UMA ABORDAGEM MULTIOBJETIVO PARA O PROBLEMA DE PLANEJAMENTO OPERACIONAL DE LAVRA

Vitor Nazário Coelho[1], Marcone Jamilson Freitas Souza[1], Igor Machado Coelho[2]
Frederico Gadelha Guimarães[3] e Raphael Carlos Cruz[1]

[1] Universidade Federal de Ouro Preto, Ouro Preto, MG, 35400-000, Brasil

[2]Universidade Federal Fluminense (UFF), Niterói, RJ, 24210-240 Brasil

[3]Federal University of Minas Gerais, Belo Horizonte, MG, 31270-901, Brazil

vncoelho@gmail.com, marcone@iceb.ufop.br, imcoelho@ic.uff.br,

fredericoguimaraes@ufmg.br e phaelcarlos@gmail.com

Abstract. *This work presents two multiobjective heuristic algorithms based on Multiobjetive Variable Neighborhood Search (MOVNS) and Nondominated Sorting Genetic Algorithm II (NSGA-II) procedures. The algorithms were applied to a problem that requires quick decisions, the open-pit-mining operational planning problem with dynamic truck allocation (OPMOP). Approximations to Pareto sets generated by the developed algorithms were compared considering the hypervolume, coverage and spacing metrics. Computational experiments have shown the superiority of the local search algorithm MOVNS, which was able to find better sets of non-dominated solutions, more diversified and with an improved convergence, compared to the population based search algorithm NSGA-II. MOVNS algorithm was also validated as a good tool for mono-objective optimization, since it achieved better solutions than a mono-objective literature algorithm.*

KEYWORDS*: Open-pit-mining, Multiobjective Optimization, MOVNS, NSGA-II*

Resumo. *Este trabalho apresenta dois algoritmos heurísticos multiobjetivos baseados nos procedimentos Multiobjetive Variable Neighborhood Search (MOVNS) e Nondominated Sorting Genetic Algorithm II (NSGA-II). Os algoritmos foram aplicados a um problema que requer decisões rápidas, o problema de planejamento operacional de lavra em minas a céu aberto com alocação dinâmica de caminhões (POLAD). As aproximações das Fronteiras de Pareto geradas pelos algoritmos desenvolvidos foram comparadas entre si tendo em vista as métricas de hipervolume, cobertura e espaçamento. Os experimentos computacionais realizados mostraram a superioridade do algoritmo baseado no procedimento de busca local MOVNS, que foi capaz de encontrar conjuntos de soluções não-dominadas mais diversificados e com uma melhor convergência, quando comparado ao algoritmo de busca populacional NSGA-II. O algoritmo MOVNS também foi validado como uma boa ferramenta de otimização mono-objetivo, visto que ele alcançou melhores soluções que um algoritmo mono-objetivo da literatura.*

PALAVRAS-CHAVE*: Planejamento operacional de lavra, Otimização Multiobjetivo, MOVNS, NSGA-II*

1 Introdução

Este trabalho trata do problema de planejamento operacional de lavra com alocação dinâmica de caminhões (POLAD). Neste problema, deve-se determinar a taxa de extração de minério e estéril nas frentes de lavra e associar a elas carregadeiras e caminhões de forma que as metas de produção e qualidade sejam satisfeitas. Além disso, procura-se minimizar o número de caminhões necessários para a execução desta tarefa. Tradicionalmente, o POLAD tem sido tratado como um problema de otimização mono-objetivo com soma ponderada de três objetivos: a minimização dos desvios de qualidade, a minimização dos desvios de produção e a minimização do número de caminhões necessários ao processo. Neste trabalho propõe-se tratá-lo por uma abordagem multiobjetivo. Desta forma, o que se procura é um conjunto de soluções não-dominadas, também chamadas de soluções eficientes, ou Fronteira de Pareto, cabendo ao tomador de decisões a escolha da solução mais adequada às suas necessidades. As soluções não-dominadas são também chamadas eficientes ou Pareto-ótimo. A Fronteira de Pareto é a imagem das soluções não-dominadas no espaço dos objetivos.

