

Modelagem de um Algoritmo Genético para o Problema de Seleção de Minérios

Queli Terezinha Schmitz¹, Rafael Stubs Parpinelli¹

¹Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina
(UDESC – Joinville)

Campus Universitário – CEP 89228-100 – Joinville – SC – Brasil

queli.schmitz@datasul.com.br, parpinelli@joinville.udesc.br

Abstract. *This paper proposes a Genetic Algorithm (GA) to solve the Ore Selection Problem (OSP). Due to the high complexity degree of the problem, involving a large number of variables and restrictions, believes that GA could be a robust method to find a feasible and optimized solution inside the vast and complex considered search space. The results obtained with the GA implemented are presented and a comparison is made with other optimization method for the OSP.*

Resumo. *Este artigo propõe um Algoritmo Genético (AG) para resolver o Problema da Seleção de Minérios (PSM). Devido ao alto grau de complexidade do problema, envolvendo um grande número de variáveis e restrições, acredita-se que AG seja um método robusto para se encontrar uma solução viável e otimizada dentro do vasto e complexo espaço de busca considerado. São apresentados os resultados obtidos com a implementação da modelagem proposta e uma comparação com outro método de otimização para o PSM é realizada.*

Palavras-chave. *Algoritmos Genéticos, blending, seleção de minérios.*

1. Introdução

Algoritmos Genéticos (AGs) são métodos de busca e otimização baseados no conceito de evolução dos seres vivos proposto por Darwin. É a meta-heurística mais difundida e estudada dentro da CE, pela sua flexibilidade, relativa simplicidade de implementação e eficácia em realizar busca global em ambientes diversos [Goldberg 1989].

Este artigo apresenta uma proposta de AG para o Problema de Seleção de Minérios (PSM), que corresponde a uma das operações de controle de qualidade de uma mineradora. Esta atividade começa com a amostragem de minério da mina. A seguir, na usina de beneficiamento, todas as amostras são submetidas a uma análise química e granulométrica, para verificar a forma e teores de elementos químicos presentes. Considerando-se as variações na análise química e granulométrica existem pelo menos vinte diferentes qualidades de minérios. Estes minérios são depositados em um pátio na forma de montes (pilhas). Estas, na forma como foram distribuídas, podem atender à demanda do mercado, e por isto, podem ser consideradas como produtos.

O problema consiste da retirada de minério de diversas pilhas, de modo que seja possível fornecer para a usina de tratamento ou um cliente externo, um produto que

esteja de acordo com as especificações técnicas relativas à qualidade e quantidade [Pinto e Merschmann 2001].

2. AG para o Problema de Seleção de Minérios

Para ilustrar o mecanismo de solução do PSM utilizando AG, a Figura 1 apresenta uma configuração que dispõe de n pilhas com diferentes quantidades de recursos naturais. Como resultado, deve-se obter uma configuração com, desejavelmente, um número mínimo de pilhas possível, que atenda também as especificações do produto final.

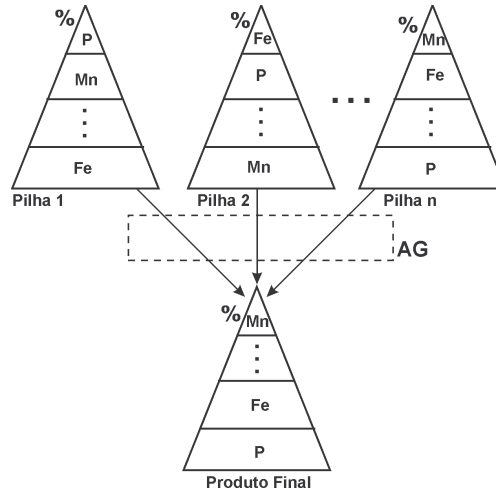


Figura 1. Modelo Proposto para utilizar AG no PSM

2.1. Modelagem Matemática do AG para o PSM

O objetivo do AG é maximizar a produção de forma a atingir a quantidade solicitada de produto e, preferencialmente, utilizar o número mínimo de pilhas possível. O espaço de busca definido para o PSM está diretamente ligado às variáveis que representam as quantidades disponíveis em cada pilha:

$$q_i : [0..Q_i] \quad \forall i, i \in [1..n] \quad (1)$$

Onde: n é o número de pilhas de minério; q_i é a quantidade utilizada da pilha i e Q_i é a quantidade disponível na pilha i . Cada variável q_i é representada por L_{bits} (a modelagem utiliza uma codificação binária das variáveis). L_i terá uma quantidade de $bits$ de forma a cobrir o domínio da variável Q_i (gene), dado por:

$$2^{L_i-1} < Q_i < 2^{L_i} \quad \forall i, i \in [1..n] \quad (2)$$

L_i é chamada de L_{gene}^i . As quantidades retiradas de cada pilha para compor o produto final correspondem às variáveis de decisão que irão compor um indivíduo da população, ou seja, uma solução para o problema. A Figura 2 ilustra a disposição das variáveis de decisão nos indivíduos.

