0]])
                if max_shift > 0:
                    shift = np.random.randint(1, max_shift + 1)
                    level.segment_sizes[chosen_positions[0]] -= shift
                    level.segment_sizes[chosen_positions[1]] += shift
            else:
                eligible_positions = [i for i, (s, val) in \
                                     enumerate(zip(level.segment_sizes,
                                                  level.segments))
                                     if val == "pillar" and s >= min_pillar]
                if len(eligible_positions) > 1:
                    chosen_positions = np.random.choice(
                        eligible_positions, 2, replace=False)
                    max_shift = np.minimum(
                        level.segment_sizes[chosen_positions[0]] - \
                        min_pillar, max_pillar - \
                        level.segment_sizes[chosen_positions[0]])
                    if max_shift > 0:
                        shift = np.random.randint(1, max_shift + 1)
                        level.segment_sizes[chosen_positions[0]] -= shift
                        level.segment_sizes[chosen_positions[1]] += shift

def level_crossover(self, other):
    if len(self.levels) == 1:
        return self, other

    n_levels = np.random.randint(1, int(len(self.levels) / 2) + 1)
    level_positions = np.random.choice(len(self.levels), n_levels,
                                       replace=False)

    levels_1 = copy.deepcopy(self.levels)
    levels_2 = copy.deepcopy(other.levels)

    for pos in level_positions:
        levels_1[pos] = copy.deepcopy(other.levels[pos])

```

Continuação do arquivo *design.py*

```

        levels_2[pos] = copy.deepcopy(self.levels[pos])

        child_1 = Design()
        child_2 = Design()
        child_1.levels = levels_1
        child_2.levels = levels_2
        return child_1, child_2

    return Design

def roulette_selection(fitvec):
    p = fitvec - fitvec.min()
    p = p + 0.1 * p.max() + 1e-6
    p = p / p.sum()
    sel = np.random.choice(len(fitvec), 2, replace=False, p=p)
    return sel

def tournament_selection(fitvec, k=5):
    fitvec = fitvec.astype(np.float)

    sel_1 = np.random.choice(len(fitvec), k, replace=False)
    sub_fit = fitvec[sel_1]
    chosen_1 = sel_1[np.argmax(sub_fit)]

    sel_2 = np.random.choice(len(fitvec), k, replace=False)
    sub_fit = fitvec[sel_2]
    chosen_2 = sel_2[np.argmax(sub_fit)]

    if chosen_2 == chosen_1:
        sub_fit[np.argmax(sub_fit)] = - np.inf
        chosen_2 = sel_2[np.argmax(sub_fit)]

    return np.array([chosen_1, chosen_2])

class GeneticOptimizer:
    def __init__(self, design_cls, fitness_fn):
        self.design_cls = design_cls
        self.fitness_fn = fitness_fn
        self.popsiz = 20
        self.mut_prob = 1.0
        self.seed = None
        self.population = []
        self.best_solution = None
        self.best_fitness = None
        self.evolution_log = []
        self.verbose = True
        self.temperature = 0.5
        self.selection_fn = roulette_selection
        self.base_variance = None
        self.forced_mutations = 10

    def populate(self, start=None):
        """Resets population"""
        if self.seed is not None:
            np.random.seed(self.seed)

        self.population = []
        if start is not None:
            self.population.append(start)
        while len(self.population) < self.popsiz:
            self.population.append(self.design_cls())

        for i, elem in enumerate(self.population):
            if self.verbose:
                print("\nIniciando população: " + str(i + 1) + " ",
                      end="")
            if not sg.one_line_progress_meter(
                title='Iniciando população...', orientation='h',
                current_value = i + 1, max_value=self.popsiz,
                no_button=False):
                return
            elem.fitness = self.fitness_fn(elem)

        #save plots
        target = 'plots/scenario_geometries_pop_'+str(i + 1)+".pdf"
        original = r'scenario_geometries.pdf'

```

Continuação do arquivo *design.py*

```

shutil.copyfile(original, target)

target = 'plots/scenario_iso_s3_pop_'+str(i + 1)+".pdf"
original = r'scenario_iso_s3.pdf'
shutil.copyfile(original, target)

target = 'plots/scenario_iso_ssr_pop_'+str(i + 1)+".pdf"
original = r'scenario_iso_ssr.pdf'
shutil.copyfile(original, target)

if self.verbose:
    print(" ")

fitvec = np.array([elem.fitness for elem in self.population])
self.evolution_log.append(copy.deepcopy(self.population))
self.best_fitness = np.max(fitvec)
self.best_solution = copy.deepcopy(
    self.population[int(np.argmax(fitvec))])

self.base_variance = self.population_variance()

def population_variance(self):
    bin_arr = np.stack([elem.encode_as_binary() for elem in \
        self.population], axis=1).T
    return np.sum(np.var(bin_arr, axis=0))

def repopulate(self):
    if self.verbose:
        print("\n")
    for i, elem in enumerate(self.population):
        if self.verbose:
            print("\nRepopulating: " + str(i + 1) + "      ",
                end="")
        for _ in range(self.forced_mutations):
            elem.mutation()
            elem.fitness = self.fitness_fn(elem)
    if self.verbose:
        print("\n")

    fitvec = np.array([elem.fitness for elem in self.population])
    max_fitness = np.max(fitvec)
    if max_fitness > self.best_fitness:
        self.best_fitness = max_fitness
        self.best_solution = copy.deepcopy(
            self.population[int(np.argmax(fitvec))])
    else:
        self.population[0] = self.best_solution

def iterate(self, i, replace_worst=False):
    """One iteration of training"""
    # selection
    fitvec = np.array([elem.fitness for elem in self.population])
    sel = self.selection_fn(fitvec)

    # crossover/mutation
    parent_1, parent_2 = self.population[sel[0]], self.population[sel[1]]
    child1, child2 = parent_1.crossover(parent_2)

    if np.random.uniform() < self.mut_prob:
        child1.mutation()
    if np.random.uniform() < self.mut_prob:
        child2.mutation()

    # updating fitness
    # if child1.fitness is None:
    child1.fitness = self.fitness_fn(child1)

    # save plots
    target = 'plots/scenario_geometries_iteration_'+str(i + 1)+"_child1.pdf"
    original = r'scenario_geometries.pdf'
    shutil.copyfile(original, target)

    target = 'plots/scenario_iso_s3_iteration_'+str(i + 1)+"_child1.pdf"
    original = r'scenario_iso_s3.pdf'
    shutil.copyfile(original, target)

```

Continuação do arquivo *design.py*

```

target = 'plots/scenario_iso_ssr_iteration_'+str(i + 1)+"_child1.pdf"
original = r'scenario_iso_ssr.pdf'
shutil.copyfile(original, target)

# if child2.fitness is None:
child2.fitness = self.fitness_fn(child2)

# save plots
target = 'plots/scenario_geometries_iteration_'+str(i + 1)+"_child2.pdf"
original = r'scenario_geometries.pdf'
shutil.copyfile(original, target)

target = 'plots/scenario_iso_s3_iteration_'+str(i + 1)+"_child2.pdf"
original = r'scenario_iso_s3.pdf'
shutil.copyfile(original, target)

target = 'plots/scenario_iso_ssr_iteration_'+str(i + 1)+"_child2.pdf"
original = r'scenario_iso_ssr.pdf'
shutil.copyfile(original, target)

# updating population
if replace_worst:
    # replace smallest fitness - more aggressive
    pos = np.argmin(fitvec)
    if child1.fitness > fitvec[pos]:
        self.population[pos] = child1
        fitvec[pos] = child1.fitness
    pos = np.argmin(fitvec)
    if child2.fitness > fitvec[pos]:
        self.population[pos] = child2
        fitvec[pos] = child2.fitness
else:
    # replacing parents if better - preserves diversity
    if (fitvec[sel[0]] < fitvec[sel[1]]) & \
        (fitvec[sel[0]] < child1.fitness):
        fitvec[sel[0]] = child1.fitness
        self.population[sel[0]] = child1
    elif (fitvec[sel[1]] < fitvec[sel[0]]) & \
        (fitvec[sel[1]] < child1.fitness):
        fitvec[sel[1]] = child1.fitness
        self.population[sel[1]] = child1
    if (fitvec[sel[0]] < fitvec[sel[1]]) & \
        (fitvec[sel[0]] < child2.fitness):
        fitvec[sel[0]] = child2.fitness
        self.population[sel[0]] = child2
    elif (fitvec[sel[1]] < fitvec[sel[0]]) & \
        (fitvec[sel[1]] < child2.fitness):
        fitvec[sel[1]] = child2.fitness
        self.population[sel[1]] = child2

# updating log
self.evolution_log.append(copy.deepcopy(self.population))
self.best_fitness = np.max(fitvec)
self.best_solution = copy.deepcopy(
    self.population[int(np.argmax(fitvec))])

def optimize(self, max_iter=1000, popsize=50, mut_prob=1.0, tol=0.1,
             patience=None, seed=None, start=None, selection_fn=None,
             variance_thr=0.00, forced_mutations=10):
    self.popsize = popsize
    self.mut_prob = mut_prob
    self.seed = seed
    self.forced_mutations = forced_mutations

    if selection_fn is not None:
        self.selection_fn = selection_fn

    if patience is None:
        patience = np.ceil(max_iter / 5)

    self.populate(start)

    stagnation = 0
    self.temperature = 1
    best_fitness = self.best_fitness
    if self.verbose:

```

Continuação do arquivo *design.py*

```

print("\nIteração: 0 | Melhor fitness: US$ " + \
      str("{:,.1f}".format(self.best_fitness)), end="")
for i in range(max_iter):
    if not sg.one_line_progress_meter(title='Algoritmo Genético...',
                                     orientation='h',
                                     current_value = i + 1,
                                     max_value=max_iter,
                                     no_button=False):
        break
    if self.population_variance() / self.base_variance < variance_thr:
        self.repopulate()

    self.iterate(i, replace_worst=i / max_iter > 0.7)

    ev = self.best_fitness - best_fitness
    if ev < tol:
        stagnation += 1
        self.temperature = np.maximum(0.05, np.minimum(
            self.temperature + 0.001, (1 - i / max_iter) ** 2))
    else:
        stagnation = 0
        self.temperature = 0.05
    if stagnation >= patience:
        if self.verbose:
            print("\nOtimização terminada na iteração " + str(i + 1))
        sg.one_line_progress_meter_cancel()
        break
    else:
        best_fitness = self.best_fitness
        if self.verbose:
            #start
            cont_p = 0
            soma_p = 0
            cont_s = 0
            soma_s = 0
            for val, s in zip(self.segments, self.segment_sizes):
                if val == 1:
                    cont_p += 1
                    soma_p += self.segment_sizes(s)
                    mean_pillar = soma_p/cont_p
                else:
                    cont_s += 1
                    soma_s += self.segment_sizes(s)
                    mean_stope = soma_s/cont_s
            #end

            print("\nIteração: " + str(i + 1) +
                  " | Melhor fitness: US$ "
                  + str("{:,.1f}".format(self.best_fitness)) +
                  " | Melhora <"
                  + str(tol) + " para " + str(stagnation) +
                  "/"
                  + str(patience) + " iterações "
                  + "mean_pillar:" + str(mean_pillar) + "mean_stope:"
                  + str(mean_stope), sep="", end="")
        print("\nFim da otimização...")