Para a resolução do POLAD, a literatura tem mostrado várias abordagem baseadas em procedimentos heurísticos, visto que métodos exatos possuem uma aplicabilidade restrita (Souza *et al.*, 2010). Dentre esses trabalhos destacamos: Costa (2005), que desenvolveu um algoritmo heurístico combinando *Greedy Randomized Adaptive Search Procedures* – GRASP (Feo e Resende, 1995) com VNS (Hansen e Mladenovic, 2001) e usando seis tipos diferentes de movimentos para explorar o espaço de soluções; Coelho *et al.* (2008), que propuseram o algoritmo GVILS, combinando os procedimentos heurísticos GRASP, VND e ILS (Lourenço *et al.*, 2003), e desenvolveram mais dois tipos de movimentos; Souza *et al.* (2010), que desenvolveram o algoritmo GGVNS, que combina as metaheurísticas *General Variable Neighborhood Search* – GVNS (Hansen *et al.*, 2008) e o procedimento GRASP; Coelho *et al.* (2011c), que apresentaram uma paralelização do algoritmo sequencial de Souza *et al.* (2010) e Coelho *et al.* (2011b), que desenvolveram um algoritmo evolutivo inspirado em Estratégias Evolutivas (Beyer e Schwefel, 2002).

Em termos de abordagem multiobjetivo para POLAD, o único trabalho encontrado na literatura foi o de Pantuza (2011). Esse autor propôs um algoritmo genético multiobjetivo híbrido baseado no procedimento *Nondominated Sorting Genetic Algorithm II* - NSGA-II (Deb *et al.*, 2002). Na abordagem utilizada, foram considerados três objetivos conflitantes: minimizar o número de caminhões necessários para o processo de produção, minimizar os desvios em relação às metas dos teores dos parâmetros de qualidade e minimizar os desvios de produção de minério. Os resultados do modelo de otimização foram validados por meio de um aplicativo de simulação.

No presente trabalho são desenvolvidos dois algoritmos multiobjetivos, sendo um baseado no procedimento *Multiobjective Variable Neighborhood Search* – MOVNS (Geiger, 2004) e outro baseado no NSGA-II. Tais algoritmos têm sido aplicados, com sucesso, na solução de diversos problemas, motivando seu uso na resolução do POLAD. O segundo, inclusive, conforme comentado anteriormente, já foi objeto de estudo em Pantuza (2011) para a solução do POLAD. No presente trabalho, ele é aperfeiçoado. Destaca-se, além disso, que os dois algoritmos são baseados em princípios diferentes, uma vez que o primeiro é baseado em busca local e o segundo, em busca populacional. Tem-se, assim, neste trabalho, também o objetivo de verificar qual dessas propostas melhor se adapta à resolução do POLAD.

O restante deste trabalho está organizado como segue. A Seção 2 detalha os algoritmos propostos para resolver o POLAD. A Seção 3 mostra os resultados dos experimentos computacionais e a Seção 4 conclui o trabalho.

2 Algoritmos desenvolvidos

2.1 Representação de uma solução, vizinhança e avaliação

Dado um conjunto de frentes de lavra F, um conjunto de caminhões V e um conjunto de carregadeiras K, uma solução para o POLAD 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 $k \in K$ alocada à frente $i \in F$. Um 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.

Na Tabela 1, mostra-se um exemplo de uma solução para o POLAD. Nesta Tabela, na primeira coluna indica-se a frente de lavra; na segunda, a carregadeira alocada a essa frente e seu *status*, se em atividade ou não; nas demais colunas, o número de viagens que cada caminhão faz a cada frente de lavra. A ocorrência da letra "X" indica que não há compatibilidade entre o caminhão e a carregadeira alocada à respectiva frente.

Tabela 1. Representação de uma solução

	Carga	Cam_1	Cam_2	...	Cam_V
F_1	$< Car_1, 1 >$	8	X	...	X
F_2	$< D, 0 >$	0	0	...	0
F_3	$< Car_8, 0 >$	0	0	...	0
...	...	...	...	...	...
F_F	$< Car_5, 1 >$	0	9	...	3

Observa-se, pela Tabela 1, que na linha F_1 da coluna *Carga* a dupla $\langle Car_1, 1\rangle$ indica que o equipamento de carga Car_1 está alocado à frente F_1 e em operação. Por outro lado, na linha F_3 dessa mesma coluna, a dupla $\langle Car_8, 0\rangle$ indica que o equipamento de carga Car_8 está alocado à frente F_3, mas não está em operação. Na célula $\langle D, 0\rangle$ mostra-se que não existe equipamento de carga alocado à frente F_2 e que, portanto, esta frente está disponível. O número 8 que aparece na terceira coluna e segunda linha indica que o caminhão Cam_1 fará oito viagens à frente F_1.