q_1	q_2	...	q_n
-------	-------	-----	-------

Figura 2. Representação de um Indivíduo

O tamanho do indivíduo ou cromossomo, dado pelo número de *bits* que o compõe, é determinado pela soma dos *bits* de cada gene, como visto na Equação (3):

$$L_{cromossomo} = \sum_{i=1}^n L_{gene\ i} \quad (3)$$

O espaço de busca do AG é de $2^{L_{cromossomo}}$ pontos. Para o exemplo utilizado para testar a modelagem proposta, retirado de [Pinto e Merschmann 2001], tem-se um cromossomo de 98 *bits*.

A função de *fitness* apresentada na Equação (4), está definida de modo a contemplar o objetivo de maximizar a quantidade produzida, penalizar indivíduos que infringem as restrições (de qualidade e quantidade) e bonificar indivíduos que utilizam um número reduzido de pilhas.

$$Fitness = F_{Obj} + r \cdot P + s \cdot B \quad (4)$$

Onde: r é o fator de penalidade; P é o valor da função de penalidade normalizada; s é o fator de bonificação e B é o valor da função de bonificação normalizada. A Função Objetivo (F_{Obj}) corresponde à soma obtida com a mistura das quantidades utilizadas de cada pilha. F_{Obj} também é um valor normalizado.

É necessário respeitar a faixa de tolerância das quantidades dos componentes químicos, que determina a qualidade do produto final. Isto é obtido pela Equação (5).

$$l\inf_c \leq \left(\sum_{i=1}^n t_{ci} \cdot q_i \right) \leq l\sup_c \quad \forall c, c \in [1..m] \quad (5)$$

Onde: c é o componente químico; m representa o número de componentes químicos; q_i é a quantidade utilizada da pilha i ; $l\inf_c$ é a quantidade mínima admissível para o componente c ; $l\sup_c$ é a quantidade máxima admissível para o componente c e t_{ci} é o teor do componente c na pilha i . Tem-se então duas penalidades: a primeira ($h1_c$), para soluções cujo somatório de teores multiplicados pelas quantidades utilizadas da pilha é menor do que o limite inferior e a segunda ($h2_c$) para soluções cujo produto do somatório pelas quantidades utilizadas da pilha é maior do que o limite superior: Estas penalidades estão modeladas nas equações (6) e (7), respectivamente.

$$h1_c = \frac{\max\{0, l\inf_c - \sum_{i=1}^n (t_{ci} \cdot q_i)\}}{l\inf_c} \quad \forall c, c \in [1..m] \quad (6)$$

$$h2_c = \frac{\max\{0, \sum_{i=1}^n (t_{ci} \cdot q_i) - l\sup_c\}}{MS_c - l\sup_c} \quad \forall c, c \in [1..m] \quad (7)$$

É necessário também respeitar a quantidade de produto requisitada, ou seja, $\sum_{i=1}^n q_i \leq Q_{Total}$. Os indivíduos que infringirem esta restrição serão penalizados pelo valor de $h3$, visto na Equação (8).

$$h3 = \frac{\max\{0, \sum_{i=1}^n q_i - Q_{Total}\}}{Q_{Total}} \quad (8)$$

Onde Q_{Total} é a quantidade requisitada do produto. A função de penalidade definida para este AG tem como objetivo diminuir o valor da função de *fitness* dos indivíduos que infringirem as restrições. A função de penalidade está definida na Equação (9).

$$P = \frac{\sum_{c=1}^m h1_c + \sum_{c=1}^m h2_c + h3}{nc_{inf} + nc_{sup} + 1} \quad \forall c, c \in [1..m] \quad (9)$$

Onde: m é o número de componentes químicos; $h1_c$ é o valor da penalidade de qualidade do produto para limite inferior; $h2_c$ é o valor da penalidade de qualidade do produto para limite superior; $h3$ é o valor da penalidade caso a quantidade produzida seja maior que a requisitada; nc_{inf} é o número de componentes que infringem a restrição de limite inferior e nc_{sup} é o número de componentes que infringem a restrição de limite superior.

