UM ALGORITMO HEURÍSTICO HÍBRIDO PARA O PLANEJAMENTO OPERACIONAL DE LAVRA

Igor Machado Coelho¹, Sabir Ribas¹, Marcone Jamilson Freitas Souza¹, Vitor Nazário Coelho¹

¹Universidade Federal de Ouro Preto, Departamento de Computação, Ouro Preto, Minas Gerais, Brasil

imcoelho@iceb.ufop.br; sabir@iceb.ufop.br; marcone@iceb.ufop.br; vncoelho@gmail.com

Abstract – This paper presents a hybrid algorithm based on Iterated Local Search metaheuristic. In order to test this algorithm, it is applied to a problem which requires fast answers, i.e., the open-pit mining problem. The proposed heuristic algorithm uses GRASP to generate an initial solution and has Variable Neighborhood Descent as a method of local search. Periodically, a mathematical programming module is triggered to solve smaller parts of the problem in optimality. To validate the proposed hybrid method its results are compared to those produced by its pure heuristic version and to the CPLEX solver. The results show the efficiency of the proposed algorithm.

Keywords – Open Pit Mining, GRASP, Variable Neighborhood Descent, Iterated Local Search, Metaheuristics

Resumo – Este trabalho apresenta um algoritmo híbrido baseado na metaheurística *Iterated Local Search* (ILS). De forma a testá-lo, ele é aplicado a um problema que requer decisões rápidas, o problema de planejamento operacional de lavra em minas a céu aberto. O algoritmo heurístico proposto utiliza GRASP para gerar uma solução inicial e tem o *Variable Neighborhood Descent* como método de busca local. Periodicamente, é acionado um módulo de programação matemática para resolver na otimalidade partes menores do problema. Para validá-lo, seus resultados são comparados com aqueles produzidos por sua versão sem o módulo de programação matemática, bem como com os do otimizador CPLEX. Experimentos computacionais mostram a efetividade da proposta.

Palavras-chave – Planejamento operacional de lavra, GRASP, *Variable Neighborhood Descent*, *Iterated Local Search*, Metaheurísticas

1. Introdução

Este trabalho trata do problema de planejamento operacional de lavra com alocação dinâmica de caminhões (POLAD). O objetivo é determinar o ritmo de lavra em cada frente, seja ela de minério ou de estéril, de forma que as metas requeridas de produção e qualidade sejam satisfeitas e o número de caminhões necessários ao processo seja reduzido.

De acordo com [1], este problema pertence à classe NP-difícil, uma vez que tem como subproblema, o Problema da Mochila Múltipla, o qual é NP-difícil. Em vista da dificuldade de resolvê-lo na otimalidade, propõe-se um algoritmo heurístico híbrido para encontrar soluções sub-ótimas. O algoritmo

proposto é baseado na metaheurística *Iterated Local Search* - ILS [2] e utiliza GRASP [3] para gerar uma solução inicial, *Variable Neighborhood Descent* - VND [4] para fazer busca local e tem um módulo de programação matemática para resolver na otimalidade partes menores do problema. O algoritmo proposto representa uma evolução em relação ao de [5], por incluir um módulo de programação matemática.

Dado que a tomada de decisão no problema em pauta deve ser rápida, o desafio é produzir soluções de boa qualidade no menor tempo. Assim, comparações com outras metodologias são feitas tendo-se como foco a qualidade da solução em um dado tempo de processamento.

O restante deste trabalho está organizado como segue. Na Seção 2 são apresentados os trabalhos correlatos. A Seção 3 apresenta a metodologia para a resolução do problema. Na Seção 4 é descrito o algoritmo proposto e a Seção 5 mostra os resultados dos experimentos computacionais. A Seção 6 conclui o trabalho.

2. Trabalhos Correlatos

Em [6] foi proposto um algoritmo, baseado em programação linear e dinâmica, que é a base para o sistema *DISPATCH*, o qual vem operando em muitas minas em todo o mundo [7].

Um modelo de programação linear por metas foi apresentado em [8] para resolver um problema de mistura de minérios no planejamento de curto prazo em uma mina de carvão. O objetivo foi maximizar um critério econômico e reduzir a soma dos desvios dos parâmetros individuais de qualidade e da tonelagem em relação aos valores desejados.

