

Um algoritmo baseado na metaheurística LAHC para resolver o Problema de Planejamento Operacional de Lavra em Minas a Céu Aberto

Arthur de A. Silva, Marcone J. F. Souza, Victor L. Guimarães

Departamento de Computação, ICEB, UFOP
35400-000, Ouro Preto, MG

E-mail: arthur.ufop@hotmail.com, marcone@iceb.ufop.br, vluz95@yahoo.com.br.

Resumo: *Este trabalho trata o problema de Planejamento Operacional de Lavra em Minas a Céu Aberto com alocação dinâmica de caminhões (POLAD). Neste problema, há um conjunto de carregadeiras, um conjunto de caminhões, um conjunto de frentes de lavra e uma meta de produção e qualidade requeridos para o material lavrado. O objetivo é determinar a alocação das carregadeiras às frentes de lavra, assim como o número de viagens que cada caminhão fará a cada frente de forma a minimizar o número de caminhões necessários ao processo e os desvios das metas de produção e qualidade. Dada sua complexidade, é proposto um algoritmo heurístico baseado na metaheurística Late Acceptance Hill-Climbing (LAHC). O algoritmo desenvolvido explora o espaço de soluções por meio de cinco movimentos, os quais são aplicados de forma auto-adaptativa, privilegiando aqueles que são mais eficazes na exploração do espaço de busca. O algoritmo proposto foi comparado com dois métodos da literatura e se mostrou competitivo, sendo capaz de gerar novas melhores soluções.*

Palavras-chave: *Planejamento operacional de lavra, LAHC, Minas a Céu Aberto*

1 Introdução

O Brasil é um dos maiores países do mundo no setor de extração de minério, ao lado de países como China, Austrália, Rússia, Ucrânia e Estados Unidos [9]. Segundo [7], carga e transporte correspondem entre 30% a 40% dos custos totais do processo de lavra em minas a céu aberto. Dada a necessidade de competitividade no mercado global, surge a necessidade de reduzir esses custos, sem prejuízo da produção e da lucratividade. Dentre os problemas que aparecem em cenários de empresas mineradoras, um dos mais importantes é o do Planejamento Operacional de Lavra a Céu Aberto com alocação dinâmica de caminhões (POLAD). Nesse problema, dado um conjunto de frentes de lavra, um ponto de descarga de material, um conjunto de caminhões, um conjunto de parâmetros de controle (Ferro, Sílica, etc.) e um britador, tem-se que decidir a alocação desses equipamentos de carga para atender às metas de produção e qualidade de minério, assim como o ritmo de lavra em cada frente, isto é, a quantidade de material em toneladas por hora (t/h) extraída de uma frente.

Em vista da dificuldade de resolver este problema na otimalidade para situações reais, ele é normalmente resolvido por meio de técnicas heurísticas [1, 5, 8].

Neste trabalho propomos resolver o POLAD por meio de um algoritmo baseado na metaheurística *Late Acceptance Hill-Climbing* – LAHC [2]. Entre as vantagens de sua utilização está o fato de a mesma possuir apenas um parâmetro, o que facilita a sua calibração. A LAHC foi escolhida por sua simplicidade e sua capacidade de obter ótimos resultados [3]. Além disso, de nosso conhecimento, ainda não foi aplicada à resolução do problema em questão.

Para testar o algoritmo foram usados cenários reais e os resultados obtidos foram comparados com aqueles produzidos pelo algoritmo GGVNS e por um método exato, apresentados em [8].

O restante deste artigo está organizado como segue. Na seção 2 são apresentados alguns trabalhos relacionados. As seções 3 e 4 descrevem o problema e o algoritmo LAHC. Na seção 5 são apresentados e discutidos os resultados da aplicação do LAHC. A seção 6 conclui o trabalho.