Para explorar o espaço de soluções foram utilizados os oito movimentos descritos em Souza *et al.* (2010), cada qual definindo uma estrutura de vizinhança. Sejam $i, j \in F$; $l, m \in V$: a vizinhança $N^{NV}(s)$ consiste no incremento ou decremento em n_{il}; $N^{CG}(s)$ consiste na troca de y_i com y_j, trocando-se também as linhas i e j de N; $N^{VC}(s)$ consiste em transferir uma viagem de n_{il} para n_{jl}; $N^{VF}(s)$ consiste em transferir uma viagem de n_{il} para n_{im}; $N^{OF}(s)$ consiste em zerar uma linha i de N; $N^{OC}(s)$ consiste em zerar a célula n_{il}; $N^{VT}(s)$ consiste em transferir uma viagem de n_{il} para n_{jm}; $N^{CT}(s)$ consiste na troca de y_i com y_j, porém, neste caso as linhas i e j de N não são trocadas e quaisquer incompatibilidades geradas são resolvidas zerando a célula de N incompatível.

Uma solução s é avaliada com relação a três objetivos conflitantes, $z_1(s)$, $z_2(s)$ e $z_3(s)$, que mensuram, respectivamente, os desvios dos parâmetros de qualidade da mistura, os desvios de produção e a quantidade de caminhões usados. Como na exploração do espaço de soluções permite-se gerar soluções inviáveis, então, a esses objetivos também estão associadas as parcelas de inviabilidade $I_1(s)$, $I_2(s)$ e $I_3(s)$, respectivamente. Tais funções objetivo e seus respectivos parâmetros são definidos em Souza *et al*. (2010).

Deste modo, o vetor z de objetivos a ser minimizado é dado pela Eq. (1):

$$z(s) = (z_1(s) + I_1(s), z_2(s) + I_2(s), z_3(s) + I_3(s)) \tag{1}$$

2.2 Algoritmos propostos

São propostos dois algoritmos multiobjetivos, denominados GMOVNS e GNSGAII-PR. O primeiro consiste na combinação dos procedimentos heurísticos *Greedy Randomized Adaptative Search Procedure* – GRASP e *Multiobjective Variable Neighborhood Search* – MOVNS (Geiger, 2004). O segundo combina os procedimentos GRASP, NSGA-II e Reconexão por Caminhos (PR) – *Path Relinking* (Glover, 1996).

O Algoritmo 1 mostra o pseudocódigo do GMOVNS.

Uma solução inicial (linhas 2 e 4 do Algoritmo 1) é gerada utilizando a mesma estratégia de Souza *et al*. (2010), baseado no procedimento parcialmente guloso GRASP. O procedimento `addSolution` (linha 5 do Algoritmo 1), detalhado no Algoritmo 2, adiciona as soluções criadas pelo GRASP na população X_e. As linhas 9 e 10 selecionam um indivíduo da população de soluções potencialmente eficientes e marca este indivíduo como "visitado". Quando todos os indivíduos estão com este marcador, a linha 33 retira tais marcadores.

As variáveis *shaking* e *level*, linha 7 do Algoritmo 1, regulam a perturbação utilizada no algoritmo. Essa versão do algoritmo MOVNS, proposta neste trabalho, possui um mecanismo que regula o nível de perturbação do algoritmo, ou seja, a variável *shaking* é incrementada quando o algoritmo passa um determinado tempo sem obter boas soluções. Da linha 12 até 15 do Algoritmo 1, ocorre o laço de perturbação do algoritmo. No mecanismo utilizado, quanto maior o valor da variável *shaking*, mais a solução será perturbada. Para cada unidade dessa variável seleciona-se, aleatoriamente, uma vizinhança entre as seis seguintes: $N^{NV}(s)$, $N^{CG}(s)$, $N^{VC}(s)$, $N^{VF}(s)$, $N^{VT}(s)$ e $N^{CT}(s)$. Em seguida, aplica-se um movimento de perturbação. A linha 30 retorna os valores das variáveis *level* e *shaking* para uma unidade quando, pelo menos, uma solução é adicionada ao conjunto de soluções potencialmente eficientes X_e.

Por fim, o procedimento `addSolution` (linhas 5 e 19 do Algoritmo 1) está detalhado no Algoritmo 2.