def export_history(self):
    exp = {}
    for i, gen in enumerate(self.evolution_log):
        fitvec = np.array([elem.fitness for elem in gen])
        exp["Geração %d" % i] = fitvec
    return pd.DataFrame(exp)

```

Arquivo *profit\_flac.py*

```

# -*- coding: utf-8 -*-
"""
@author: diogocordova
"""

import numpy as np
import pandas as pd
import functions as f

def flac_process(scenario):
    # leitura da lista de slices totais, na ordem de cima para baixo da
    # esquerda para direita
    lists = open(".\SLICES\lista.txt", "r")
    list_slices = lists.read().split("\n")
    lists.close()
    list_slices_arr = np.array(list_slices)

    scenario_bin = scenario.encode_as_binary()

    ### separa em arquivo de pilares e stopes
    idx_stopes = [i for i, e in enumerate(scenario_bin) if e == 1]
    idx_pilares = [i for i, e in enumerate(scenario_bin) if e == 0]
    pilares_slices = list_slices_arr[idx_pilares]
    stope_slices = list_slices_arr[idx_stopes]

    ### converte para lista
    pilares_slices_list = pilares_slices.tolist()
    stope_slices_list = stope_slices.tolist()
    pilares_slices_list = ['.\SLICES\'+ sl +
                           ".txt" for sl in pilares_slices_list]
    stope_slices_list = ['.\SLICES\'+ sl +
                          ".txt" for sl in stope_slices_list]

    ### Translada os solidos devido precisao do STL
    coord_min = np.array([528400,6506100,0])

    ### ASCII to STL
    f.ascii_2_stl(stope_slices_list,"stopes",coord_min)
    f.ascii_2_stl(pilares_slices_list,"pilares",coord_min)

    ### Modelar no FLAC3D cenario de stopes e pilares,
    ### arquivos 'stopes.stl' e 'pilares.stl'
    f.flac('py_stopes.f3dat')

    return stope_slices

def profit(stope_slices, price,r1,r2,c_lavra, c_transp,c_process,
          c_ga,c_ref,royalty,pillar_support,hw_support,pof_max,max_dil,
          hw_support_efficiency):

    # Abrir arquivo "slices_data.xlsx"
    from config import gui
    slices_file_attributes = gui['-slices_attributes-']
    df_slices = pd.read_excel(slices_file_attributes, sheet_name=0, header=0,
                             index_col=0)

    # Abrir arquivo "results.dat"
    results_file = open("results.dat", "r")
    linhas = results_file.readlines()
    # pof_ssr = float(linhas[6])/100
    pof_bsr = float(linhas[7])/100
    dil = float(linhas[8])/100
    rec = float(linhas[9])/100
    dil_vol = float(linhas[10])
    pillar_vol_failure = float(linhas[11])

    #pof = (pof_ssr+pof_bsr)/2
    pof = pof_bsr
    results_file.close()

    # Obter teor médio, toneladas e DMT do cenário
    df_slices_scenario = df_slices.loc[stope_slices]
    sums_slices_scenarios = df_slices_scenario.sum(axis=0)
    mean_slices_scenarios = df_slices_scenario.mean(axis=0)
    teor = sums_slices_scenarios['g']/sums_slices_scenarios['ton']
    ton = sums_slices_scenarios['ton']

```

Continuação do arquivo *profit\_flac.py*

```
dist = mean_slices_scenarios['dist']

if pof>=pof_max:
    pillar_support_cost = (pillar_support*pillar_vol_failure)
else:
    pillar_support_cost = 0
if dil>=max_dil:
    hw_support_cost = hw_support*dil_vol
    actual_dil = dil-((dil-max_dil)*hw_support_efficiency)
else:
    hw_support_cost = 0
    actual_dil = dil

gross_value = price*teor*ton*r1*r2

cost_value = (((ton*r1*(1+actual_dil))*(c_lavra+c_process+c_ga
                +(c_transp*dist)))+
              (teor*ton*r1*r2*c_ref)+(royalty*gross_value)+
              pillar_support_cost+hw_support_cost)
profit = gross_value - cost_value

output = [profit, pof, rec, dil, dil_vol, pillar_vol_failure, dist,
          ton, teor]

return output
```



## Arquivo GA.py

```

# -*- coding: utf-8 -*-
"""
@author: diogocordova
"""

import os
os.chdir(r"C:\DR\DR_FLAC3D 7")
import profit_flac
import design
import numpy as np
import plotly.express as px
import plotly.offline as py
import pandas as pd
import shutil
import functions as f

def fitness(scenario):
    global price, r1, r2, c_lavra, c_transp, c_process, c_ga, c_ref, royalty
    global pillar_support, hw_support, pof_max, max_dil, hw_support_efficiency
    from config import gui
    price = float(gui['price'])
    r1 = float(gui['r1'])/100
    r2 = float(gui['r2'])/100
    c_lavra = float(gui['c_lavra'])
    c_transp = float(gui['c_transp'])
    c_process = float(gui['c_process'])
    c_ga = float(gui['c_ga'])
    c_ref = float(gui['c_ref'])
    royalty = float(gui['royalty'])/100
    pillar_support = float(gui['pillar_support'])
    hw_support = float(gui['hw_support'])
    pof_max = float(gui['pof_max'])/100
    max_dil = float(gui['max_dil'])/100
    hw_support_efficiency = float(gui['hw_support_efficiency'])/100
    global scenarios_array
    stope_slices = profit_flac.flac_process(scenario)
    output = profit_flac.profit(stope_slices,price,r1,r2,c_lavra, c_transp,
                               c_process,c_ga,c_ref,royalty,pillar_support,
                               hw_support,pof_max,max_dil,
                               hw_support_efficiency)
    output_arr = np.array(output)

    ### Salvar resultado fitness
    scenarios_array = np.column_stack((scenarios_array, output_arr))
    scenarios_df = pd.DataFrame(scenarios_array)
    scenarios_df = scenarios_df.rename(index={
        0: 'profit', 1: 'pof', 2: 'rec', 3: 'dil', 4: 'dil_vol',
        5: 'pillar_vol_failure', 6: 'DMT', 7: 'ton', 8: 'teor'})
    file = open("scenarios_fitness.txt", "w")
    file.write(scenarios_df.to_string())
    file.close()
    return output[0]

def otimizador():
    from config import gui
    global scenarios_array
    scenarios_array = np.zeros((9,0))

    # Modelo FLAC3D inicial (octree, inicial e excavar galerias),
    # "orezone.stl" e "galerias.dxf"
    print("Iniciando modelo numérico base...")
    f.flac('master.f3dat')

    # Cria o objeto design, que obedece os tamanhos dos niveis e minimos e
    # maximos de pilares e stopes
    min_pillar = int(gui['min_pillar'])
    max_pillar = int(gui['max_pillar'])
    min_stope = int(gui['min_stope'])
    max_stope = int(gui['max_stope'])
    levels = eval(gui['level_slices'])

    Design = design.make_design(levels, min_pillar=min_pillar,
                                max_pillar= max_pillar, min_stope=min_stope,
                                max_stope=max_stope)

    # Inicia o algoritmo genetico

```

Continuação do arquivo *GA.py*

```

patience = int(gui['patience'])
popsize = int(gui['popsize'])
max_iter = int(gui['max_iter'])
forced_mutations = int(gui['forced_mutations'])
tol = float(gui['tol'])

opt = design.GeneticOptimizer(Design, fitness)
opt.optimize(patience=patience, popsize=popsize, max_iter=max_iter,
             selection_fn=None,
             variance_thr=0.00, forced_mutations=forced_mutations, tol=tol)

# Guarda o historico dos resultados
history = opt.export_history()
history["id"] = np.arange(history.shape[0])
history_melt = history.melt(id_vars="id", var_name="Geração",
                           value_name="Fitness")
history_melt["Geração"] = history_melt["Geração"].apply(
    lambda x: int(x.replace("Geração ", "")))

box_plt = px.box(history_melt, x="Geração", y="Fitness")

rpt_filename = gui['-evolution_report-']
py.plot(box_plt, filename=rpt_filename,)

best_slices = profit_flac.flac_process(opt.best_solution)
best_output = profit_flac.profit(best_slices, price, r1, r2, c_lavra,
                                c_transp, c_process, c_ga, c_ref, royalty,
                                pillar_support, hw_support, pof_max, max_dil,
                                hw_support_efficiency)

target_filename = gui['-best_cenario_filename-']
original = r'stopes.stl'
shutil.copyfile(original, target_filename)

```

## Arquivo GUI.py