Um sistema para o despacho ótimo de caminhões em uma mineração de ferro foi desenvolvido em [9] com o objetivo de minimizar o tempo de fila da frota de caminhões, aumentar a produtividade desta e melhorar a qualidade do minério lavrado. O sistema tem o Algoritmo Genético com ferramenta de otimização e é a base do sistema SMART MINE, atualmente muito utilizado em várias minas brasileiras.

Um sistema de otimização e simulação, chamado de OTISIMIN, foi desenvolvido em [10] para análise de cenário de produção em minas a céu aberto. Esse modelo de otimização foi aperfeiçoado posteriormente por [11], incluindo-se restrições de atendimento às metas de produção e qualidade. Em [1] foi proposta uma modelagem heurística para o problema, baseada em *Variable Neighborhood Search* (VNS). O algoritmo proposto explora o espaço de soluções com base em seis tipos de movimentos, sendo estes os primeiros descritos na subseção 3.2 do presente trabalho. Pelos experimentos realizados, o VNS foi capaz de gerar soluções de melhor qualidade em um menor tempo de processamento que aquelas obtidas pelo modelo exato, resolvido pelo LINGO.

Em [12], o problema é resolvido por um algoritmo heurístico, denominado GVILS, que combina os procedimentos heurísticos GRASP, VND e ILS. Além dos movimentos de [1], o algoritmo GVILS faz uso de mais outros dois movimentos para explorar o espaço de soluções. Na modelagem heurística considerou-se, também, a minimização do número de veículos. Usando quatro problemas-teste da literatura, o GVILS foi comparado com o otimizador CPLEX 9.1 aplicado a um modelo de programação matemática. Foram realizados testes envolvendo 15 minutos de processamento. Em dois dos problemas, o algoritmo proposto mostrou-se bastante superior; enquanto nos dois outros ele foi competitivo com o CPLEX, produzindo soluções médias com valores até 0,08% piores, na média.

Em [5], são feitos testes com tempos limite menores, comparando a abordagem heurística proposta com o otimizador CPLEX 9.1 aplicado ao mesmo modelo matemático de [12]. Observou-se uma melhora no desempenho da abordagem heurística proposta em relação à resolução do problema com *solvers* exatos quando o tempo de solução é um fator crítico. Os testes foram feitos com um limite de dois minutos de processamento.

Tais resultados motivaram o desenvolvimento do presente trabalho, que busca explorar as características vantajosas de ambas as abordagens heurística e exata.

3. Metodologia

3.1 Modelo Exato

A formulação de programação matemática usada neste trabalho é a mesma de [12], a qual é uma evolução do modelo de [1]. Nesta formulação, considera-se a função de avaliação dada pela Equação (1):

$$\begin{aligned} \min \sum_{j \in T} \lambda_j^- d_j^- + \sum_{j \in T} \lambda_j^+ d_j^+ + \alpha^- P_m^- + \alpha^+ P_m^+ \\ + \beta^- P_e^- + \beta^+ P_e^+ + \sum_{l \in V} \omega_l U_l \end{aligned} \quad (1)$$

Na Equação (1) busca-se minimizar os desvios positivos (d_j^+) e negativos (d_j^-) das metas de cada parâmetro de controle j da mistura, bem como minimizar os desvios positivos e negativos das metas de produção de minério e estéril, representados pelas variáveis de decisão P_m^+ , P_m^- , P_e^+ e P_e^- , respectivamente. Nesta função também considera-se a minimização do número de veículos utilizados, representado pela variável binária U_l , que vale 1 se o veículo l for utilizado e 0, caso contrário.

As constantes λ_j^- , λ_j^+ , α^- , α^+ , β^- , β^+ e ω_l são pesos que refletem a importância de cada componente da função objetivo.

3.2 Modelo Heurístico

Representação de uma solução

Uma solução é representada por uma matriz $R = [Y|N]$, sendo Y a matriz $|F| \times 1$ e N a matriz $|F| \times |V|$. Cada célula y_i da matriz $Y_{|F| \times 1}$ representa a carregadeira k alocada à frente i . O valor -1 significa que não existe carregadeira alocada. Se não houver viagens feitas a uma frente i , a carregadeira k associada a tal frente é considerada *inativa* e não é penalizada por produção abaixo da mínima para este equipamento de carga.