O pseudocódigo do GNSGAII-PR está esquematizado no Algoritmo 3.

Analogamente ao Algoritmo 1, a população inicial P_0 do Algoritmo 3 é iniciada com a adição de indivíduos gerados pelo procedimento GRASP (linha 5 do Algoritmo 3); porém, neste caso, o critério de parada do GRASP é o tamanho da população P_0 (linha 2 do Algoritmo 3).

Algoritmo 1: GMOVNS

Entrada: Vizinhanças $N_k(x)$; *graspMax*; *levelsMax*
Saída: Aproximação de um conjunto eficiente X_e

para $i \leftarrow 1$ **até** $graspMax$ **faça**
    $s_w \leftarrow$ ConstróiSoluçãoEstéril()
    Gere um número aleatório $\gamma \in [0, 1]$
    $s_i \leftarrow$ ConstróiSoluçãoMinério(s_w, γ)
    addSolution($X_e, s_i, f(s_i)$)
fim
$level \leftarrow 1$; $shaking \leftarrow 1$
enquanto *Critério de parado não satisfeito* **faça**
    Seleciona uma solução não visitada $s \in X_e$
    Marque s como *visitada*
    $s' \leftarrow s$
    para $i \leftarrow 1$ **até** $shaking$ **faça**
        Selecione aleatoriamente uma vizinhança $N_k(.)$
        $s' \leftarrow$ Perturbação(s', k)
    fim
    $k_{ult} \leftarrow k$
    incrementa $\leftarrow$ *verdadeiro*
    para todo $s'' \in N_{k_{ult}}(s')$ **faça**
        addSolution($X_e, s'', f(s''), Added$)
        se *Added* = *verdadeiro* **então**
            incrementa $\leftarrow$ *falso*;
        fim
    fim
    se *incrementa* $=$ *verdadeiro* **então**
        level $\leftarrow$ *level* $+ 1$
    senão
        level $\leftarrow 1$; *shaking* $\leftarrow 1$
    fim
    se *level* $\geq$ *levelMax* **então**
        shaking $\leftarrow$ *shaking* $+ 1$; *level* $\leftarrow 1$
    fim
    se *todas* $s \in X_e$ *estão marcadas como visitadas* **então**
        Marque todos $s \in X_e$ como *não-visitado*
    fim
fim
retorna X_e

Algoritmo 2: addSolution

Entrada: População X_e potencialmente eficiente; Solução s e sua avaliação $z(s)$
Saída: X_e; Added (opcional)

Added $\leftarrow$ verdadeiro
para todo $x \in X_e$ **faça**
    se $z(x) \preceq z(s)$ **então**
        Added $\leftarrow$ falso
        Break
    fim
    se $z(s) \prec z(x)$ **então**
        $X_e \leftarrow X_e \setminus x$
    fim
fim
se *Added* $=$ *verdadeiro* **então**
    $X_e \leftarrow X_e \cup s$
fim
retorna X_e

Algoritmo 3: GNSGAII-PR

Entrada: Tamanho da população N; Vizinhanças $N_k(x)$
Saída: Aproximação de um conjunto eficiente X_e

1 População inicial P_0
2 **enquanto** $|P_0| < N$ **faça**
3     $s_w \leftarrow$ ConstróiSoluçãoEstéril()
4   Gere um número aleatório $\gamma \in [0, 1]$
5     $s_i \leftarrow$ ConstróiSoluçãoMinério(s_w, γ)
6   addSolution($P_0, s_i, f(s_i)$)
7 **fim**
8 $Q_0 \leftarrow$ SelecaoCruzamentoPRMutacao(P_0, Vizinhanças $N^{(k)}(.)$)
9 $X_e \leftarrow$ *Nondominated Sorting Genetic Algorithm II*(P_0, Q_0, N, SelecaoCruzamentoPRMutacao(...))
10 **retorna** X_e

Na linha 8 do Algoritmo 3 é acionado o procedimento de seleção, cruzamento e mutação, `SelecaoCruzamentoPRMutacao`, cujo pseudocódigo é mostrado no Algoritmo 4. Neste trabalho, a Reconexão por Caminhos foi utilizada como um operador avançado de cruzamento, tal como em Ribeiro e Resende (2012).