```

# -*- coding: utf-8 -*-
"""
@author: diogocordova
"""

import PySimpleGUI as sg
import GA
import config
global gui
gui={}

def gui():
    sg.theme('Reddit')
    sg.theme_input_background_color('white')

    arquivos_layout = [
        [sg.Frame('Wireframes', size=(600,150), layout=[
            [sg.T('Zona de otimização dos painéis (.dxf):', justification='r',
                size=(30, 1)),
            sg.I(key='-orezone_filename-', size=(55, 1), justification='l'),
            sg.FileBrowse(file_types=(("DXF Files (.dxf)", "*.dxf")),
                button_text='...')],
            [sg.T('Bounding Box do hangingwall (.dxf):', justification='r',
                size=(30, 1)),
            sg.I(key='-hw_filename-', size=(55, 1), justification='l'),
            sg.FileBrowse(file_types=(("DXF Files (.dxf)", "*.dxf")),
                button_text='...')],
            [sg.T('Bounding Box do footwall (.dxf):', justification='r',
                size=(30, 1)),
            sg.I(key='-fw_filename-', size=(55, 1), justification='l'),
            sg.FileBrowse(file_types=(("DXF Files (.dxf)", "*.dxf")),
                button_text='...')],
            [sg.T('Subníveis (.dxf):', justification='r', size=(30, 1)),
            sg.I(key='-galerias_filename-', size=(55, 1), justification='l'),
            sg.FileBrowse(file_types=(("DXF Files (.dxf)", "*.dxf")),
                button_text='...')]]],
        [sg.Frame('Fatias (slices) do corpo de minério', size=(600,90),
            layout=[
            [sg.T('Pasta contendo slices em formato STL:', justification='r',
                size=(30, 1)),sg.I(key='-slices_folder-', size=(55, 1)),
            sg.FolderBrowse(button_text='...')],
            [sg.T('Planilha com atributos dos slices (.xlsx):',
                justification='r', size=(30, 1)),
            sg.I(key='-slices_attributes-',size=(55, 1)),
            sg.FileBrowse(file_types=(("Excel Files (.xlsx)", "*.xlsx")),
                button_text='...')]]])
        ]

    geometricos_layout = [
        [sg.Frame('Pilares', size=(600,75), layout=[
            [sg.T('Largura mínima dos pilares (m)', justification='r',
                size=(40, 1)), sg.I(key='min_pillar', justification='c',
                size=(10, 1))],
            [sg.T('Largura máxima dos pilares (m)', justification='r',
                size=(40, 1)), sg.I(key='max_pillar', justification='c',
                size=(10, 1))]]],
        [sg.Frame('Realces', size=(600,75), layout=[
            [sg.T('Largura mínima dos realces (m)', justification='r',
                size=(40, 1)), sg.I(key='min_stope', justification='c',
                size=(10, 1))],
            [sg.T('Largura máxima dos realces (m)', justification='r',
                size=(40, 1)), sg.I(key='max_stope', justification='c',
                size=(10, 1))]]],
        [sg.Frame('Subníveis', size=(600,75), layout=[
            [sg.T('Quantidade', justification='r', size=(40, 1)),
            sg.I(key='qtd_subniveis', justification='c', size=(10, 1))],
            [sg.T('Número de slices por subnível (separado por vírgulas)',
                justification='r', size=(40, 1)),
            sg.I(key='leve_slices', justification='c', size=(30, 1))]]])
        ]

    economicos_layout = [
        [sg.Frame('Dados econômicos', size=(600,260), layout=[
            [sg.T('Custo de lavra (U$/ton)', justification='r', size=(30, 1)),
            sg.I(key='c_lavra', justification='c', size=(10, 1))],
            [sg.T('Custo de transporte (U$/ton/DMT)', justification='r',

```

## Continuação do arquivo GUI.py

```

        size=(30, 1)),
        sg.I(key='c_transp', justification='c', size=(10, 1)),
        [sg.T('Custo de processo (US$/ton)', justification='r',
            size=(30, 1)),
            sg.I(key='c_process', justification='c', size=(10, 1)),
            [sg.T('Custo G&A (US$/ton)', justification='r', size=(30, 1)),
                sg.I(key='c_ga', justification='c', size=(10, 1)),
                [sg.T('Custo de Refino (US$/g metal)', justification='r',
                    size=(30, 1)),
                        sg.I(key='c_ref', justification='c', size=(10, 1)),
                        [sg.T('Royalty (%)', justification='r', size=(30, 1)),
                            sg.I(key='royalty', justification='c', size=(10, 1)),
                            [sg.T('Preço do Ouro (US$/g)', justification='r', size=(30, 1)),
                                sg.I(key='price', justification='c', size=(10, 1)),
                                [sg.T('Recuperação de lavra nos realces (%)', justification='r',
                                    size=(30, 1)),
                                        sg.I(key='r1', justification='c', size=(10, 1)),
                                        [sg.T('Recuperação Planta de Processo (%)', justification='r',
                                            size=(30, 1)),
                                                sg.I(key='r2', justification='c', size=(10, 1))]]]]]]]]
    ]

##### GEOMECANICA LAYOUT

desempenho_layout = [
    [sg.Frame('Hanging wall', size=(600,120), layout=[
        [sg.T('Critério de diluição, s3 menor que (MPa):',
            justification='r', size=(40, 1)),
            sg.I(key='s3_menor', justification='c', size=(10, 1)),
            [sg.T('Diluição esperada (%):', justification='r', size=(40, 1)),
                sg.I(key='max_dil', justification='c', size=(10, 1)),
                [sg.T('Custo de suporte (US$/m3 de diluição):',
                    justification='r', size=(40, 1)),
                        sg.I(key='hw_support', justification='c', size=(10, 1)),
                        [sg.T('Eficiência do suporte do HW (%):', justification='r',
                            size=(40, 1)),
                                sg.I(key='hw_support_efficiency',
                                    justification='c', size=(10, 1))]]]]],
        [sg.Frame('Pilares', size=(600,100), layout=[
            [sg.T('Critério de falha:', justification='r', size=(40, 1)),
                sg.Combo(('BSR>0.7', 'SSR<1.01', 'FS<1'),
                    size=(20, 12), key='-LIST-', enable_events=True)],
            [sg.T('PoF máxima aceitável (%):', justification='r',
                size=(40, 1)),
                sg.I(key='pof_max', justification='c', size=(10, 1)),
                [sg.T('Custo de suporte (US$/m3 de pilar com falha):',
                    justification='r', size=(40, 1)),
                    sg.I(key='pillar_support', justification='c', size=(10, 1))]]]]],
        ]
    ]

propriedades_layout = [
    [sg.Frame('Orezone', size=(190,155), layout=[
        [sg.T('Densidade (t/m3):', justification='r', size=(15, 1)),
            sg.I(key='ore_density', justification='c', size=(5, 1)),
            [sg.T('K - Bulk (GPa):', justification='r', size=(15, 1)),
                sg.I(key='ore_bulk', justification='c', size=(5, 1)),
                [sg.T('G - Shear (GPa):', justification='r', size=(15, 1)),
                    sg.I(key='ore_shear', justification='c', size=(5, 1)),
                    [sg.T('Coesão (MPa):', justification='r', size=(15, 1)),
                        sg.I(key='ore_cohesion', justification='c', size=(5, 1)),
                        [sg.T('Ângulo de atrito (°):', justification='r', size=(15, 1)),
                            sg.I(key='ore_phi', justification='c', size=(5, 1)),
                            ]
                        ]
                    ]
                ]
            ]
        ]
    ],
    [sg.Frame('Hangingwall', size=(190,155), layout=[
        [sg.T('Densidade (t/m3):', justification='r', size=(15, 1)),
            sg.I(key='hw_density', justification='c', size=(5, 1)),
            [sg.T('K - Bulk (GPa):', justification='r', size=(15, 1)),
                sg.I(key='hw_bulk', justification='c', size=(5, 1)),
                [sg.T('G - Shear (GPa):', justification='r', size=(15, 1)),
                    sg.I(key='hw_shear', justification='c', size=(5, 1)),
                    [sg.T('Coesão (MPa):', justification='r', size=(15, 1)),
                        sg.I(key='hw_cohesion', justification='c', size=(5, 1)),
                        [sg.T('Ângulo de atrito (°):', justification='r', size=(15, 1)),
                            sg.I(key='hw_phi', justification='c', size=(5, 1)),
                            ]
                        ]
                    ]
                ]
            ]
        ]
    ],
    [sg.Frame('Footwall', size=(190,155), layout=[

```

### Continuação do arquivo GUI.py

```

[sg.T('Densidade (t/m3):', justification='r', size=(15, 1)),
 sg.I(key='fw_density', justification='c', size=(5, 1))],
[sg.T('K - Bulk (GPa):', justification='r', size=(15, 1)),
 sg.I(key='fw_bulk', justification='c', size=(5, 1))],
[sg.T('G - Shear (GPa):', justification='r', size=(15, 1)),
 sg.I(key='fw_shear', justification='c', size=(5, 1))],
[sg.T('Coesão (MPa):', justification='r', size=(15, 1)),
 sg.I(key='fw_cohesion', justification='c', size=(5, 1))],
[sg.T('Ângulo de atrito (°):', justification='r', size=(15, 1)),
 sg.I(key='fw_phi', justification='c', size=(5, 1))],
)],
],
sg.Frame('Subdomínio 1', size=(190,240), layout=[
[sg.Checkbox('Usar subdomínio', enable_events=True,
            default=False, key='-checkbox_dom1-'),sg.Push()],
[sg.T('Wireframe (.dxf):', justification='r', size=(15, 1)),
 sg.I(key='-dom1_file-', disabled=True, justification='l',
      size=(5, 1)),
 sg.FileBrowse(file_types=(("DXF Files (.dxf)", "*.dxf")),
               button_text='...', disabled=True,
               key='-dom1_button-')],
[sg.T('Nome (sigla):', justification='r', size=(15, 1)),
 sg.I(key='-dom1_name-', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('Densidade (t/m3):', justification='r', size=(15, 1)),
 sg.I(key='dom1_density', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('K - Bulk (GPa):', justification='r', size=(15, 1)),
 sg.I(key='dom1_bulk', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('G - Shear (GPa):', justification='r', size=(15, 1)),
 sg.I(key='dom1_shear', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('Coesão (MPa):', justification='r', size=(15, 1)),
 sg.I(key='dom1_cohesion', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('Ângulo de atrito (°):', justification='r', size=(15, 1)),
 sg.I(key='dom1_phi', disabled=True, justification='c',
      size=(5, 1))],
)],
sg.Frame('Subdomínio 2', size=(190,240), layout=[
[sg.Checkbox('Usar subdomínio', enable_events=True,
            default=False, key='-checkbox_dom2-'),
 sg.Push()],
[sg.T('Wireframe (.dxf):', justification='r', size=(15, 1)),
 sg.I(key='-dom2_file-', disabled=True, justification='l',
      size=(5, 1)),
 sg.FileBrowse(file_types=(("DXF Files (.dxf)", "*.dxf")),
               button_text='...',
               disabled=True, key='-dom2_button-')],
[sg.T('Nome (sigla):', justification='r', size=(15, 1)),
 sg.I(key='-dom2_name-', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('Densidade (t/m3):', justification='r', size=(15, 1)),
 sg.I(key='dom2_density', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('K - Bulk (GPa):', justification='r', size=(15, 1)),
 sg.I(key='dom2_bulk', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('G - Shear (GPa):', justification='r', size=(15, 1)),
 sg.I(key='dom2_shear', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('Coesão (MPa):', justification='r', size=(15, 1)),
 sg.I(key='dom2_cohesion', disabled=True, justification='c',
      size=(5, 1))],
[sg.T('Ângulo de atrito (°):', justification='r',
      size=(15, 1)), sg.I(key='dom2_phi', disabled=True,
      justification='c', size=(5, 1))],
)],
sg.Frame('Subdomínio 3', size=(190,240), layout=[
[sg.Checkbox('Usar subdomínio', enable_events=True,
            default=False, key='-checkbox_dom3-'),
 sg.Push()],
[sg.T('Wireframe (.dxf):', justification='r', size=(15, 1)),
 sg.I(key='-dom3_file-', disabled=True, justification='l',
      size=(5, 1)),

```

### Continuação do arquivo GUI.py