Na matriz $N_{|F| \times |V|}$, cada célula n_{il} representa o número de viagens do caminhão $l \in V$ a uma frente $i \in F$. O valor 0 (zero) significa que não há viagem para aquele caminhão. O valor -1 informa a incompatibilidade entre o caminhão e a carregadeira alocada àquela frente.

Avaliação de uma solução

Como os movimentos desenvolvidos podem gerar soluções inviáveis, uma solução é avaliada por uma função f baseada em penalidades, dada pela Equação (2). Nesta equação, a componente f_s^M é a função objetivo propriamente dita (mesma do modelo matemático - Equação (1)), e avalia s quanto ao atendimento às metas de produção e qualidade, bem como número de caminhões utilizados. As demais componentes penaliza a ocorrência de inviabilidade na solução corrente. Assim, $f^p(s)$ avalia s quanto ao desrespeito aos limites de produção estabelecidos para a quantidade de minério e estéril; $f_j^q(s)$ avalia s quanto à inviabilidade em relação ao j -ésimo parâmetro de controle; $f_l^u(s)$ avalia s quanto ao desrespeito do atendimento da taxa de utilização máxima do l -ésimo caminhão e $f_k^c(s)$ avalia s quanto ao desrespeito aos limites de produtividade da k -ésima carregadeira.

$$f(s) = f^M(s) + f^p(s) + \sum_{j \in T} f_j^q(s) + \sum_{l \in V} f_l^u(s) + \sum_{k \in C} f_k^c(s) \quad (2)$$

Geração de uma solução inicial

Uma solução inicial para o problema é obtida por um procedimento construtivo parcialmente guloso, semelhante à da fase de construção GRASP. A construção é feita em duas etapas. As alocações das

carregadeiras e a distribuição das viagens às frentes são feitas, na primeira etapa, às frentes de estéril, e na segunda, às frentes de minério.

Esta estratégia é adotada tendo em vista que nas frentes de estéril o importante é atender à produção, não sendo necessário observar a qualidade. A classificação dos elementos candidatos a ser inseridos na solução é feita considerando que, para as frentes de estéril a melhor frente é a que possui a maior massa, a melhor carregadeira é a que oferece a maior produção e o melhor caminhão é o de maior capacidade. Já para as frentes de minério considera-se que a melhor frente é a que possui o menor desvio dos teores em relação às metas, a melhor carregadeira é a que oferece a maior produção e o melhor caminhão é o de menor capacidade.

A escolha da frente na segunda etapa da construção (minério) é feita usando-se uma função guia, tal como em [13]. Inicialmente, todas as frentes candidatas são ordenadas tendo em vista os valores dos desvios das metas de qualidade. À r -ésima frente melhor classificada, é associada a função $bias(r) = 1/(2r)$. A seguir, cada frente candidata é escolhida com probabilidade $p(r) = bias(r)/\sum bias(r)$. Assim, dá-se maior probabilidade de escolha à frente que mais contribui para minimizar os desvios das metas.

Estruturas de Vizinhança

Para explorar o espaço de soluções do problema foram usados oito movimentos, apresentados a seguir, sendo os seis primeiros os de [1].

Movimento Número de Viagens - $N^{NV}(s)$. Este movimento consiste em aumentar ou diminuir o número de viagens de um caminhão l em uma frente i , onde esteja operando um equipamento de carga compatível. Desta maneira, neste movimento uma célula n_{il} da matriz N tem seu valor acrescido ou decrescido de uma unidade.

Movimento Carga - $N^{CG}(s)$. Consiste em trocar duas células distintas y_i e y_k da matriz Y , ou seja, trocar os equipamentos de carga que operam nas frentes i e k , caso as duas frentes possuam equipamentos de carga alocados. Havendo apenas uma frente com equipamento de carga, esse movimento consistirá em realocar o equipamento de carga à frente disponível. Para manter a compatibilidade entre carregadeiras e caminhões, as viagens feitas às frentes são realocadas junto com as frentes escolhidas.