O procedimento *Nondominated Sorting Genetic Algorithm II* (Deb *et al.*, 2002) é ativado na linha 9 do Algoritmo 3. As etapas de "seleção, cruzamento e mutação" são substituídas pelo procedimento `SelecaoCruzamentoPRMutacao`.

Algoritmo 4: SelecaoCruzamentoPRMutacao

Entrada: População P de pais; Vizinhanças $N^{(k)}(.)$; Parâmetros da mutação *mutationRate* e *LocalSearchRate*
Saída: População Q de *offprings*

1 **enquanto** $|Q| < N$ **faça**
2   Selecione dois indivíduos aleatórios s_1 e $s_2 \in P$
3     $s \leftarrow$ melhor indivíduo (s_1, s_2, ReconexãoPorCaminhos(s_1, s_2), ReconexãoPorCaminhos(s_2, s_1))
4   addSolution($Q, s, f(s)$)
5   Gere um número aleatório $ap_{mutation} \in [0, 1]$
6   **se** $ap_{mutation} < mutationRate$ **então**
7   Selecione aleatoriamente uma vizinhança $N^{(k)}(.)$
8         $s' \leftarrow N^{(k)}(s)$
9   **senão**
10         $s' \leftarrow s$
11   **fim**
12   Gere um número aleatório $ap_{LocalSearch} \in [0, 1]$
13   **se** $ap_{localSearch} < LocalSearchRate$ **então**
14         $s'' \leftarrow$ VND(s')
15   addSolution($Q, s'', f(s'')$)
16   **senão**
17   addSolution($Q, s', f(s')$)
18   **fim**
19 **fim**
20 **retorna** Q

O Algoritmo 4 realiza a aplicação dos operadores genéticos, ou seja, dada uma população de pais P, esse procedimento faz a seleção, cruzamento e mutação, de forma a obter uma população Q de *offprings*.

Na linha 3 do Algoritmo 4 o indivíduo s recebe o melhor indivíduo (considerando a função de avaliação mono-objetivo de Souza *et al.* (2010)) entre os dois indivíduos selecionados aleatoriamente, s_1 e s_2, ou um dos dois indivíduos retornados pelo procedimento de Reconexão por Caminhos, ou seja, a Reconexão é acionada tanto com o indivíduo s_1 sendo a solução base quanto sendo a solução guia. Na Reconexão considera-se como atributo a posição que uma carregadeira ocupa em uma solução. A cada passo do procedimento, verificamos se a carregadeira k alocada na frente de lavra i da solução guia é igual à carregadeira alocada na frente i da solução base. Caso a carregadeira seja diferente, move-se a carregadeira k da solução base para a frente i, sendo mantidas as viagens que eram associadas a esta carregadeira. Desta forma, são mantidos os critérios de compatibilidade entre as carregadeiras e os caminhões. Então, realiza-se uma busca local baseada no procedimento VND (Mladenovic e Hansen, 1997), utilizando apenas as estruturas que modificam o número de viagens dos caminhões, no caso, as vizinhanças: $N^{NV}(s)$, $N^{VC}(s)$ e $N^{VF}(s)$, preservando assim as carregadeiras já fixadas. Para prosseguir para o passo seguinte, listamos todos estes movimentos e aplicamos aquele que possui melhor resultado considerando a função de avaliação mono-objetivo. Este procedimento é repetido até que todos os atributos sejam fixados, ou seja, que a carregadeira alocada a cada frente da solução base seja a mesma da solução guia.

Já a linha 8 do Algoritmo 4 aplica uma mutação no indivíduo s caso a variável aleatória $ap_{mutation}$ seja menor que a taxa de mutação $mutationRate$. O mesmo acontece na linha 14, em que aplica-se o procedimento VND caso uma condição análoga seja satisfeita. Finalmente, as linhas 4, 15 e 17 do Algoritmo 4 verificam se os indivíduos s, s' e s'' devem ser adicionadas à população de *offprings* Q.

3 Resultados Computacionais

Os algoritmos propostos *GNSGAII-PR* e *GMOVNS* foram implementados em C++ com auxílio do *framework* OptFrame 1.5 [1] (Coelho *et al.*, 2011a, 2010).

Os experimentos foram realizados em um microcomputador DELL XPS 8300 Intel Core i7-2600, 8MB Cache, 3.4GHz, 16GB RAM, sob sistema operacional Ubuntu 10.10. Para testar os algoritmos desenvolvidos, foi usado o conjunto de 8 problemas-teste de Souza *et al.* (2010).