```

        sg.FileBrowse(file_types=("DXF Files (.dxf)", "*.dxf")),
                        button_text="...", disabled=True,
                        key='-dom3_button-']),
    [sg.T('Nome (sigla):', justification='r', size=(15, 1)),
      sg.I(key='-dom3_name-', disabled=True, justification='c',
          size=(5, 1))],
    [sg.T('Densidade (t/m3):', justification='r', size=(15, 1)),
      sg.I(key='dom3_density', disabled=True, justification='c',
          size=(5, 1))],
    [sg.T('K - Bulk (GPa):', justification='r', size=(15, 1)),
      sg.I(key='dom3_bulk', disabled=True, justification='c',
          size=(5, 1))],
    [sg.T('G - Shear (GPa):', justification='r', size=(15, 1)),
      sg.I(key='dom3_shear', disabled=True, justification='c',
          size=(5, 1))],
    [sg.T('Coesão (MPa):', justification='r', size=(15, 1)),
      sg.I(key='dom3_cohesion', disabled=True, justification='c',
          size=(5, 1))],
    [sg.T('Angulo de atrito (°):', justification='r',
          size=(15, 1)), sg.I(key='dom3_phi', disabled=True,
          justification='c', size=(5, 1))],
    ]),
    ]

flac3d_layout = [
    [sg.Frame('Parâmetros', size=(600,110), layout=[
        [sg.T('Arquivo executável do FLAC3D (.exe)', justification='r',
            size=(40, 1)),
          sg.I(key='-flac3d_exe_file-', justification='c', size=(39, 1)),
          sg.FileBrowse(file_types=("EXE Files (.exe)", "*.exe")),
                        button_text="...")],
        [sg.T('Somatório de forças desbalanceadas (default = 1e-4):',
            justification='r', size=(40, 1)),
          sg.I(key='ratio', justification='c', size=(10, 1))],
        [sg.T('Modo determinístico (0 = off, 1 = on):',
            justification='r', size=(40, 1)),
          sg.I(key='det', justification='c', size=(10, 1))]]]),
    [sg.Frame('Bounding Box', size=(600,130), layout=[
        [sg.T('min', justification='r', size=(27, 1)),
          sg.T('max', justification='c', size=(10, 1))],
        [sg.T('Coordenadas X:', justification='r', size=(20, 1)),
          sg.I(key='x_min', justification='c', size=(8, 1)),
          sg.I(key='x_max', justification='c', size=(8, 1)),
          sg.T('N° de elementos em X:', justification='r', size=(20, 1)),
          sg.I(key='x_elem', justification='c', size=(5, 1))],
        [sg.T('Coordenadas Y:', justification='r', size=(20, 1)),
          sg.I(key='y_min', justification='c', size=(8, 1)),
          sg.I(key='y_max', justification='c', size=(8, 1)),
          sg.T('N° de elementos em Y:', justification='r', size=(20, 1)),
          sg.I(key='y_elem', justification='c', size=(5, 1))],
        [sg.T('Coordenadas Z:', justification='r', size=(20, 1)),
          sg.I(key='z_min', justification='c', size=(8, 1)),
          sg.I(key='z_max', justification='c', size=(8, 1)),
          sg.T('N° de elementos em Z:', justification='r', size=(20, 1)),
          sg.I(key='z_elem', justification='c', size=(5, 1))]]]),
    [sg.Frame('Octree', size=(600,100), layout=[
        [sg.T('Subdivisão de segmentos:', justification='r', size=(20, 1)),
          sg.I(key='segments', justification='c', size=(5, 1))],
        [sg.T('Tamanho mín. (m):', justification='r', size=(20, 1)),
          sg.I(key='min_segments', justification='c', size=(5, 1))],
        [sg.T('Extrapolação do limite (m):', justification='r',
            size=(20, 1)), sg.I(key='gap_limit', justification='c',
            size=(5, 1))]]])
    ]

stress_layout = [
    [sg.Frame('Parâmetros', size=(600,80), layout=[
        [sg.T('Horizontal/Vertical stress-ratio:', justification='r',
            size=(40, 1)),
          sg.I(key='stress_ratio', justification='c', size=(10, 1))],
        [sg.T('Overburden (MPa):', justification='r', size=(40, 1)),
          sg.I(key='overburden', justification='c', size=(10, 1))]]])
    ]
]

```

##### GEOMECANICA LAYOUT

Continuação do arquivo *GUI.py*

```

geomecanica_layout = [
    [sg.Frame('', size=(700,500), layout=[
        [sg.TabGroup([[sg.Tab('FLAC3D', flac3d_layout),
            sg.Tab('Propriedades', propriedades_layout),
            sg.Tab('Tensões in situ', stress_layout),
            sg.Tab('Desempenho', desempenho_layout),
            ]], size=(710,600),))]
    ]

algoritmo_layout = [
    [sg.Frame('Parâmetros', size=(600,150), element_justification='c',
        layout=[
            [sg.T('População inicial randômica (0-200):', justification='r',
                size=(30, 1)),
            sg.I(key='popsize', justification='c', size=(10, 1))],
            [sg.T('Número de iterações máximas (0-1000):', justification='r',
                size=(30, 1)),
            sg.I(key='max_iter', justification='c', size=(10, 1))],
            [sg.T('Paciência (0-100):', justification='r', size=(30, 1)),
            sg.I(key='patience', justification='c', size=(10, 1))],
            [sg.T('Mutações forçadas (0-50):', justification='r',
                size=(30, 1)),
            sg.I(key='forced_mutations', justification='c', size=(10, 1))],
            [sg.T('Fator de melhora (0.1):', justification='r', size=(30, 1)),
            sg.I(key='tol', justification='c', size=(10, 1))],
        ]
    ]

resultado_layout = [
    [sg.Frame('Arquivos de saída', size=(600,100),
        element_justification='c', layout=[
            [sg.T('Wireframe do cenário otimizado (.stl)', justification='r',
                size=(30, 1)),
            sg.I(key='-best_cenario_filename-', size=(55, 1)),
            sg.FileSaveAs('...', file_types=(("STL Files (.stl)", "*.stl"),)),
            [sg.T('Relatório do Algoritmo Genético (.html)', justification='r',
                size=(30, 1)),
            sg.I(key='-evolution_report-', size=(55, 1)),
            sg.FileSaveAs('...',
                file_types=(("HTML Files (.html)", "*.html"),)),
            ]],
        ]
    ]
    [sg.Frame('Resultados', size=(600,350), element_justification='c',
        layout=[
            [sg.MLine(autoscroll=True, reroute_stdout=True, reroute_stderr=True,
                key='-OUTPUT-', size=(95,20))]
        ]
    ]
    ]

left_col = [
    [sg.TabGroup([[sg.Tab('Arquivos', arquivos_layout),
        sg.Tab('Parâmetros geométricos', geometricos_layout),
        sg.Tab('Dados econômicos', economicos_layout),
        sg.Tab('Geomecânica', geomecanica_layout),
        sg.Tab('Algoritmo Genético', algoritmo_layout),
        sg.Tab('Executar', resultado_layout)],
        size=(630,500),),
    ]

right_col = [
    [sg.VPush()],
    [sg.VPush()],
    [sg.VPush()],
    [sg.VPush()],
    [sg.B('Otimizar', size=(12,1))],
    [sg.Cancel(button_text='Cancelar', size=(12,1))],
    [sg.VPush()],
    [sg.VPush()],
    [sg.VPush()],
    [sg.I(key='-FILE_OPEN-', visible=False, enable_events=True),
        sg.FileBrowse(file_types = (('Save File (.json)', '*.json*')),
            button_text='Abrir Form', size=(12,1))],
    [sg.I(key='-FILE_SAVE-', visible=False, enable_events=True),
        sg.FileSaveAs(file_types = (('Save File (.json)', '*.json*')),
            default_extension = '.json', button_text='Salvar Form',
            size=(12,1))],
    ]

layout = [[sg.Column(left_col, element_justification='l'),

```



## Continuação do arquivo *GUI.py*

```

        sg.Column(right_col, element_justification='c',
                  vertical_alignment='t')]]

window = sg.Window('OP&RA - Otimizador de Pilares & Realces Abertos',
                  layout, element_justification='l', size=(770,500),
                  font=('Segoe UI', 9, ['normal']))

while True:
    # Event Loop
    event, gui = window.read()
    if event == 'Cancelar' or event == 'Cancel' or event == sg.WIN_CLOSED:
        break

    if event == '-checkbox_dom1-':
        if gui['-checkbox_dom1-'] is True:
            window['-dom1_button-'].update(disabled=False)
            window['-dom1_file-'].update(disabled=False)
            window['-dom1_name-'].update(disabled=False)
            window['dom1_density'].update(disabled=False)
            window['dom1_bulk'].update(disabled=False)
            window['dom1_shear'].update(disabled=False)
            window['dom1_cohesion'].update(disabled=False)
            window['dom1_phi'].update(disabled=False)

        else:
            window['-dom1_button-'].update(disabled=True)
            window['-dom1_file-'].update(disabled=True)
            window['-dom1_name-'].update(disabled=True)
            window['dom1_density'].update(disabled=True)
            window['dom1_bulk'].update(disabled=True)
            window['dom1_shear'].update(disabled=True)
            window['dom1_cohesion'].update(disabled=True)
            window['dom1_phi'].update(disabled=True)

    if event == '-checkbox_dom2-':
        if gui['-checkbox_dom2-'] is True:
            window['-dom2_button-'].update(disabled=False)
            window['-dom2_file-'].update(disabled=False)
            window['-dom2_name-'].update(disabled=False)
            window['dom2_density'].update(disabled=False)
            window['dom2_bulk'].update(disabled=False)
            window['dom2_shear'].update(disabled=False)
            window['dom2_cohesion'].update(disabled=False)
            window['dom2_phi'].update(disabled=False)