2 Trabalhos relacionados

Em [6] é desenvolvido um sistema, denominado OTISIMIN, para resolver o POLAD. O sistema é dividido em dois módulos, sendo que no primeiro há um modelo de programação linear que otimiza o processo de mistura de minérios, levando em consideração as restrições de qualidade impostas e alocando os equipamentos de carga para as frentes de lavra. No segundo, o sistema aplica a técnica de simulação discreta para verificar se é possível realizar o planejamento obtido pelo modelo de otimização. A função objetivo considerada é a maximização do ritmo de lavra. O autor, no entanto, não considera a minimização do número de caminhões e tampouco a minimização dos desvios das metas de produção e qualidade da mistura requerida.

Já [4] apresenta dois métodos para resolução do problema. O primeiro é um método exato que consiste em uma formulação de programação linear inteira mista e o segundo é uma técnica heurística. Na formulação apresentada, o autor aperfeiçoou os modelos de [6], considerando na função objetivo o atendimento às metas de qualidade e produção de minério. No entanto, ele não considerou a minimização do número de caminhões necessários ao processo. Segundo o autor, fixado um tempo de processamento, o algoritmo heurístico desenvolvido é capaz de gerar soluções de melhor qualidade que aquelas geradas pelo método exato.

Em [5] os autores conseguiram obter bons resultados desenvolvendo um algoritmo genético que tem como objetivo a redução dos custos com transporte de material dentro da mina. Porém, eles não consideraram várias restrições existentes no problema como, por exemplo, compatibilidade entre equipamentos de carga e transporte, blendagem do minério extraído, entre outras.

Em [8] foi desenvolvido um algoritmo heurístico, baseado em GRASP e GVNS para solucionar o problema. São utilizados oito tipos de vizinhanças para explorar o espaço de soluções. O trabalho compara os resultados obtidos pelo algoritmo desenvolvido com os obtidos por uma modelagem matemática também apresentada. Os autores mostram a superioridade do algoritmo heurístico frente ao modelo matemático.

Em [1] é apresentada uma heurística construtiva híbrida para auxiliar o planejamento operacional de curto prazo do processo de lavra em minas a céu aberto. O trabalho aborda a alocação dinâmica dos equipamentos de carga e transporte, levando em consideração o tempo de fila; porém não minimiza o número de caminhões necessários ao processo.

3 Descrição do problema

No POLAD há um conjunto de carregadeiras com diferentes produtividades, um conjunto de caminhões com diferentes capacidades, um conjunto de frentes de lavra (minério ou estéril) e uma meta de produção e qualidade (%Fe, %SiO₂, etc) requeridos para o material lavrado. O material extraído das frentes pelas carregadeiras são transportados pelos caminhões para os pontos de descarga (britador e depósito). O objetivo é determinar a alocação das carregadeiras às frentes de lavra, assim como o número de viagens que cada caminhão fará a cada frente de forma a minimizar o número de caminhões necessários ao processo bem como os desvios das metas de produção e qualidade.

As seguintes restrições devem ser atendidas: 1) O ritmo de lavra (de minério e de estéril), deve estar entre um ritmo mínimo e máximo predefinido; 2) A qualidade do minério extraído deve estar entre um valor mínimo e máximo esperado pelo britador; 3) Cada carregadeira deve ser alocada somente a uma frente de lavra; 4) Cada frente pode ter somente uma carregadeira operando; 5) As carregadeiras devem operar respeitando um ritmo máximo e mínimo predefinido; 6) Caminhões só podem fazer viagens onde há carregadeiras alocadas que sejam compatíveis; 7) Devido aos possíveis imprevistos, como manutenção, chuva, entre outros, cada caminhão pode produzir no máximo $Tx\%$ da sua capacidade total em um período de tempo.

4 Modelo Heurístico

4.1 Representação da solução

Assim como em [8], a solução é representada por uma matriz $R = [Y|N]$, onde Y é uma matriz $|F| \times 1$, e N é uma matriz $|F| \times |T|$. F é o conjunto de frentes e T é o conjunto de caminhões. Cada célula y_i da matriz $Y_{|F| \times 1}$ representa a carregadeira k alocada a frente i . Caso não tenha uma carregadeira alocada para frente i a célula y_i recebe -1.