A bonificação dos indivíduos neste problema ocorre somente para soluções que não infringirem as restrições, cujo objetivo é bonificar indivíduos que utilizam um número reduzido de pilhas. A função de bonificação é dada pela Equação (10).

$$B_i = \frac{n - n_{pi}}{n} \quad \forall i, i \in [1..n] \quad (10)$$

Onde: n é o número de pilhas e n_{pi} é o número de pilhas usadas pelo indivíduo i .

Este trabalho propõe uma Seleção por Torneio Estocástico. O elitismo é simples e preserva sempre para a próxima geração o melhor indivíduo que surge. Optou-se por utilizar o cruzamento de um ponto com probabilidade de 80%. A mutação tem uma probabilidade de ocorrência padrão de 1%. O critério para finalizar o AG poderá ser o limite de gerações ou a estagnação do mesmo. Como para o problema inicial abordado a solução ótima é conhecida, o AG pode parar quando convergir para tal configuração.

3. Implementação Computacional

Para a implementação do modelo proposto, foi utilizado o pacote de programas GALOPPS versão 3.2 [Goodman 1996]. É flexível, gratuito, escrito em Linguagem C e baseado na arquitetura do SGA (*Simple Genetic Algorithm*) de Goldberg, o que facilita o entendimento e utilização.

Foi criada uma biblioteca, que contém as estruturas, funções e variáveis especificamente para o PSM, a fim de separar as definições de funções para o PSM das funções padrão do GALOPPS. As funções definidas na biblioteca, são responsáveis pela importação dos dados contidos nos arquivos texto de entrada para o problema: *componentes.txt* e *ilhas.txt*.

As Figuras 3 e 4 ilustram os arquivos de entrada que contém os dados necessários para o PSM abordado [Pinto e Merschmann 2001]. O arquivo *ilhas.txt* (Figura 3) contém, respectivamente: o número que identifica a ilha (1ª coluna), as

porcentagens dos componentes envolvidos no problema e a última coluna corresponde à quantidade disponível de material nesta pilha. No arquivo *componentes.txt* (Figura 4) estão o número que identifica o componente (1ª coluna), o teor mínimo (%), máximo e ideal de cada um deles no resultado (teores desejados).

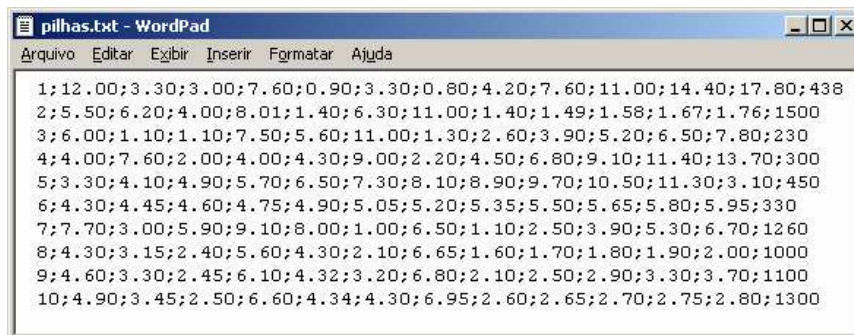


Figura 3. Arquivo de entrada *pilhas.txt*



Figura 4. Arquivo de entrada *componentes.txt*

No momento da execução do AG, é chamada uma função para carregar as informações contidas nestes dois arquivos para uma estrutura *Pilha*, que por sua vez contém uma sub-estrutura *Componentes*.

4. Resultados

Como resultado da simulação realizada, é apresentado um relatório que mostra a evolução do AG, contendo o pior *fitness*, o médio e melhor de cada geração. Também é apresentada a quantidade produzida, o número de pilhas utilizadas, quantidades retiradas das pilhas e teores do produto. Para efeito de comparações, são apresentadas informações de entrada do problema como quantidades disponíveis nas pilhas e teores desejados para os componentes do produto. A Figura 5 apresenta os resultados obtidos para o exemplo estudado.