        else:
            window['-dom2_button-'].update(disabled=True)
            window['-dom2_file-'].update(disabled=True)
            window['-dom2_name-'].update(disabled=True)
            window['dom2_density'].update(disabled=True)
            window['dom2_bulk'].update(disabled=True)
            window['dom2_shear'].update(disabled=True)
            window['dom2_cohesion'].update(disabled=True)
            window['dom2_phi'].update(disabled=True)

    if event == '-checkbox_dom3-':
        if gui['-checkbox_dom3-'] is True:
            window['-dom3_button-'].update(disabled=False)
            window['-dom3_file-'].update(disabled=False)
            window['-dom3_name-'].update(disabled=False)
            window['dom3_density'].update(disabled=False)
            window['dom3_bulk'].update(disabled=False)
            window['dom3_shear'].update(disabled=False)
            window['dom3_cohesion'].update(disabled=False)
            window['dom3_phi'].update(disabled=False)

        else:
            window['-dom3_button-'].update(disabled=True)
            window['-dom3_file-'].update(disabled=True)
            window['-dom3_name-'].update(disabled=True)
            window['dom3_density'].update(disabled=True)
            window['dom3_bulk'].update(disabled=True)
            window['dom3_shear'].update(disabled=True)
            window['dom3_cohesion'].update(disabled=True)
            window['dom3_phi'].update(disabled=True)

    if event == '-FILE_OPEN-':

```



Continuação do arquivo *GUI.py*

```

openfile = gui['-FILE_OPEN-']
sg.user_settings_load(openfile)
for elem in gui:
    gui[elem] = sg.user_settings_get_entry(elem)
    window[elem].update(gui[elem])
if gui['-checkbox_dom1-'] is True:
    window['-dom1_button-'].update(disabled=False)
    window['-dom1_file-'].update(disabled=False)
    window['-dom1_name-'].update(disabled=False)
    window['dom1_density'].update(disabled=False)
    window['dom1_bulk'].update(disabled=False)
    window['dom1_shear'].update(disabled=False)
    window['dom1_cohesion'].update(disabled=False)
    window['dom1_phi'].update(disabled=False)

else:
    window['-dom1_button-'].update(disabled=True)
    window['-dom1_file-'].update(disabled=True)
    window['-dom1_name-'].update(disabled=True)
    window['dom1_density'].update(disabled=True)
    window['dom1_bulk'].update(disabled=True)
    window['dom1_shear'].update(disabled=True)
    window['dom1_cohesion'].update(disabled=True)
    window['dom1_phi'].update(disabled=True)

if gui['-checkbox_dom2-'] is True:
    window['-dom2_button-'].update(disabled=False)
    window['-dom2_file-'].update(disabled=False)
    window['-dom2_name-'].update(disabled=False)
    window['dom2_density'].update(disabled=False)
    window['dom2_bulk'].update(disabled=False)
    window['dom2_shear'].update(disabled=False)
    window['dom2_cohesion'].update(disabled=False)
    window['dom2_phi'].update(disabled=False)

else:
    window['-dom2_button-'].update(disabled=True)
    window['-dom2_file-'].update(disabled=True)
    window['-dom2_name-'].update(disabled=True)
    window['dom2_density'].update(disabled=True)
    window['dom2_bulk'].update(disabled=True)
    window['dom2_shear'].update(disabled=True)
    window['dom2_cohesion'].update(disabled=True)
    window['dom2_phi'].update(disabled=True)

if gui['-checkbox_dom3-'] is True:
    window['-dom3_button-'].update(disabled=False)
    window['-dom3_file-'].update(disabled=False)
    window['-dom3_name-'].update(disabled=False)
    window['dom3_density'].update(disabled=False)
    window['dom3_bulk'].update(disabled=False)
    window['dom3_shear'].update(disabled=False)
    window['dom3_cohesion'].update(disabled=False)
    window['dom3_phi'].update(disabled=False)

else:
    window['-dom3_button-'].update(disabled=True)
    window['-dom3_file-'].update(disabled=True)
    window['-dom3_name-'].update(disabled=True)
    window['dom3_density'].update(disabled=True)
    window['dom3_bulk'].update(disabled=True)
    window['dom3_shear'].update(disabled=True)
    window['dom3_cohesion'].update(disabled=True)
    window['dom3_phi'].update(disabled=True)

window['-OUTPUT-'].update('')

if event == '-FILE_SAVE-':
    event, gui = window.read()
    savefile = gui['-FILE_SAVE-']
    sg.user_settings_filename(savefile)
    for elem in gui:
        sg.user_settings_set_entry(elem, gui[elem])

sg.user_settings_delete_entry("...")
sg.user_settings_delete_entry("...0")

```

Continuação do arquivo *GUI.py*

```
sg.user_settings_delete_entry("...1")
sg.user_settings_delete_entry("...2")
sg.user_settings_delete_entry("...3")
sg.user_settings_delete_entry("...4")
sg.user_settings_delete_entry("...5")
sg.user_settings_delete_entry("...6")
sg.user_settings_delete_entry("...7")
sg.user_settings_delete_entry("-dom1_button-")
sg.user_settings_delete_entry("-dom2_button-")
sg.user_settings_delete_entry("-dom3_button-")
sg.user_settings_delete_entry("-OUTPUT-")
sg.user_settings_delete_entry("Salvar Form")
sg.user_settings_delete_entry("Abrir Form")
sg.user_settings_delete_entry("-FILE_SAVE-")
sg.user_settings_delete_entry("-FILE_OPEN-")
sg.user_settings_save(savefile)

if event == 'Otimizar':
    window['-OUTPUT-'].update('')
    print("Otimizando...")
    #try:
    config.gui = gui
    GA.otimizador()
    #except:
    #    print('\nOtimização cancelada..')
print(event, gui)
window.close()
return gui
```

**APÊNDICE B – CÓDIGOS ESCRITOS NO FLAC3D**Arquivo *master.f3dat*

```
1 FILE: master.f3dat, DATE: qui 7. jul 15:03:09 2022
2 -----
3 ; Define a função que estabelece a conexão socket
4
5 def open_socket
6   array data(1)
7   array config(200)
8   ; Abre a conexão socket
9   s = socket.open(0,1,3333)
10  c = socket.read(config, 53, 1)
11  command
12    ; Inicia a importação das geometrias e faz adensamento
13    call 'octree'
14    ; Inicializa as tensões e condições de borda do modelo
15    call 'initial'
16    ; Escava os subníveis (galerias)
17    call 'excavation'
18  end_command
19  ; Fecha a conexão socket
20  data(1) = -666
21  b = socket.write(data, 1, 1)
22  b = socket.close(1)
23  command
24    program exit
25  end_command
26 end
27 [ open_socket ]
```

Arquivo *octree.f3dat*

```

1 FILE: octree.f3dat, DATE: qui 7. jul 15:06:07 2022
2 -----
3 geometry import [config(52)]
4 geometry import 'fw_boundingbox.dxf'
5 geometry import 'hw_boundingbox.dxf'
6 geometry import 'galerias.dxf'
7
8 ; Cria uma caixa
9 zone create brick size ([config(1)],[config(2)],[config(3)]) ...
10 point 0 ([config(4)],[config(5)],[config(6)]) ...
11 point 1 ([config(7)],[config(5)],[config(6)]) ...
12 point 2 ([config(4)],[config(8)],[config(6)]) ...
13 point 3 ([config(4)],[config(5)],[config(9)]) ...
14 group 'orezone_box'
15
16 zone face skin slot "Any" ;identifica as faces da caixa
17
18 ; Adensamento da zona mineralizada pelo metodo octree
19 zone densify segments [config(12)] gradient-limit maximum-length [config(11)] repeat ...
20 range geometry-distance 'orezone' gap [config(10)] extent
21
22 ; Adensamento das galerias pelo metodo octree
23 zone densify segments [config(12)] gradient-limit maximum-length [config(11)] repeat ...
24 range geometry-distance 'galerias' gap [config(10)] extent
25
26 ; Agrupar elemento coincidentes, para garantir que esta tudo conectado
27 zone attach by-face
28
29 ; Cria grupos
30 zone group 'footwall' range geometry-space 'fw_boundingbox' count odd
31 zone group 'hangingwall_box' range geometry-space 'hw_boundingbox' count odd
32 zone group 'orezone' range geometry-space 'orezone' count odd
33 zone group 'galerias' range geometry-space 'galerias' count odd
34 zone group 'near_hw' range geometry-distance 'galerias' gap 5 geometry-space 'galerias' count 2
   direction (0,1,0)
35
36 define f_vol_galerias
37   global v_galerias = 0
38   loop foreach local zone zone.list
39     s = zone.group(zone)
40     if s == 'galerias' then
41       v_galerias = v_galerias + zone.vol(zone)
42     endif
43   end_loop
44 end
45 @f_vol_galerias
46
47 ; Gera plotagem
48 plot create 'octree_groups'
49 plot clear
50 plot active on
51 plot background 'white'
52 plot outline active on width 2 color 'black'
53 plot legend active on heading color 'black' copyright color 'black' ...
54 placement left size 25,50 ...
55 step active off ...
56 time-real active off ...
57 time-model active off ...
58 title-customer active off ...
59 view-info active off
60 plot title-job active off
61 plot title active off
62 plot view projection parallel magnification 1 ...
63 center (203.01,69.667,31.363) eye (-50.158,69.667,31.363) roll 0.000 ...
64 clip-front -1e+10 clip-back 1e+10
65 plot item create axes active on ...
66 axis-x color 'red' draw-positive on draw-negative off label-positive 'X' ...
67 axis-y color 'lime' draw-positive on draw-negative off label-positive 'Y' ...
68 axis-z color 'blue' draw-positive on draw-negative off label-positive 'Z' ...
69 screen 10 10 scale 5 ...

```

Continuação do arquivo *octree.f3dat*