Na matriz $N_{|F| \times |T|}$ cada célula n_{il} representa a quantidade de viagens que cada caminhão $l \in T$ realizará na frente i . Se o caminhão l não realiza viagens para a frente i , a célula n_{il} recebe o valor 0.

4.2 Estruturas de vizinhança

Para explorar o espaço de busca foram considerados cinco movimentos, cada qual dando origem a uma vizinhança N^v de uma solução s , sendo $v \in \{TF, RF, RC, RV, AV\}$. Esses movimentos, definidos a seguir, são um subconjunto daqueles de [8].

Troca de frentes – $N^{TF}(s)$ Este movimento considera duas frentes de lavra i e k , em que haja uma carregadeira em pelo menos uma delas (ex. $y_i \geq 0$), e troca os equipamentos de carga, bem como as viagens associadas aos caminhões (caso as duas frentes contenham carregadeiras) ou move o equipamento de carga e as viagens associadas dos caminhões de uma frente para outra na qual não haja uma carregadeira alocada ($y_l = -1$).

Remoção de uma viagem recebida por uma frente – $N^{RF}(s)$ Consiste em escolher uma frente i que possua uma carregadeira alocada e que esteja recebendo ao menos uma viagem. Depois é removida uma viagem de algum caminhão l que foi designado para transportar material da frente i ($n_{il} \leftarrow n_{il} - 1$, desde que $n_{il} > 0$).

Remoção de uma viagem de um caminhão – $N^{RC}(s)$ Consiste em escolher um caminhão l que realiza ao menos uma viagem. Depois é removida uma viagem de alguma frente i que esteja recebendo viagens de l ($n_{il} \leftarrow n_{il} - 1$, desde que $n_{il} > 0$).

Remoção das viagens de um caminhão – $N^{RV}(s)$ Consiste em escolher um caminhão l e remover todas as viagens realizadas por ele ($n_{il} = 0 \forall i \in F$).

Adição de uma viagem – $N^{AV}(s)$ Consiste em escolher uma frente de lavra i que possua uma carregadeira ($y_i \neq -1$), e um caminhão l compatível com a carregadeira alocada em i , e adicionar uma viagem desse caminhão a essa frente ($n_{il} \leftarrow n_{il} + 1$).

O conjunto de todas essas vizinhanças N^v de uma solução s define a vizinhança $N(s)$, isto é, $N(s) = \bigcup_v N^v(s)$.

4.3 Avaliação da solução

Os movimentos podem gerar soluções infactíveis. Assim, uma solução s é avaliada por uma função mono-objetivo $f : S \rightarrow \mathbb{R}$, onde S representa o conjunto de todas as soluções possíveis s geradas pelos movimentos desenvolvidos. A função f , definida pela equação (1), e que deve ser minimizada, consiste em duas partes: a primeira corresponde à função objetivo propriamente dita e a segunda, às várias funções de penalização criadas para desestimular a ocorrência de infactibilidade na solução corrente.

$$f(s) = f^{MP}(s) + f^p(s) + \sum_{j \in P} f_j^q(s) + \sum_{l \in T} f_l^u(s) + \sum_{k \in K} f_k^c(s) \quad (1)$$

Na equação (1), f^{MP} representa a função objetivo, calculada pela equação (2).

$$f^{MP}(s) = \sum_{j \in P} \lambda_j^- d_j^- + \sum_{j \in P} \lambda_j^+ d_j^+ + \alpha^- D_o^- + \alpha^+ D_o^+ + \beta^- D_w^- + \beta^+ D_w^+ + \sum_{l \in T} \omega_l U_l \quad (2)$$

Na equação (2), $f^{MP}(s)$ avalia s em relação à qualidade do material, desvios das metas de produção e utilização dos caminhões. P é o conjunto de parâmetros de controle; d_j^- corresponde ao desvio negativo do parâmetro de controle j na mistura final (ton/h); d_j^+ corresponde ao desvio positivo do parâmetro de controle j na mistura final (ton/h); λ_j^- e λ_j^+ correspondem a penalidades por desvios negativo e positivo do controle de parâmetro j , respectivamente; α^- e α^+ correspondem a penalidades por desvios negativo e positivo da produção de minério, respectivamente; D_o^- e D_o^+ correspondem aos desvios negativo e positivo da produção de minério recomendada, respectivamente; β^- e β^+ correspondem a penalidades por desvios negativo e positivo da produção de estéril, respectivamente; D_w^- e D_w^+ correspondem aos desvios negativo e positivo da produção de estéril recomendada, respectivamente; T corresponde ao conjunto de caminhões; ω_l corresponde à penalidade pela utilização do l -ésimo caminhão; U_l recebe 1 se o caminhão l está sendo usado, e 0 caso contrário.