Geração	Fitness											
	Mínimo	Médio	Máximo									
1	0.1424602114	0.2070236444	0.4061709661									
2	0.1288474008	0.3237460980	0.4850784016									
3	0.0563635505	0.3607481205	0.4890126896									
.												
5000	0.0000000000	0.8996909878	0.9411229135									

Total Produzido: 5070 toneladas												
Numero de Pilhas: 7												
	Pilha:											
Qtd.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		

Utiliz.	300	1455	0	139	450	330	0	0	1096	1300		
Disp.	438	1500	230	300	450	330	1260	1000	1100	1300		

	Componente:											
Teor	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12

Produz.	5.22	4.43	3.29	6.68	3.52	5.02	7.57	3.03	3.50	3.97	4.44	4.11

Ideal	4.71	3.84	2.26	6.12	4.32	4.67	6.54	3.10	3.55	4.01	4.46	4.11
Mínimo	0.80	1.10	1.00	3.00	1.60	2.20	4.45	1.00	3.50	2.00	3.00	2.10
Máximo	13.00	22.00	22.00	8.80	15.00	11.00	24.00	6.00	16.00	17.00	10.00	4.12

Figura 5. Relatório de resultados do AG para o Exemplo Estudado

Os testes de execução do AG foram feitos utilizando uma população de 100 indivíduos e um número máximo de gerações de 5000. Ao fator de penalidade (r) foi atribuído o valor -10 , a fim de baixar consideravelmente o valor do *fitness* dos indivíduos que infringem as restrições. Este valor baixo também se deve ao fato de que a função de penalidade é composta por 3 restrições. Aos indivíduos que não são penalizados (não infringem nenhuma restrição), é dada uma bonificação de acordo com o número de pilhas utilizadas. Ao fator de bonificação (s) foi atribuído o valor $+1$, para não aumentar demais o valor do *fitness*, que pode chegar a no máximo $1,9$ (quando houver quantidade disponível em somente uma pilha e esta for utilizada para resolver o PSM).

Além deste caso, o Algoritmo Genético implementado para o Problema de Seleção de Minérios foi utilizado em outro exemplo amostrado de Pinto e Merschmann (2001). O computador utilizado foi um AMD Duron 750 MHz, 128 Mb de memória RAM e HD de 40 Gb, que mostrou desempenho satisfatório para execução dos exemplos, que serão apresentados nas próximas seções.

4.1. Resultados para o Exemplo Amostrado

Os teores desejados para os 12 componentes envolvidos no problema exemplo, que possui 10 pilhas de minério, cuja quantidade a ser produzida é de 5070 toneladas são apresentados na Tabela 1. O somatório dos percentuais não chega a 100% pois o restante representa material que não faz parte do controle de qualidade do produto.

Tabela 1. Teores desejados para o produto

Componente	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Teor Mínimo	0,80	1,10	1,00	3,00	1,60	2,20	4,45	1,00	3,50	2,00	3,00	2,10
Teor Máximo	13,00	22,00	22,00	8,80	15,00	11,00	24,00	6,00	16,00	17,00	10,00	4,12
Teor Ideal	4,71	3,84	2,26	6,12	4,32	4,67	6,54	3,10	3,55	4,01	4,46	4,11

A Tabela 2 mostra os teores do produto obtido com a evolução do AG, sendo que todos os componentes respeitam as faixas de valores admissíveis para eles. Nesta

tabela também é apresentado o desvio obtido com relação à quantidade ideal de cada componente.

Tabela 2. Teores obtidos utilizando AG

Componente	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Teor Produzido	5,22	4,43	3,29	6,68	3,52	5,02	7,57	3,03	3,50	3,97	4,44	4,11
Desvio	0,51	0,59	1,03	0,56	-0,80	0,35	1,03	-0,07	-0,05	-0,04	-0,02	0,00

O mesmo exemplo foi testado em uma implementação comercial que utiliza o software Lingo, desenvolvido por Lindo Systems Inc., utilizado para resolver problemas de otimização envolvendo Programação Linear, Programação Linear Inteira, Programação Linear Inteira Binária, Programação Não-Linear, entre outros.

O Lingo é uma biblioteca de funções matemáticas e estatísticas, e permite que os dados sejam armazenados em arquivos externos. Para resolver o PSM, o modelo comercial utiliza Programação Inteira Mista, fazendo uso de variáveis binárias para selecionar as pilhas que serão utilizadas na solução. A principal técnica utilizada pelo Lingo para resolver este problema é a de “*Branch-and-Bound*”, que se baseia na geração de uma árvore de decisão que divide todo o espaço de soluções (“*Branch*”). Os limites inferiores e superiores (“*Bound*”) da função objetivo, quando alcançados, permitem eliminar todo um sub-ramo da árvore de decisão, facilitando a obtenção do resultado otimizado [Akabay 1996]. Os valores destacados na linha “Desvio” da Tabela 2 representam os componentes que chegaram mais próximos dos teores ideais com relação à implementação comercial.