```
70 font size 10 family 'Arial' style bold ...
71 transparency 0 ...
72 legend active off
73 plot item create zone active on ...
74 label group slot 'Default' ...
75 color-list global on clear ...
76 label "rockmass" color 'lightgray' ...
77 label "near_hw" color 'red' ...
78 label "orezone" color 'green' ...
79 label "galerias" color 'pink' ...
80 polygons fill on outline active on width 1 color 'black' ...
81 outline-transparency 80 lighting on offset 0.5 2 selected-only off selected-highlight off ...
82 null-faces-only off ...
83 map axis xyz translate (0,0,0) scale (1,1,1) ...
84 deformation-factor active off ...
85 hide-null mechanical on thermal off fluid off ...
86 transparency 0 ...
87 cut active on type plane ...
88 surface on front off back off ...
89 origin (203.015,52.8473,16.4671) normal (1,0,0) ...
90 clip active off ...
91 legend active on ...
92 title active on text "" size 55 family 'Times New Roman' style bold color 'black' ...
93 cut ...
94 title active on size 44 family 'Arial' style normal color 'black' ...
95 origin active off ...
96 normal active off ...
97 orientation active off ...
98 map active on size 44 family 'Arial' style normal color 'black' ...
99 hide-null active off ...
100 deformation-factor active on size 44 family 'Arial' style normal color 'black' ...
101 color-list label active on size 44 family 'Arial' style normal color 'black' ...
102 heading active on text "" size 44 family 'Arial' style normal color 'black' ...
103 count active off
104 plot 'octree_groups' export pdf filename 'initial_zone_groups_section'
105
106 model save 'octree'
```

Arquivo *initial.f3dat*

```

1  FILE: initial.f3dat, DATE: qui 7. jul 15:03:51 2022
2  -----
3  ; Constitutive Model and Properties
4  zone cmodel assign mohr-coulomb
5
6  ; orezone
7  zone property density [config(20)] ...
8      bulk [config(17)] ...
9      shear [config(18)] ...
10     cohesion [config(19)] ...
11     friction [config(21)] ...
12     range group 'orezone_box' or 'orezone' or 'galerias'
13
14 ; hangingwall
15 zone property density [config(25)] ...
16     bulk [config(22)] ...
17     shear [config(23)] ...
18     cohesion [config(24)] ...
19     friction [config(26)] ...
20     range group 'hangingwall_box' or 'near_hw'
21
22 ; footwall
23 zone property density [config(30)] ...
24     bulk [config(27)] ...
25     shear [config(28)] ...
26     cohesion [config(29)] ...
27     friction [config(31)] ...
28     range group 'footwall'
29
30 ; Verifica se o subdominio 1 esta sendo utilizado
31 def dom1
32     if config(47) == 1 then
33         command
34             geometry import 'dom1.dxf'
35             zone property density [config(35)]...
36                 bulk [config(32)]...
37                 shear [config(33)]...
38                 cohesion [config(34)]...
39                 friction [config(36)]...
40                 range geometry-space 'dom1' count odd
41         end_command
42     endif
43 end
44 @dom1
45
46 ; Verifica se o subdominio 2 esta sendo utilizado
47 def dom2
48     if config(48) == 1 then
49         command
50             geometry import 'dom2.dxf'
51             zone property density [config(40)]...
52                 bulk [config(37)]...
53                 shear [config(38)]...
54                 cohesion [config(39)]...
55                 friction [config(41)]...
56                 range geometry-space 'dom2' count odd
57         end_command
58     endif
59 end
60 @dom2
61
62 ; Verifica se o subdominio 3 esta sendo utilizado
63 def dom3
64     if config(49) == 1 then
65         command
66             geometry import 'dom3.dxf'
67             zone property density [config(45)]...
68                 bulk [config(42)]...
69                 shear [config(43)]...
70                 cohesion [config(44)]...

```

Continuação do arquivo *initial.f3dat*

```
71 friction [config(46)]...
72 range geometry-space 'dom3' count odd
73     end_command
74 endif
75 end
76 @dom3
77
78 ; Boundary Conditions (roller boundaries - velocity-normal)
79 zone face apply velocity-normal 0 range group "orezone_boxWest" or "orezone_boxEast"
80 zone face apply velocity-normal 0 range group "orezone_boxNorth" or "orezone_boxSouth"
81 zone face apply velocity-normal 0 range group "orezone_boxBottom"
82 zone face apply stress-normal [config(16)] range group "orezone_boxTop"
83
84 ; Initial Conditions
85 model gravity 9.81
86 zone initialize-stresses ratio [config(15)] overburden [config(16)]
87
88 ; Solve to initial equilibrium
89 model history mechanical ratio
90 def model_det
91     if config(13) = 1 then
92         command
93             model deterministic on
94         end_command
95     else
96         command
97             model deterministic off
98         end_command
99     endif
100 end
101 @model_det
102
103 model large-strain off
104 model solve elastic
```

Arquivo *excavation.f3dat*

```

1 FILE: excavation.f3dat, DATE: qui 7. jul 15:04:40 2022
2 -----
3 ; Reset displacements and velocities for reference
4 zone gridpoint initialize displacement (0,0,0)
5 zone gridpoint initialize velocity (0,0,0)
6
7 ; Excavate
8 zone delete range group 'galerias'
9
10 ; Salva o historico do somatorio de forças
11 model history mechanical ratio
12
13 ; Verifica configuração do modo deterministico
14 def model_det
15     if config(13) = 1 then
16         command
17             model deterministic on
18         end_command
19     else
20         command
21             model deterministic off
22         end_command
23     endif
24 end
25 @model_det
26
27 ; Solve to equilibrium after excavation
28 model solve
29
30 ; Plotagem do gráfico do somatório de forças
31 plot create 'forces'
32 plot clear
33 plot active on
34 plot background 'white'
35 plot outline active on width 2 color 'black'
36 plot legend active on heading color 'black' copyright color 'black' ...
37     placement left size 25,50 ...
38     step active off ...
39     time-real active off ...
40     time-model active off ...
41     title-customer active off ...
42     view-info active off
43 plot title-job active off
44 plot title active off
45 plot view projection perspective magnification 1 ...
46     center (0,0,0) eye (-0.79117,-2.6992,-0.23998) roll 0.000 ...
47     clip-front -1e+10 clip-back 1e+10
48 plot item create chart-history active on ...
49     history '1' name "1 mechanical ratio limit" style line reversed-x off reversed-y off ...
50     style-line width 2 style solid color 'cyan' ...
51     alias "" ...
52     vs step reversed off ...
53     begin 0 end 0 skip 0 ...
54     axis-x log off minimum auto maximum auto inside on ...
55         label "Step" exponent show on value 3 auto on ...
56     axis-y log off minimum auto maximum auto inside on ...
57         label "Y-Axis" exponent show on value -2 auto on ...
58     title active on text "" size 25 family 'Arial' style bold ...
59     labels active on size 20 family 'Times New Roman' style bold ...
60     marks active on size 18 family 'Times New Roman' style normal ...
61     chart width 2 box off color 'black' position 1 1 size 98 98 ...
62     grid active off ...
63     transparency 0 ...
64     legend active on ...
65         title active on text "" size 55 family 'Times New Roman' style bold color 'black' ...
66         series active on size 44 family 'Arial' style normal color 'black' ...
67         range off ...
68         vs active on text "" size 44 family 'Arial' style normal color 'black' ...
69         vs-range off
70 plot 'forces' export pdf filename 'initial_mech_ratio'
```



Continuação do arquivo *excavation.f3dat*

```

71
72 ; Plotagem da geometria inicial com galerias escavadas
73 plot create 'gal_excavation'
74 plot clear
75 plot active on
76 plot background 'white'
77 plot outline active on width 2 color 'black'
78 plot legend active on heading color 'black' copyright color 'black' ...
79   placement left size 25,50 ...
80   step active off ...
81   time-real active off ...
82   time-model active off ...
83   title-customer active off ...
84   view-info active off
85 plot title-job active off
86 plot title active off
87 plot view projection parallel magnification 1 ...
88   center (203.01,69.667,31.363) eye (-50.158,69.667,31.363) roll 0.000 ...
89   clip-front -1e+10 clip-back 1e+10
90 plot item create axes active on ...
91   axis-x color 'red' draw-positive on draw-negative off label-positive 'X' ...
92   axis-y color 'lime' draw-positive on draw-negative off label-positive 'Y' ...
93   axis-z color 'blue' draw-positive on draw-negative off label-positive 'Z' ...
94   screen 10 10 scale 5 ...
95   font size 10 family 'Arial' style bold ...
96   transparency 0 ...
97   legend active off
98 plot item create zone active on ...
99   label group slot 'Default' ...
100  color-list global on clear ...
101   label "rockmass" color 'lightgray' ...
102   label "near_hw" color 'red' ...
103   label "orezone" color 'green' ...
104   label "galerias" color 'pink' ...
105   polygons fill on outline active on width 1 color 'black' ...
106   outline-transparency 80 lighting on offset 0.5 2 selected-only off selected-highlight off
...
107  null-faces-only off ...
108  map axis xyz translate (0,0,0) scale (1,1,1) ...
109  deformation-factor active off ...
110  hide-null mechanical on thermal off fluid off ...
111  transparency 0 ...
112  cut active on type plane ...
113   surface on front off back off ...
114   origin (203.015,52.8473,16.4671) normal (1,0,0) ...
115  clip active off ...
116  legend active on ...
117   title active on text "" size 55 family 'Times New Roman' style bold color 'black' ...
118   cut ...
119   title active on size 44 family 'Arial' style normal color 'black' ...
120   origin active off ...
121   normal active off ...
122   orientation active off ...
123   map active on size 44 family 'Arial' style normal color 'black' ...
124   hide-null active off ...
125   deformation-factor active on size 44 family 'Arial' style normal color 'black' ...
126   color-list label active on size 44 family 'Arial' style normal color 'black' ...
127   heading active on text "" size 44 family 'Arial' style normal color 'black' ...
128   count active off
129 plot 'gal_excavation' export pdf filename 'initial_gal_excavation'
130 ;
131 model save 'excavated_galerias'

```

Arquivo *py\_stop.es.f3dat*

```
1 FILE: py_stop.es.f3dat, DATE: qui 7. jul 15:04:57 2022
2 -----
3 ; Abre o resultado do modelo com galerias escavadas
4 model restore 'excavated_galerias'
5
6 ; Abre a conexao socket
7 def open_socket
8     s = socket.open(0,1,3333)
9     socket.read(config, 53, 1)
10    command
11        ; chama o arquivo que escava os realces
12        call 'excavation_stop.es'
13    end_command
14    data(1) = -666
15    b = socket.write(data, 1, 1)
16    b = socket.close(1)
17    command
18        program exit
19    end_command
20 end
21 [ open_socket ]
```

Arquivo *excavation\_stopes.f3dat*