As demais parcelas da equação (1) penalizam as inviabilidades da solução s . A parcela $f^p(s)$ avalia s com relação ao não atendimento aos limites de produção de minério e estéril. A parcela $f_j^q(s)$ avalia o quanto na solução s corrente não se está respeitando os limites do j -ésimo parâmetro de qualidade da mistura. A parcela $f_l^u(s)$, por sua vez, penaliza s com relação à superação do ritmo máximo de produção do l -ésimo caminhão. Por fim, a parcela $f_k^c(s)$ penaliza a solução corrente que não atenda aos limites de produtividade da carregadeira k .

4.4 Solução inicial

Cada carregadeira $k \in K$ é alocada de maneira arbitrária a uma frente de lavra disponível, ou seja, que não esteja com outra carregadeira alocada. Não é atribuída nenhuma viagem aos caminhões; sendo assim, para a maioria das instâncias do problema a solução inicial é uma solução infactível.

4.5 Algoritmo proposto

O algoritmo proposto, nomeado LAHC, é baseado na metaheurística *Late Acceptance Hill-Climbing* [2]. Tal como o nome sugere, essa metaheurística é uma evolução do procedimento heurístico *Hill-Climbing* (HC), diferenciando-se deste na atualização da solução corrente. Ao contrário do HC, que compara o valor da solução candidata com o valor da solução corrente, no LAHC o valor da solução candidata é comparado com o valor de uma solução que era corrente há algumas iterações anteriores. O Algoritmo 1 apresenta seus passos.

O algoritmo recebe uma função $f(\cdot)$, definida pela Equação (1), para avaliar a solução, um conjunto de vizinhanças (N), um tempo máximo para a execução do algoritmo (*tempoLimite*), uma solução inicial s , o tamanho $|L|$ de uma lista $L = \{f'_0, f'_1, \dots, f'_{|L|-1}\}$ que é utilizada no algoritmo para armazenar os valores das $|L|$ últimas soluções correntes e o intervalo de *atprob* iterações para a atualização das probabilidades de escolha de cada vizinhança.

Na linha 2 todos os elementos da lista recebem o valor da função de avaliação aplicada à solução inicial. Na linha 6 é atualizada a melhor solução conhecida, no caso, a solução inicial.

A seguir, o algoritmo entra em um laço de repetição, que é interrompido quando o tempo limite é alcançado. Na linha 9 é escolhida uma vizinhança v com uma probabilidade $prob_v$ e gera-se aleatoriamente uma solução candidata a partir da solução corrente s . O Algoritmo 2 apresenta o procedimento que escolhe uma vizinhança e gera a partir dela uma nova solução.

A linha 11 do Alg. 1 faz com que a lista L funcione de forma circular. Como se trata de um problema de minimização, na linha 12 é verificado se o valor da função f da solução candidata é menor que o valor da solução que era a solução corrente há algumas iterações (f'_{vs}). Caso seja, a solução corrente é atualizada (Linha 13) e a taxa de sucesso da vizinhança v é aumentada em