Primeiramente, foi realizada uma bateria de testes com o objetivo de validar o algoritmo GNSGAII-PR. Três variantes desse algoritmo foram testadas, sendo que cada uma delas corresponde a diferentes tamanhos da população. Para a primeira variante, denominada GNSGAII-PR-20, o tamanho da população foi fixado em 20. Para a segunda, denominada GNSGAII-PR-35, o tamanho da população foi fixado em 35. Já a terceira variante, denominada GNSGAII-PR-65, o tamanho da população foi fixado em 65 indivíduos. Todos os oito problemas-teste foram executados 30 vezes pelos algoritmos, com um tempo computacional limitado a 120 segundos, tal como em Souza *et al.* (2010).

As Tabelas 2 e 3 apresentam os resultados obtidos pelos algoritmos GNSGAII-PR-20, GNSGAII-PR-35 e GNSGAII-PR-65 em relação à função de avaliação dada pela Equação (1), à página 4. Nesta primeira etapa de análise, apenas as métricas de Espaçamento (Schott, 1995) e Hipervolume (Zitzler e Thiele, 1998) foram utilizadas. Nessas tabelas, a coluna "Instância" indica o problema-teste utilizado. A coluna "Melhor" indica o melhor valor da métrica analisada obtido nas 30 execuções. As colunas "Média" e "Desv.Padrão" indicam a média e desvio-padrão da amostra, respectivamente.

[1] Disponível em http://sourceforge.net/projects/optframe/

A Tabela 6 apresenta a média das cardinalidades das frentes de pareto obtidas pelas variantes e pelo algoritmo GMOVNS, também desenvolvido neste trabalho. Desta forma, os valores indicados nesta tabela são a soma do número de soluções não-dominadas obtidas em cada execução dividido pelo número de execuções.

Na Tabela 2 são apresentados os valores da métrica de hipervolume para as três variantes do algoritmo GNSGAII-PR. Analisando-a, percebe-se que para as instâncias opm1, opm2, opm3, opm5 e opm8 a variante GNSGAII-PR-20 foi a que obteve o maior número de melhores valores para essa métrica; porém, a variante GNSGAII-PR-35 apresentou o maior número de menores valores médios e os menores desvios padrão. A variante GNSGAII-PR-65 conseguiu obter a melhor média e o melhor valor para a instância opm6.

A Tabela 3 mostra os resultados da comparação entre as variantes utilizando a métrica de Espaçamento. Novamente, a variante GNSGAII-PR-35 obteve os melhores resultados na média e os menores desvios padrão. A variante GNSGAII-PR-65 obteve os melhores valores de espaçamento, como pode ser visto nas instâncias opm1, opm3, opm4, opm5, opm6 e opm8. Todavia, analisando a Tabela 6, percebemos que a variante GNSGAII-PR-65 obteve frentes de pareto com poucos indivíduos quando comparada com outros algoritmos desenvolvidos. Este resultado forçou uma análise mais aprofundada nas frentes de pareto geradas por essa variante. Desta forma, foi observado que as frentes de pareto onde o espaçamento obtido foi igual a 0, eram frentes compostas por apenas dois indivíduos.

Para a segunda análise de resultados, o algoritmo GMOVNS foi comparado com a variante do algoritmo GNSGAII-PR que obteve o melhor desempenho em termos de convergência e diversidade, ou seja, a variante GNSGAII-PR-35. As Tabelas 4 e 5 indicam os valores obtidos pelos algoritmos utilizando as métricas de Hipervolume, Espaçamento e Cobertura (Zitzler e Thiele, 1998).

Analisando-se as Tabelas 4 e 5 percebe-se que o algoritmo GMOVNS obteve melhores resultados em todas as instâncias. Observando a Tabela 4, nota-se que o algoritmo GMOVNS foi capaz de obter um melhor convergência e diversidade, obtendo as melhores médias para as métricas de hipervolume e espaçamento. A Tabela 5 indica a hegemonia desse algoritmo em relação ao algoritmo GNSGAII-PR, apontando execuções com valores de cobertura iguais a 1 e médias acima ou iguais a 0,66.

A última bateria de testes buscou verificar se esta nova proposta multiobjetivo conseguiria uma boa solução mono-objetivo. A Tabela 7 mostra a comparação dos resultados dos valores das melhores soluções mono-objetivo obtidas pelo algoritmo GMOVNS com aqueles gerados pelo algoritmo GGVNS de Souza *et al.* (2010).