```

1 FILE: excavation_stopes.f3dat, DATE: qui 7. jul 15:05:08 2022
2 -----
3 ; Importa cenario atual de realces
4 geometry import 'stopes.stl'
5
6 ; Reclassifica os elementos conforme o cenario
7 zone group 'stopes' range geometry-space 'stopes' count odd
8
9 ; Importa cenario atual de pilares
10 geometry import 'pilares.stl'
11
12 ; Reclassifica os elementos conforme o cenario
13 zone group 'pilares' range geometry-space 'pilares' count odd
14
15 ; Reclassifica os elementos conforme o cenario
16 zone group 'near_hw' range geometry-distance 'stopes' gap 10 geometry-space 'stopes' count 2
17 direction (0,1,0)
18
19 ; Função que calcula o volume dos realces
20 define f_vol_stopes
21   global v_stopes = 0
22   loop foreach local zone zone.list
23     s = zone.group(zone)
24     if s == 'stopes' then
25       v_stopes = v_stopes + zone.vol(zone)
26     endif
27   end_loop
28 @f_vol_stopes
29
30 ; Função que calcula o volume dos pilares
31 define f_vol_pilares
32   global v_pilares = 0
33   loop foreach local zone zone.list
34     s = zone.group(zone)
35     if s == 'pilares' then
36       v_pilares = v_pilares + zone.vol(zone)
37     endif
38   end_loop
39 end
40 @f_vol_pilares
41
42 ; Reset displacements and velocities for reference
43 zone gridpoint initialize displacement (0,0,0)
44 zone gridpoint initialize velocity (0,0,0)
45
46 ; Excava os realces
47 zone delete range group 'stopes'
48
49 ; Gera o historico do somatorio de forças
50 model history mechanical ratio
51
52 ; Verifica a configuração para o modo deterministico
53 def model_det
54   if config(13) = 1 then
55     command
56       model deterministic on
57     end_command
58   else
59     command
60       model deterministic off
61     end_command
62   endif
63 end
64 @model_det
65
66 ; Faz o somatorio de forças conforme o ratio definido
67 model solve ratio [config(14)]
68
69 ; Calcula o PoF dos pilares e volume de elementos com falha

```

Continuação do arquivo *excavation\_stopes.f3dat*

```

70 fish define check_ssr_pilares
71     vol = 0
72     sum = 0
73     cont = 0
74     num = 0
75     global pof_ssr = 0
76     loop foreach local zone zone.list
77         s = zone.group(zone)
78         if s == 'pilares' then
79             num = num + 1
80             s1 = (zone.stress.min(zone))*-1
81             s3 = (zone.stress.max(zone))*-1
82             phi = zone.prop(zone, 'friction')
83             c = zone.prop(zone, 'cohesion')
84             k = (1 + math.sin(math.degrad*phi))/(1-math.sin(math.degrad*phi))
85             sc = (2 * c * math.cos(math.degrad*phi))/(1 - (math.sin(math.degrad*phi)))
86             s1_ = sc + (k * s3)
87             if (s1-s3)#0 then
88                 ssr = math.abs((s1_ - s3)/(s1 - s3))
89                 if ssr < 1.01 then
90                     vol = vol + zone.vol(zone)
91                     sum = sum + ssr*zone.vol(zone)
92                     cont = cont + 1
93                 endif
94             endif
95         endif
96     end_loop
97     ssr_mean = sum/vol
98     pof_ssr = (cont*100.0)/num
99     pillar_vol_failure = vol
100    io.out(string.build("PoF SSR %1 %", pof_ssr))
101 end
102 @check_ssr_pilares
103
104 ; Calcula volume de diluição no HW e porcentagem
105 fish define check_s3
106     vol = 0
107     sum = 0
108     global dil = 0
109     global dil_vol = 0
110     loop foreach local zone zone.list
111         s = zone.group(zone)
112         if s == 'near_hw' then
113             s3 = (zone.stress.max(zone))*-1
114             if s3<config(26) then
115                 vol = vol + zone.vol(zone)
116                 sum = sum + s3*zone.vol(zone)
117             endif
118         endif
119     end_loop
120     s3_mean = sum/vol
121     dil = (vol*100.0)/(vol+v_stopes+v_galerias)
122     dil_vol = vol
123 end
124 @check_s3
125
126 ; Calcula recuperação global
127 fish define global_rec
128     global rec = 0
129     rec = ((v_stopes+v_galerias)*100.0)/(v_stopes+v_galerias+v_pilares)
130     io.out(string.build("Global REC %1 %", rec))
131 end
132 @global_rec
133
134 ; Grava resultados no arquivo results.dat
135 fish define results
136     array aa(20)
137     local status = file.open('results.dat', 1, 1)
138     aa(1) = 'pof_ssr'
139     aa(2) = ''

```

Continuação do arquivo *excavation\_stopes.f3dat*

```

140     aa(3) = 'dil'
141     aa(4) = 'rec'
142     aa(5) = 'dil_vol'
143     aa(6) = 'pillar_vol_failure'
144     aa(7) = pof_ssr
145     aa(8) = ''
146     aa(9) = dil
147     aa(10) = rec
148     aa(11) = dil_vol
149     aa(12) = pillar_vol_failure
150     status = file.write(aa,12)
151     status = file.close
152 end
153 @results
154
155 ; Plot mech-history
156 plot create 'forces'
157 plot clear
158 plot active on
159 plot background 'white'
160 plot outline active on width 2 color 'black'
161 plot legend active on heading color 'black' copyright color 'black' ...
162     placement left size 25,50 ...
163     step active off ...
164     time-real active off ...
165     time-model active off ...
166     title-customer active off ...
167     view-info active off
168 plot title-job active off
169 plot title active off
170 plot view projection perspective magnification 1 ...
171     center (0,0,0) eye (-0.79117,-2.6992,-0.23998) roll 0.000 ...
172     clip-front -1e+10 clip-back 1e+10
173 plot item create chart-history active on ...
174     history '1' name "1 mechanical ratio limit" style line reversed-x off reversed-y off ...
175     style-line width 2 style solid color 'cyan' ...
176     alias "" ...
177     vs step reversed off ...
178     begin 0 end 0 skip 0 ...
179     axis-x log off minimum auto maximum auto inside on ...
180     label "Step" exponent show on value 3 auto on ...
181     axis-y log off minimum auto maximum auto inside on ...
182     label "Y-Axis" exponent show on value -2 auto on ...
183     title active on text "" size 25 family 'Arial' style bold ...
184     labels active on size 20 family 'Times New Roman' style bold ...
185     marks active on size 18 family 'Times New Roman' style normal ...
186     chart width 2 box off color 'black' position 1 1 size 98 98 ...
187     grid active off ...
188     transparency 0 ...
189     legend active on ...
190     title active on text "" size 55 family 'Times New Roman' style bold color 'black' ...
191     series active on size 44 family 'Arial' style normal color 'black' ...
192     range off ...
193     vs active on text "" size 44 family 'Arial' style normal color 'black' ...
194     vs-range off
195 plot 'forces' export pdf filename 'scenario_mech_ratio'
196
197 ; Plot geometries
198 plot create 'geometries'
199 plot clear
200 plot active on
201 plot background 'white'
202 plot outline active on width 2 color 'black'
203 plot legend active on heading color 'black' copyright color 'black' ...
204     placement left size 25,50 ...
205     step active off ...
206     time-real active off ...
207     time-model active off ...
208     title-customer active off ...
209     view-info active off

```

Continuação do arquivo *excavation\_stopes.f3dat*

```

210 plot title-job active off
211 plot title active off
212 plot view projection parallel magnification 1 ...
213 center (168.012,65.5466,20.263) eye (368.331,-134.772,183.822) roll 0.0000 ...
214 clip-front -1e+10 clip-back 1e+10
215 plot item create geometry active on ...
216 node state free pixel-size 2 ...
217 edge state free width 1 style solid ...
218 polygon state on fill on outline active on width 1 color 'black' ...
219 polygon-transparency 0 ...
220 outline-transparency 80 lighting on offset 0.5 2 cut-line width 1 ...
221 color-by label set ...
222 color-list global on clear ...
223 label "orezone" active on color 'cyan' ...
224 label "galerias" active on color 'pink' ...
225 label "stopes" active on color 'seagreen' ...
226 label "pilares" active on color 'yellow' ...
227 map axis xyz translate (0,0,0) scale (1,1,1) ...
228 transparency 0 ...
229 cut active off ...
230 clip active off ...
231 legend active on ...
232 title active off label active on size 44 family 'Arial' style normal color 'black' ...
233 map active on size 44 family 'Arial' style normal color 'black' heading active on text
"Geometrias" size 44 family 'Arial' style bold color 'black' ...
234 import-progress active on size 44 family 'Arial' style normal color 'black'
235 plot item create axes active on ...
236 axis-x color 'red' draw-positive on draw-negative off label-positive 'X' ...
237 axis-y color 'lime' draw-positive on draw-negative off label-positive 'Y' ...
238 axis-z color 'blue' draw-positive on draw-negative off label-positive 'Z' ...
239 screen 10 10 scale 5 ...
240 font size 10 family 'Arial' style bold ...
241 transparency 0 ...
242 legend active off
243 plot item create data-label active on ...
244 text active on size 10 family 'Times New Roman' style normal ...
245 uniform-color off ...
246 color-list global off clear ...
247 scale target 0.05 value 1 ...
248 draw-as arrow shape (3,30,10) quality 20 ...
249 by-magnitude off ...
250 color 'black' ...
251 transparency 0 ...
252 legend active on ...
253 title active on text "Dilution (%)" size 44 family 'Arial' style bold color 'black' ...
254 label active on size 44 family 'Arial' style normal color 'black' ...
255 heading active on text [dil] size 44 family 'Arial' style normal color 'black' ...
256 maximum active off size 44 family 'Arial' style normal color 'black' ...
257 scale active off size 44 family 'Arial' style normal color 'black' ...
258 vector active off size 44 family 'Arial' style normal color 'black'
259 plot item create data-label active on ...
260 text active on size 10 family 'Times New Roman' style normal ...
261 uniform-color off ...
262 color-list global off clear ...
263 scale target 0.05 value 1 ...
264 draw-as arrow shape (3,30,10) quality 20 ...
265 by-magnitude off ...
266 color 'black' ...
267 transparency 0 ...
268 legend active on ...
269 title active on text "Global Recovery (%)" size 44 family 'Arial' style bold color 'black'
...
270 label active on size 44 family 'Arial' style normal color 'black' ...
271 heading active on text [rec] size 44 family 'Arial' style normal color 'black' ...
272 maximum active off size 44 family 'Arial' style normal color 'black' ...
273 scale active off size 44 family 'Arial' style normal color 'black' ...
274 vector active off size 44 family 'Arial' style normal color 'black'
275 plot item create data-label active on ...
276 text active on size 10 family 'Times New Roman' style normal ...
277 uniform-color off ...

```

Continuação do arquivo *excavation\_stopes.f3dat*