Algoritmo 1: Algoritmo LAHC proposto

Entrada: $f(\cdot), N, tempoLimite, s, |L|, atprob$
Saída: s^*

```

1  $tempoAtual \leftarrow 0$ ;
2  $f'_k \leftarrow f(s) \quad \forall k \in \{0, \dots, |L| - 1\}$ ;
3  $prob_v \leftarrow 1/|N| \quad \forall v \in N$ ; /* Cria o vetor de probabilidades para cada vizinhança  $v$  */
4  $qtde_v \leftarrow 0 \quad \forall v \in N$ ; /* Quantidade de vezes que a vizinhança  $v$  foi escolhida */
5  $sucesso_v \leftarrow 0 \quad \forall v \in N$ ; /* Quantidade de soluções candidatas geradas pela vizinhança  $v$ , que passaram a
ser a solução corrente */
6  $s^* \leftarrow s$ ; /* Melhor solução encontrada */
7  $iter \leftarrow 0$ ;
8 enquanto  $tempoAtual \leq tempoLimite$  faça
9    $s' \leftarrow geraVizinho(s, v, prob)$ ;
10   $qtde_v \leftarrow qtde_v + 1$ ;
11   $vs \leftarrow iter \bmod |L|$ ;
12  se  $f(s') \leq f'_{vs}$  então
13     $s \leftarrow s'$ ;
14     $sucesso_v \leftarrow sucesso_v + 1$ ;
15    se  $f(s) < f(s^*)$  então
16       $s^* \leftarrow s$ ;
17    fim
18  fim
19  se  $iter \bmod atprob == 0$  então
20     $atualizar(prob, qtde, sucesso)$ ; /* Realiza os cálculos das Equações (3) e (4) */
21     $qtde_v \leftarrow sucesso_v \leftarrow 0 \quad \forall v \in N$ ;
22  fim
23   $f'_{vs} \leftarrow f(s)$ ;
24   $iter \leftarrow iter + 1$ ;
25   $Atualize\ tempoAtual$ ;
26 fim
27 Retorne  $s^*$ ;

```

Algoritmo 2: Procedimento: $geraVizinho$

Entrada: $s, v, prob$
Saída: s', v

```

1  $probabilidade \leftarrow$  valor aleatório no intervalo  $[0, 1[$ ;
2  $soma \leftarrow 0$ ;
3 para todo  $vizinhança\ v' \in N$  faça
4    $soma \leftarrow soma + prob_{v'}$ ;
5   se  $soma \geq probabilidade$  então
6      $v \leftarrow v'$ ;
7     break;
8   fim
9 fim
10  $s' \leftarrow$  aplique o movimento  $v$  à solução  $s$ ;
11 Retorne  $s'$ ;

```

uma unidade (Linha 14). Se após a atualização da solução corrente ela apresentar um valor de f menor que o da melhor solução conhecida (Linha 15), esta é atualizada (Linha 16).

A vizinhança é selecionada de forma adaptativa, isto é, à medida que as soluções vão sendo criadas, a probabilidade associada a cada vizinhança é ajustada. As vizinhanças que produzirem melhores soluções terão maior probabilidade de serem escolhidas. A atualização dessas probabilidades ocorre a cada $atprob$ iterações (Linha 19). O bloco de avaliações deve ser um número suficientemente grande para que um número razoável de soluções candidatas tenham sido avaliadas; neste trabalho utilizamos $atprob = 3000$. O procedimento de atualização das probabilidades (Linha 20) utiliza as equações (3) e (4). Após o cálculo dessas probabilidades o histórico armazenado nas variáveis $qtde$ e $sucesso$ é zerado (Linha 21). Na linha 23 a lista L é atualizada com o valor f da solução corrente.

$$\varphi_v = \frac{sucesso_v + 1}{qtde_v + 1} \quad \forall v \quad (3)$$

$$prob_v = \frac{\varphi_v}{\sum_{t \in \{1, \dots, |N|\}} \varphi_t} \quad \forall v \quad (4)$$

5 Experimentos computacionais e análises

O algoritmo foi codificado em Java, compilado com o eclipse versão 4.0 e executado em um computador intel core i5-2410, 2.3 GHz, com 4 GB de RAM e sistema operacional Windows 7. Para testá-lo foram utilizados os 8 cenários de teste (opm1, ..., opm8) de [8], os quais estão disponibilizados em <http://www.decom.ufop.br/prof/marcone/projects/mining.html>.