Analisando-se a Tabela 7 percebemos que o algoritmo GMOVNS mostrou-se competitivo com o algoritmo mono-objetivo GGVNS, obtendo melhores soluções que o algoritmo GGVNS em quatro instâncias.

4 Conclusões

Este trabalho teve seu foco no problema de planejamento operacional de lavra considerando alocação dinâmica de caminhões (POLAD). Ele foi resolvido por meio de uma abordagem multiobjetivo, levando-se em consideração três objetivos conflitantes: minimização dos desvios de metas de produção e qualidade para o produto formado, e minimização do número de veículos necessários ao processo produtivo.

Tabela 2. GNSGAII-PR-20 × GNSGAII-PR-35 × GNSGAII-PR-65: Hipervolume

Instância	Tempo	GNSGAII-PR-20 (10^6)			GNSGAII-PR-35 (10^6)			GNSGAII-PR-65 (10^6)		
	(min)	Melhor	Média	Desv. Padrão	Melhor	Média	Desv. Padrão	Melhor	Média	Desv. Padrão
opm1	2	**162,47**	129,87	12,18	155,56	**137,87**	11,10	151,93	130,76	8,24
opm2	2	**99,86**	78,85	8,03	96,29	85,47	6,01	95,65	**86,28**	4,63
opm3	2	**3476,42**	**2924,38**	352,35	3353,66	2887,70	260,69	2433,28	1729,77	267,98
opm4	2	3380,31	2636,62	438,54	**3389,25**	**2650,98**	383,86	2284,42	1814,17	219,71
opm5	2	**88,50**	70,25	7,27	81,91	**74,30**	5,15	94,47	73,65	6,29
opm6	2	111,11	94,36	8,42	114,90	99,51	7,31	**121,88**	**100,06**	8,15
opm7	2	837,15	741,27	54,04	**907,52**	**747,81**	113,07	648,16	518,37	79,11
opm8	2	**1219,13**	1063,71	106,92	1192,36	**1070,22**	95,20	898,82	729,66	106,29

Tabela 3. GNSGAII-PR-20 × GNSGAII-PR-35 × GNSGAII-PR-65: Espaçamento

Instância	Tempo	GNSGAII-PR-20			GNSGAII-PR-35			GNSGAII-PR-65		
	(min)	Melhor	Média	Desv. Padrão	Melhor	Média	Desv. Padrão	Melhor	Média	Desv. Padrão
opm1	2	2510,81	11761,60	8210,30	2589,54	**6924,40**	5007,20	**1659,79**	7731,50	5872,20
opm2	2	2172,80	8190,80	6106,20	**1812,14**	6557,20	5030,70	2081,74	**5198,40**	2674,60
opm3	2	10626,50	16365,40	3736,40	4850,50	**14368,60**	5173,60	**0,00**	16547,90	12236,10
opm4	2	11550,50	**18200,80**	7931,00	4865,89	18295,30	7046,60	**0,00**	19781,90	28634,90
opm5	2	2751,91	11210,10	7047,00	2578,39	**7544,50**	5529,40	**1906,21**	8124,10	6340,20
opm6	2	2080,01	9494,40	8071,50	2218,05	5963,80	4650,80	**1878,61**	**5746,60**	5326,50
opm7	2	9395,22	18402,70	5493,70	**4924,61**	**14673,60**	4646,00	4995,21	16309,80	7041,80
opm8	2	9880,23	19584,60	5308,20	4974,85	**14874,70**	5897,70	**0,00**	16017,40	8068,00

Tabela 4. GMOVNS × GNSGAII-PR-35: Hipervolume e Espaçamento

Instância	Espaçamento				Hipervolume (10^6)			
	Melhor		Média		Melhor		Média	
	GMOVNS	GNSGAII-PR-35	GMOVNS	GNSGAII-PR-35	GMOVNS	GNSGAII-PR-35	GMOVNS	GNSGAII-PR-35
opm1	**1662,54**	2589,54	**2868,68**	6924,42	**209,21**	155,56	**181,86**	137,87
opm2	**1066,13**	1812,14	**2041,59**	6557,20	**139,74**	96,29	**120,88**	85,47
opm3	**2250,96**	4850,50	**6188,99**	14368,62	**3919,54**	3353,66	**3515,59**	2887,70
opm4	**2411,89**	4865,89	**8232,88**	18295,27	**4075,53**	3389,25	**3522,17**	2650,98
opm5	**1400,84**	2578,39	**2563,72**	7544,48	**108,08**	81,91	**97,33**	74,30
opm6	**1087,95**	2218,05	**2025,84**	5963,77	**158,11**	114,90	**140,95**	99,51
opm7	**2917,66**	4924,61	**8311,12**	14673,57	**1003,01**	907,52	**860,02**	747,81
opm8	**2258,14**	4974,85	**8045,67**	14874,73	**1374,06**	1192,36	**1148,71**	1070,22