A Tabela 3 apresenta os teores obtidos para o produto utilizando a solução comercial que emprega o método “*Branch-and-Bound*”. Esta, por sua vez, infringiu o limite superior máximo admissível para o componente número doze (C12). Sete dos doze componentes apresentam diferenças maiores com relação aos teores ideais do que a solução utilizando AG.

Tabela 3. Teores obtidos usando o Lingo

Componente	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Teor Produzido	5,54	4,23	3,73	7,13	4,32	4,97	6,69	3,07	3,91	4,76	5,61	5,66
Desvio	0,83	0,39	1,47	1,01	0,00	0,30	0,15	-0,03	0,36	0,75	1,15	1,55

A Tabela 4 apresenta as quantidades retiradas de cada pilha de minério pelas soluções usando AG e usando o software Lingo. A solução comercial devolveu uma solução que atendeu a demanda de 5070 toneladas e para isto foram utilizadas 8 pilhas de minério para compor a solução, sendo que o AG também obteve a quantidade solicitada (5070 ton) e utilizou apenas 7 das dez pilhas disponíveis.

Tabela 4. Quantidades Retiradas de Cada Pilha – Ex.1

Pilha	1	2	3	4	5	6	7	8	9	10	Qtotal
QAG	300	1455	0	139	450	330	0	0	1096	1300	5.070
Qlingo	438	1125,70	230	300	450	330	896,30	0	0	1300	5.070

O resultado obtido para o exemplo utilizando AG é melhor do que o encontrado por Pinto e Merschmann (2001) no que concerne ao número de pilhas, pois utilizou 9 pilhas de minério para produzir as 5070 toneladas. Neste artigo não havia o parâmetro “Teores Ideais”, somente uma faixa de valores admissíveis para o produto final (mínimo e máximo de cada componente). Devido a este fato, os teores obtidos por esta solução

foram considerados como “Teores Ideais” para o AG e não foram comparados. Para alcançar o resultado desejado para o Exemplo Amostrado 1 foram necessárias 5000 gerações (executadas em 26 segundos).

5. Conclusões

Para determinar os parâmetros que seriam utilizados no AG, foram realizados vários testes, alterando os valores dos coeficientes de penalidade e bonificação, utilização do fator de *crowding*, entre outros. O algoritmo foi executado utilizando valores baixos para o fator de penalidade ($r=-1$, $r=-2$), porém os resultados obtidos não foram satisfatórios. Isto se deve ao fato de a função de penalidade ser composta por três restrições ($h1$, $h2$ e $h3$), o que implica na necessidade de diminuir o seu peso na determinação do *fitness* do indivíduo.

Verificou-se que quanto mais componentes envolvidos no problema, mais gerações são necessárias para que o AG encontre uma solução que atenda os objetivos estabelecidos. Isto se deve ao número de cálculos adicionais para cada indivíduo da população (restrições para os limites inferior e superior de cada componente). No Exemplo estudado, onde há 12 componentes envolvidos, foram necessárias 5000 gerações para obter o resultado. O número de pilhas envolvidas no problema também é um fator bastante relevante, pois determina o tamanho do cromossomo.

Os resultados mostram que a utilização de uma função de bonificação para aumentar o valor do *fitness* dos indivíduos, de acordo com o número de pilhas utilizadas e o baixo valor para o fator de penalidade, são pertinentes para alcançar os objetivos desejados. Outro parâmetro que foi utilizado é o fator de *crowding*, que para o Exemplo estudado reduziu o número de gerações de 10000 para 5000 para obter o resultado esperado.

Com base nos resultados obtidos utilizando o software Lingo, é possível perceber apesar de enumerar todas as soluções, o método “*Branch-and-Bound*” é mais suscetível a entrar em regiões de “ótimos locais”. A modelagem utilizando AG se mostrou bastante eficiente e eficaz frente ao PSM. Trabalhos futuros podem se dirigir para um ajuste automático dos parâmetros do AG ou para o desenvolvimento de uma *interface* amigável para interação usuário/sistema.

Referências Bibliográficas

- Akbay, K. S. (1996) “Using simulation optimization to find the best solution”, IIE Solutions, v28. n5 p. 24-26.
- Goldberg, D. E. (1989) “Genetic Algorithms in Search, Optimization and Machine Learning”, Reading: Addison-Wesley.
- Goodman, E. D. (1996) “An Introduction to GALOPPS”, East Lansing: Michigan State University.
- Pinto, L. R.; Merschmann, L. H. C. (2001) “Planejamento Operacional da Lavra de Mina Usando Modelos Matemáticos”, REM: Ouro Preto, v. 54 n. 9, p. 211-214.