Os pesos da função de avaliação, descrita pela equação (2), são os mesmos usados em [8], a saber: $\alpha^+ = \alpha^- = \beta^+ = \beta^- = 100$, $\lambda^- = \lambda^+ = 1 \forall j \in T$, $\omega_l = 1 \forall l \in P$, $Tx_l = 75\% \forall l \in T$.

Dado o caráter estocástico dos algoritmos LAHC e GGVNS, eles foram executados em cada cenário 30 vezes, tendo como tempo limite de execução 2 minutos. Todos os experimentos utilizaram uma lista L de tamanho 10.000, definido empiricamente. Os resultados obtidos para o LAHC foram comparados com os da formulação matemática e do algoritmo GGVNS de [8].

A Tabela 1 apresenta os melhores resultados conhecidos da literatura e aqueles obtidos pelo CPLEX, GGVNS e LAHC em dois minutos de processamento. A coluna “Opt*” é marcada com um asterisco quando o custo encontrado é ótimo. Na coluna “Melhor custo conhecido”, os melhores resultados conhecidos não são necessariamente obtidos em 2 minutos de processamento.

Tabela 1: Melhores resultados alcançados por CPLEX, GGVNS e LAHC

Instância	Melhor custo Conhecido	Opt*	CPLEX	GGVNS	LAHC
			Upper Bound	Melhor	Melhor
opm1	227.12		230.65	230.12	227.66
opm2	256.37		4,858.39	256.37	259.09
opm3	164,027.15	*	164,042.60	164,039.12	164,038.77
opm4	164,056.68	*	164,061.80	164,099.66	164,091.49
opm5	226.35		7,229.07	228.09	226.35
opm6	235.93		236.58	236.58	235.93
opm7	164,017.46	*	164,017.46	164,021.38	164018.76
opm8	164,018.65	*	164,018.65	164,023.73	164021.23

* Considerando o *mi*gap de tolerância do CPLEX $\leq 10^{-5}$, exceto para o opm3, que usou um *mi*gap de tolerância $\leq 10^{-4}$.

Como se observa na Tabela 1, o LAHC encontrou melhores resultados que o GGVNS em sete das oito instâncias. Já comparando-o com o CPLEX, ele encontrou melhores resultados em cinco instâncias. Quando a comparação é com relação às melhores soluções, verifica-se que o LAHC melhorou os resultados conhecidos de duas instâncias (opm5 e opm6), enquanto o GGVNS produz a melhor solução em uma única instância (opm2) e o CPLEX é o melhor em duas instâncias (opm7 e opm8). Nas demais instâncias, os melhores resultados foram obtidos pelo CPLEX rodando em duas horas de processamento.

A Tabela 2 apresenta os resultados médios obtidos pelos algoritmos LAHC e GGVNS, assim como os do modelo exato. A coluna “Gap” foi obtida pela expressão: $Gap_r^m = (f_r^m - f_r^*) / (f_r^*) \forall r \in \{opm1, \dots, opm8\}, \forall m \in \{CPLEX, GGVNS, LAHC\}$, sendo f_i^* o melhor resultado encontrado para a instância r e f_r^m , com $m \in \{LAHC, GGVNS, CPLEX\}$, a média de 30 execuções dos algoritmos LAHC e GGVNS quando $m = LAHC$ ou $m = GGVNS$, respectivamente, e o *upper bound* da instância r quando $m = CPLEX$.

Tabela 2: Resultados médios dos algoritmos GGVNS e LAHC e *upper bound* do CPLEX

Instância	Melhor custo Conhecido	CPLEX		GGVNS		LAHC	
		Upper Bound	Gap	Média	Gap	Média	Gap
opm1	227.12	230.65	1.55	230.12	1.32	229.57	1.07
opm2	256.37	4,858.39	>100.00	256.56	0.07	268.82	4.85
opm3	164,027.15	164,042.60	0.01	164,064.68	0.02	164,063.16	0.02
opm4	164,056.68	164,061.80	0.00	164,153.92	0.06	164,113.67	0.03
opm5	226.35	7,229.07	>100.00	228.09	0.77	227.34	0.43
opm6	235.93	236.58	0.28	237.97	0.86	238.82	1.22
opm7	164,017.46	164,017.46	0.00	164,021.89	0.00	164022.60	0.00
opm8	164,018.65	164,018.65	0.00	164,027.29	0.00	164026.15	0.00