Movimento Realocar Viagem de um Caminhão - $N^{VC}(s)$. Consiste em selecionar duas células n_{il} e n_{kl} da matriz N e repassar uma unidade de n_{il} para n_{kl} . Assim, um caminhão l deixa de realizar uma viagem em uma frente i para realizá-la em outra frente k . Restrições de compatibilidade entre equipamentos são respeitadas neste movimento, havendo realocação de viagens apenas quando houver compatibilidade entre eles.

Movimento Realocar Viagem de uma Frente - $N^{VF}(s)$. Duas células n_{il} e n_{ik} da matriz N são selecionadas e uma unidade de n_{il} é realocada para n_{ik} . Portanto, esse movimento consiste em realocar uma viagem de um caminhão l para um caminhão k que esteja operando na frente i . Restrições de compatibilidade entre equipamentos são respeitadas neste movimento, havendo realocação de viagens apenas quando houver compatibilidade entre eles.

Movimento Operação Frente - $N^{OF}(s)$. Consiste em retirar de operação o equipamento de carga que esteja em operação na frente i . O movimento retira todas as viagens feitas a esta frente, deixando o equipamento *inativo*. O equipamento retorna à operação assim que uma nova viagem é associada a ele.

Movimento Operação Caminhão - $N^{OC}(s)$. Consiste em selecionar uma célula n_{il} da matriz N e zerar seu conteúdo, significando retirar de atividade um caminhão l que esteja operando em uma frente i .

Movimento Troca de Viagens - $N^{VT}(s)$. Duas células da matriz N são selecionadas e uma viagem é realocada de uma célula para outra. Tal movimento pode ocorrer entre quaisquer células da matriz N , respeitando-se as restrições de compatibilidade entre equipamentos.

Movimento Troca de Carregadeiras - $N^{CT}(s)$. Consiste em trocar duas células distintas y_i e y_k da matriz Y , ou seja, trocar os equipamentos de carga que operam nas frentes i e k . Analogamente ao movimento CG , os equipamentos de carga das frentes são trocados, mas as viagens feitas às frentes não são alteradas. Para manter a compatibilidade entre carregadeiras e caminhões, as viagens feitas a frentes com equipamentos de carga incompatíveis são removidas.

4. Algoritmo proposto

O algoritmo proposto, denominado H-GVILS, consiste em adicionar um módulo de programação matemática ao algoritmo heurístico GVILS de [5] .

No algoritmo GVILS, o qual é baseado em ILS e cujo pseudocódigo está apresentado na Figura 1, uma solução inicial é construída pelo procedimento descrito na Subseção 3.2. A busca local é feita pelo procedimento VND usando-se os movimentos descritos na subseção 3.2. Estrategicamente, a busca local opera nas vizinhanças em uma ordem pré-definida, começando das que exigem menor esforço computacional para aquelas que exigem maior esforço. Assim, o VND segue a seguinte ordem de exploração: N^{CG} , N^{NV} , N^{VC} , N^{VF} . Os movimentos relativos às vizinhanças N^{CT} , N^{OC} , N^{OF} e N^{VT} não foram utilizados na busca local, mas apenas como perturbação. Isto foi feito tendo em vista que estes últimos movimentos requerem esforço computacional muito elevado, inadequado para o tempo computacional disponível para a tomada de decisão .

```
Algoritmo GVILS( Solução  $s$  )
1  $s \leftarrow ConstróiSoluçãoEstéril ()$ 
2  $s \leftarrow ConstróiSoluçãoMinério ( s )$ 
3  $s \leftarrow VND( s )$ 
4 enquanto ( critério de parada não satisfeito ) faça
5    $s' \leftarrow Perturbação ( s, nível )$ 
6    $s'' \leftarrow VND ( s' )$ 
7   se (  $f(s'') < f(s)$  ) então
8      $s \leftarrow s''$ 
9   fim-se
10 fim-enquanto
11 Retorne  $s$ 
fim GVILS
```

Figura 1: Algoritmo GVILS

O objetivo da perturbação é diversificar a busca, gerando uma solução diferente e cada vez mais *distante* da região atual de exploração no espaço de busca. Para cumprir esta missão, são estabelecidos vários níveis de perturbação. Para cada nível n , são aplicados à solução corrente $n + 2$ movimentos, escolhidos aleatoriamente dentre os oito descritos na Subseção 3.2. A essa solução perturbada é aplicada busca local, baseada no procedimento VND (linha 6 da Fig. 1).