```

278 color-list global off clear ...
279 scale target 0.05 value 1 ...
280 draw-as arrow shape (3,30,10) quality 20 ...
281 by-magnitude off ...
282 color 'black' ...
283 transparency 0 ...
284 legend active on ...
285 title active on text "PoF SSR (%)" size 44 family 'Arial' style bold color 'black' ...
286 label active on size 44 family 'Arial' style normal color 'black' ...
287 heading active on text [pof_ssr] size 44 family 'Arial' style normal color 'black' ...
288 maximum active off size 44 family 'Arial' style normal color 'black' ...
289 scale active off size 44 family 'Arial' style normal color 'black' ...
290 vector active off size 44 family 'Arial' style normal color 'black'
291 plot 'geometries' export pdf filename 'scenario_geometries'
292
293 ; Plot iso s3
294 plot create 'iso_s3'
295 plot clear
296 plot active on
297 plot background 'white'
298 plot outline active on width 2 color 'black'
299 plot legend active on heading color 'black' copyright color 'black' ...
300 placement left size 25,50 ...
301 step active off ...
302 time-real active off ...
303 time-model active off ...
304 title-customer active off ...
305 view-info active off
306 plot title-job active off
307 plot title active off
308 plot view projection parallel magnification 1 ...
309 center (168.012,65.5466,20.263) eye (368.331,-134.772,183.822) roll 0.0000 ...
310 clip-front -1e+10 clip-back 1e+10
311 plot item create geometry active on ...
312 type label color-by set ...
313 color-list global off clear ...
314 label "galerias" active on color 'pink' ...
315 label "stopes" active on color 'green' ...
316 label "pilares" active on color 'yellow' ...
317 point size 2 ...
318 line width 1 style solid ...
319 polygons fill on outline active off width 1 color 'black' ...
320 outline-transparency 80 lighting on offset 0.5 2 ...
321 cut-line width 4 ...
322 node-state free edge-state free poly-state on ...
323 set "galerias" set "stopes" all-sets off ...
324 map axis xyz translate (0,0,0) scale (0.8,0.8,0.8) ...
325 transparency 70 ...
326 cut active off ...
327 clip active off ...
328 legend active on ...
329 title active on text "" size 55 family 'Times New Roman' style bold color 'black' ...
330 map active on size 44 family 'Arial' style normal color 'black' ...
331 color-list label active on size 44 family 'Arial' style normal color 'black' ...
332 heading active on text "" size 44 family 'Arial' style normal color 'black'
333 plot item create zone-isosurface active on ...
334 stress quantity maximum stress on log off ...
335 method inverse-distance-weighting power 3 radius-ratio 0.75 null fluid off mechanical on
thermal off ...
336 isovalue -200000 surface none autocalc on ...
337 color-by group slot 'Default' ...
338 color-list global off clear ...
339 label "rockmass" color 'red' ...
340 label "near_hw" color 'red' ...
341 label "pilares" color 'pink' ...
342 polygons fill on outline active off ...
343 polygon-transparency 0 ...
344 outline-transparency 80 lighting on offset 0.5 2 ...
345 cut-line width 1 ...
346 hide-null mechanical on thermal off fluid off ...

```

Continuação do arquivo *excavation\_stopes.f3dat*

```

347 map axis xyz translate (0,0,0) scale (1,1,1) ...
348 deformation-factor active off ...
349 transparency 0 ...
350 cut active off ...
351 clip active off ...
352 legend active on ...
353 title active off ...
354 method active off ...
355 color-by active off ...
356 color-list label active off ...
357 heading active on text "Isosuperfície sig3 = 0.2 MPa" size 44 family 'Arial' style bold
color 'black' ...
358 count active off ...
359 map active off ...
360 deformation-factor active on size 44 family 'Arial' style bold color 'black'
361 plot item create axes active on ...
362 axis-x color 'red' draw-positive on draw-negative off label-positive 'X' ...
363 axis-y color 'lime' draw-positive on draw-negative off label-positive 'Y' ...
364 axis-z color 'blue' draw-positive on draw-negative off label-positive 'Z' ...
365 screen 10 10 scale 5 ...
366 font size 10 family 'Arial' style bold ...
367 transparency 0 ...
368 legend active off
369 plot item create data-label active on ...
370 text active on size 10 family 'Times New Roman' style normal ...
371 uniform-color off ...
372 color-list global off clear ...
373 scale target 0.05 value 1 ...
374 draw-as arrow shape (3,30,10) quality 20 ...
375 by-magnitude off ...
376 color 'black' ...
377 transparency 0 ...
378 legend active on ...
379 title active on text "Dilution (%)" size 44 family 'Arial' style bold color 'black' ...
380 label active on size 44 family 'Arial' style normal color 'black' ...
381 heading active on text [dil] size 44 family 'Arial' style normal color 'black' ...
382 maximum active off size 44 family 'Arial' style normal color 'black' ...
383 scale active off size 44 family 'Arial' style normal color 'black' ...
384 vector active off size 44 family 'Arial' style normal color 'black'
385 plot item create data-label active on ...
386 text active on size 10 family 'Times New Roman' style normal ...
387 uniform-color off ...
388 color-list global off clear ...
389 scale target 0.05 value 1 ...
390 draw-as arrow shape (3,30,10) quality 20 ...
391 by-magnitude off ...
392 color 'black' ...
393 transparency 0 ...
394 legend active on ...
395 title active on text "Global Recovery (%)" size 44 family 'Arial' style bold color 'black'
...
396 label active on size 44 family 'Arial' style normal color 'black' ...
397 heading active on text [rec] size 44 family 'Arial' style normal color 'black' ...
398 maximum active off size 44 family 'Arial' style normal color 'black' ...
399 scale active off size 44 family 'Arial' style normal color 'black' ...
400 vector active off size 44 family 'Arial' style normal color 'black'
401 plot item create data-label active on ...
402 text active on size 10 family 'Times New Roman' style normal ...
403 uniform-color off ...
404 color-list global off clear ...
405 scale target 0.05 value 1 ...
406 draw-as arrow shape (3,30,10) quality 20 ...
407 by-magnitude off ...
408 color 'black' ...
409 transparency 0 ...
410 legend active on ...
411 title active on text "PoF SSR (%)" size 44 family 'Arial' style bold color 'black' ...
412 label active on size 44 family 'Arial' style normal color 'black' ...
413 heading active on text [pof_ssr] size 44 family 'Arial' style normal color 'black' ...
414 maximum active off size 44 family 'Arial' style normal color 'black' ...

```



Continuação do arquivo *excavation\_stopes.f3dat*

```

415         scale active off size 44 family 'Arial' style normal color 'black' ...
416         vector active off size 44 family 'Arial' style normal color 'black'
417 plot 'iso_s3' export pdf filename 'scenario_iso_s3'
418
419 ; PLOT SSR
420 plot create 'iso_ssr'
421 plot clear
422 plot active on
423 plot background 'white'
424 plot outline active on width 2 color 'black'
425 plot legend active on heading color 'black' copyright color 'black' ...
426     placement left size 25,42 ...
427     step active off ...
428     time-real active off ...
429     time-model active off ...
430     title-customer active off ...
431     view-info active off
432 plot title-job active off
433 plot title active off
434 plot view projection parallel magnification 1 ...
435     center (122.53,40.72,9.943) eye (253.09,-89.841,116.55) roll 0.000 ...
436     clip-front -1e+10 clip-back 1e+10
437 plot item create geometry active on ...
438     type label color-by set ...
439     color-list global off clear ...
440         label "galerias" active on color 'pink' ...
441         label "stopes" active on color 'green' ...
442         label "pilares" active on color 'yellow' ...
443     point size 2 ...
444     line width 1 style solid ...
445     polygons fill on outline active off width 1 color 'black' ...
446     outline-transparency 80 lighting on offset 0.5 2 ...
447     cut-line width 4 ...
448     node-state free edge-state free poly-state on ...
449     set "galerias" set "stopes" all-sets off ...
450     map axis xyz translate (0,0,0) scale (0.8,0.8,0.8) ...
451     transparency 70 ...
452     cut active off ...
453     clip active off ...
454     legend active on ...
455         title active on text "" size 55 family 'Times New Roman' style bold color 'black' ...
456         map active on size 44 family 'Arial' style normal color 'black' ...
457         color-list label active on size 44 family 'Arial' style normal color 'black' ...
458         heading active on text "" size 44 family 'Arial' style normal color 'black'
459 plot item create zone-isosurface active on ...
460     stress-strength-ratio log off ...
461     method inverse-distance-weighting power 3 radius-ratio 0.75 null fluid off mechanical off
462     thermal off ...
463     isovalue 1.05 surface none autocalc on ...
464     color-by group slot 'Default' ...
465     color-list global off clear ...
466         label "hangingwall_box" color 'none' ...
467         label "near_hw" color 'none' ...
468         label "orezone_box" color 'none' ...
469         label "footwall" color 'none' ...
470         label "pilares" color 'cardinal' ...
471     polygons fill on outline active off ...
472     outline-transparency 80 lighting on offset 0.5 2 ...
473     cut-line width 1 ...
474     hide-null mechanical on thermal off fluid off ...
475     map axis xyz translate (0,0,0) scale (0.8,0.8,0.8) ...
476     deformation-factor active off ...
477     transparency 0 ...
478     cut active off ...
479     clip active off ...
480     legend active on ...
481         title active on text "" size 55 family 'Times New Roman' style bold color 'black' ...
482         method active on size 44 family 'Arial' style normal color 'black' ...
483         color-by active on size 44 family 'Arial' style normal color 'black' ...
484         color-list label active on size 44 family 'Arial' style normal color 'black' ...

```

Continuação do arquivo *excavation\_stopes.f3dat*

```
484     heading active on text "Iso-surface value 1.05" size 44 family 'Arial' style normal color
      'black' ...
485     count active off ...
486     map active on size 44 family 'Arial' style normal color 'black' ...
487     deformation-factor active on size 44 family 'Arial' style normal color 'black'
488 plot item create axes active on ...
489     axis-x color 'red' draw-positive on draw-negative off label-positive 'X' ...
490     axis-y color 'lime' draw-positive on draw-negative off label-positive 'Y' ...
491     axis-z color 'blue' draw-positive on draw-negative off label-positive 'Z' ...
492     screen 10 10 scale 5 ...
493     font size 10 family 'Arial' style bold ...
494     transparency 0 ...
495     legend active off
496 plot 'iso_ssr' export pdf filename 'scenario_iso_ssr'
```