Pela Tabela 2 verifica-se que em termos de resultados médios, o algoritmo LAHC superou o GGVNS em cinco instâncias e o CPLEX em três. Em termos de variabilidade da solução final, o LAHC superou o GGVNS em três instâncias, empatou em três e perdeu em duas. Em relação

ao CPLEX, o LAHC o superou em termos de qualidade da solução final, em três instâncias, empatou em duas e perdeu em três. Destaca-se que nas duas instâncias (opm2 e opm5) em que o LAHC ganhou do CPLEX, a qualidade da solução final do LAHC foi muito superior à do CPLEX, enquanto que naquelas em que o LAHC perdeu, a diferença não foi tão significativa.

6 Conclusão

Este trabalho tratou o problema do planejamento operacional de lavra em minas a céu aberto. Para resolvê-lo foi desenvolvido um algoritmo baseado na metaheurística *Late Acceptance Hill-Climbing* (LAHC). O algoritmo proposto explora o espaço de soluções utilizando cinco tipos diferentes de movimento, os quais são escolhidos de forma autoadaptativa levando em consideração o sucesso da aplicação desses movimentos em iterações progressas.

O algoritmo foi testado em cenários reais e comparado com outros dois métodos da literatura, o algoritmo heurístico GGVNS de [8] e o otimizador CPLEX 12.5 aplicado à formulação matemática desses mesmos autores. O algoritmo proposto foi capaz de melhorar duas das oito instâncias testadas, assim como produzir resultados finais de boa qualidade e com menor *gap* que os outros dois métodos da literatura em termos de soluções médias.

Agradecimentos

Os autores agradecem à Universidade Federal de Ouro Preto, e às agências CAPES, CNPq e FAPEMIG pelo suporte financeiro ao desenvolvimento deste trabalho.

Referências

- [1] M. Amaral e L.R. Pinto, “Planejamento de operações de lavra em minas a céu aberto com alocação de equipamentos de carga e de transporte”, em *Anais do XLII Simpósio Brasileiro de Pesquisa Operacional*, Bento Gonçalves, pp. 1177-1188, Rio de Janeiro, 2010.
- [2] E.K. Burke e Y. Bykov, “A late acceptance strategy in hill-climbing for exam timetabling problems”, in *Proceedings of the PATAT 2008 Conference*, Montreal, Canadá, 2008.
- [3] E.K. Burke e Y. Bykov, “The late acceptance hill-climbing heuristic”, Relatório Técnico CSM-192, University of Stirling, 2012.
- [4] F.P. Costa, “Aplicações de técnicas de otimização a problemas de planejamento operacional de lavra em minas a céu aberto”, Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mineral, Universidade Federal de Ouro Preto, Ouro Preto, 2005.
- [5] M.X. He, J.C. Wei, X.M. Lu e B.X. Huang, “The Genetic Algorithm for Truck Dispatching Problems in Surface Mine”, *Journal of Information Technology*, 9 (2010) 710-714.
- [6] L.H.C Merschmann, “Desenvolvimento de um sistema de otimização e simulação para análise de cenários de produção em minas a céu aberto”. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia de Produção, UFRJ, Rio de Janeiro, 2002.
- [7] Y. Qing-Xia, “Computer simulation of drill-rig/shovel operations in open-pit mines”. *1982 Winter Simulation Conference*, pp. 463-468, 1982.
- [8] M.J.F. Souza, I.M. Coelho, S. Ribas, H.G. Santos e L.H.C. Merschmann, “A hybrid heuristic algorithm for the open-pit-mining operational planning problem”. *European Journal of Operational Research*, 207 (2010) 1041-1051.
- [9] UNCTAD, “Iron ore market 2008–2010”, *TRUST FUND ON IRON ORE INFORMATION*, Genebra. United Nations Conference on Trade and Development, 2009.