Após *IterMax* iterações sem melhora em um dado nível, este é aumentado. No caso de se encontrar uma solução de melhora, o nível de perturbação volta ao seu nível mais baixo.

O algoritmo H-GVILS difere do GVILS em dois aspectos. A partir de um determinado nível de perturbação, a busca local (linha 6 da Figura 1) não passa a ser feita de forma heurística e sim de forma exata. Para tanto, fixa-se uma parte grande do problema e resolve-se na otimalidade uma parte pequena do POLAD. Além disso, há diferença também na maneira como são feitas as perturbações após esse determinado nível de perturbação ser alcançado.

No H-GVILS, para cada nível p de perturbação, são aplicados à solução corrente $p + 2$ movimentos, escolhidos aleatoriamente dentre os oito descritos na Subseção 3.2. A diferença é que após um nível H de perturbações, o método começa a fazer perturbações da seguinte maneira: inicialmente seleciona-se aleatoriamente 10% das frentes e fixa-se no modelo exato todas as configurações das outras frentes não selecionadas, ou seja, as células das matrizes Y , de carregadeiras e N , de viagens, que não pertencem a estes 10% selecionados. Desta maneira, executa-se o otimizador de programação matemática e retorna-se como solução perturbada a solução encontrada pelo método exato. A cada nível são aumentados mais 10% de frentes livres, de forma que ao término de dez níveis, o método é executado totalmente de forma exata. O procedimento de solução via o otimizador é interrompido em duas situações: (i) após encontrar a solução

ótima ou (ii) após decorridos τ segundos de processamento.

5. Resultados Computacionais

O algoritmo H-GVILS proposto foi desenvolvido em C++ usando o framework de otimização OptFrame, disponível em <https://sourceforge.net/projects/optframe> compilado pelo g++ 4.0. O modelo exato foi resolvido pelo CPLEX 9.1 e pelo GLPK 4.9 na versão híbrida do algoritmo. Os testes foram executados em um microcomputador Pentium IV, 3.0 GHZ e 2 GB de RAM, com Windows XP Professional.

Os cenários utilizados para teste, bem como os parâmetros do algoritmo proposto, foram os mesmos de [5].

Cada problema-teste foi executado 10 vezes pelo algoritmo H-GVILS, considerando-se como critério de parada os tempos de 2 e 15 minutos. A seguir, ele foi comparado com o *solver* CPLEX 9.1, bem como o GVILS de [12], em 15 minutos de execução e de [5], em 2 minutos de execução. Não houve comparação com o algoritmo de [1] visto que este não contemplava restrições relativas à taxa de utilização de caminhões e a função de custo era diferente.

As Tabelas 1 e 2 apresentam os resultados médios das funções objetivo obtidas pelos algoritmos H-GVILS, GVILS e pelo otimizador CPLEX em 15 e 2 minutos de processamento, respectivamente.

Tabela 1: Resultados 15 min. de processamento

Problema	CPLEX	GVILS	H-GVILS
-teste	15 min	15 min	15 min
PADC01	227,32	227,22	227,16
PADC02	257,30	254,38	253,45
PADC03	164.040	164.089,28	164.076,04
PADC04	164.054	164.187,48	164.172,49

Tabela 2: Resultados 2 min. de processamento

Problema	CPLEX	GVILS	H-GVILS
-teste	2 min	2 min	2 min
PADC01	7.228,95	227,73	227,73
PADC02	7.276,39	256,72	256,72
PADC03	164.043,55	164.099,06	164.099,06
PADC04	164.085,84	164.245,03	164.245,03

Como pode ser observado pela Tabela 1, o algoritmo H-GVILS é superior a GVILS em todos os problemas-teste. Comparando-o com o CPLEX, verifica-se que ele o superou nos problemas-teste PADC01 e PADC02 em até 1,49%; no entanto, teve desempenho inferior nos outros dois problemas, mas aquém apenas a 0,07% da solução do CPLEX, no máximo. Já na Tabela 2, em que o foco está no tempo computacional reduzido, no caso de dois minutos de processamento, verifica-se que tanto o GVILS quanto o H-GVILS superam, em muito, as soluções do CPLEX nos dois primeiros problemas-teste. Entretanto, tais algoritmos têm desempenho inferior nos demais problemas, mas ficam com soluções piores em 0,1%, no máximo. Observa-se, também, que o algoritmo H-GVILS obteve os mesmos resultados do GVILS, visto que sua busca híbrida é apenas executada nas últimas perturbações do ILS, o que ocorreu após os dois minutos de execução do método.

6. Conclusões

Este trabalho teve seu foco no problema de planejamento operacional de lavra considerando alocação dinâmica de caminhões. Foi proposto um algoritmo heurístico híbrido, que combina uma abordagem de programação matemática com os procedimentos heurísticos GRASP, *Variable Neighborhood Descent* e *Iterated Local Search*, para resolvê-lo. A integração proposta foi validada considerando-se que resultados melhores foram obtidos. O algoritmo heurístico também se mostrou adequado para ser usado como ferramenta de apoio à decisão, uma vez que resultados de qualidade são obtidos rapidamente.

Agradecimentos: Os autores agradecem ao CNPq, processo 474831/2007-8, e à FAPEMIG, processo CEX 2991-06.1/07, pelo apoio a este trabalho.

Referências:

- [1] COSTA, F. P. *Aplicações de técnicas de otimização a problemas de planejamento operacional de lavras em mina a céu aberto*. Dissertação (Mestrado em Engenharia Mineral) — PPGEM/UFOP, Ouro Preto, 2005.
- [2] LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of Metaheuristics*. Boston: Kluwer, 2003.
- [3] FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, p. 109–133, 1995.
- [4] MLADENOVIC, N.; HANSEN, P. A variable neighborhood search. *Computers and Operations Research*, v. 24, p. 1097–1100, 1997.
- [5] COELHO, I. M.; RIBAS, S.; SOUZA, M. J. F. Um algoritmo baseado em grasp, iterated local search para a otimização do planejamento operacional de lavra. In: *Anais do XI Encontro de Modelagem Computacional*. Volta Redonda/RJ: [s.n.], 2008.
- [6] WHITE, J. W.; OLSON, J. P. Computer-based dispatching in mines with concurrent operating objectives. *Mining Engineering*, v. 38, n. 11, p. 1045–1054, 1986.
- [7] RODRIGUES, L. F. *Análise comparativa de metodologias utilizadas no despacho de caminhões em minas a céu aberto*. Dissertação (Mestrado em Engenharia de Produção) — DEP/UFMG, Belo Horizonte, 2006.
- [8] CHANDA, E. K. C.; DAGDELEN, K. Optimal blending of mine production using goal programming and interactive graphics systems. *Int. Journal of Surface Mining, Reclamation and Environment*, v. 9, p. 203–208, 1995.
- [9] ALVARENGA, G. B. *Despacho ótimo de caminhões numa mineração de ferro utilizando algoritmo genético com processamento paralelo*. Dissertação (Mestrado em Engenharia Mineral) — DEE/UFMG, Belo Horizonte, 1997.
- [10] MERSCHMANN, L. H. C. *Desenvolvimento de um sistema de otimização e simulação para análise de cenários de produção em minas a céu aberto*. Dissertação (Mestrado em Engenharia de Produção) — PEP/COPPE/UFRJ, Rio de Janeiro, 2002.
- [11] COSTA, F. P.; SOUZA, M. J. F.; PINTO, L. R. Um modelo de alocação dinâmica de caminhões. *Rev. Brasil Mineral*, v. 231, p. 26–31, 2004.
- [12] COELHO, I. M.; RIBAS, S.; SOUZA, M. J. F. Um algoritmo baseado em grasp, vnd e iterated local search para a resolução do planejamento operacional de lavra. In: *XV Simpósio de Engenharia de Produção*. Bauru/SP: [s.n.], 2008.
- [13] BRESINA, J. L. Heuristic-biased stochastic sampling. In: *Proceedings of the 13th National Conference on Artificial Intelligence*, AAAI Press. Portland: [s.n.], 1996. p. 271–278.