Tabela 5. GMOVNS × GNSGAII-PR-35: Cobertura

Instância	Tempo (min)	Cobertura					
		C(GMOVNS,GNSGAII-PR-35)			C(GNSGAII-PR-35,GMOVNS)		
		Melhor	Média	Desv. Padrão	Melhor	Média	Desv. Padrão
opm1	2	**1,00**	**0,90**	0,12	0,92	0,04	0,17
opm2	2	**1,00**	**0,86**	0,09	0,10	0,01	0,02
opm3	2	**1,00**	**0,75**	0,19	0,21	0,04	0,07
opm4	2	**1,00**	**0,80**	0,15	0,12	0,02	0,04
opm5	2	**1,00**	**0,88**	0,11	0,070	0,01	0,02
opm6	2	**1,00**	**0,80**	0,16	0,08	0,02	0,03
opm7	2	**1,00**	**0,66**	0,25	0,40	0,08	0,11
opm8	2	**1,00**	**0,66**	0,19	0,79	0,09	0,16

Tabela 6. Média do número de soluções não-dominadas obtidas

Instância	GMOVNS	GNSGAII-PR-20	GNSGAII-PR-35	GNSGAII-PR-65
opm1	39,93	8,10	12,00	11,17
opm2	64,23	8,14	13,04	13,63
opm3	26,20	11,34	13,8	5,97
opm4	24,40	9,94	11,47	5,4
opm5	41,27	8,37	11,00	11,37
opm6	61,43	8,47	12,47	13,30
opm7	26,33	11,34	13,94	7,50
opm8	23,63	11,04	13,9	7,90

Tabela 7. Comparação de resultados: GMOVNS × GGVNS

Algorithm	opm1	opm2	opm3	opm4	opm5	opm6	opm7	opm8
	Melhor	Melhor	Melhor	Melhor	Melhor	Melhor	Melhor	Melhor
GMOVNS	**228,12**	257,75	164051,45	164111,76	**227,16**	239,70	**164019,95**	**164022,17**
GGVNS	230,12	**256,37**	**164039,12**	**164099,66**	228,09	**236,58**	164021,28	164023,73

Na abordagem multiobjetivo não há uma única solução que satisfaça a todos os objetivos. O que se procura é um conjunto de soluções não-dominadas, também chamadas de soluções eficientes, ou Fronteira de Pareto, cabendo ao tomador de decisões a escolha da solução mais adequada.

Em virtude da complexidade combinatória do problema, foram propostos dois algoritmos heurísticos multiobjetivos. O primeiro deles, denominado GNSGAII-PR, combina os procedimentos *Greedy Randomized Adaptive Search Procedures* (GRASP), *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) e a Reconexão por Caminhões como operador de cruzamento. O segundo algoritmo, denominado GMOVNS, combina os procedimentos GRASP e *Multiobjective Variable Neighborhood Search* (MOVNS).

Para comparar o desempenho desses algoritmos, foram usados oito problemas-teste da literatura e três métricas de avaliação: Hipervolume, Cobertura e Espaçamento. Os experimentos computacionais mostraram que o algoritmo GMOVNS teve desempenho melhor que o GNSGAII-PR, uma vez que ele obteve, em todos os problemas-teste utilizados, frentes de pareto mais diversificadas e com uma melhor convergência. Utilizando uma função mono-objetivo, as soluções obtidas pelo algoritmo GMOVNS também foram comparadas a um algoritmo de otimização mono-objetivo da literatura, denominado GGVNS, de Souza *et al*. (2010). O algoritmo GMOVNS foi capaz de obter melhores soluções em quatro problemas-teste, mostrando, assim, o poderio deste algoritmo tanto para aplicações multiobjetivos quanto para aplicações mono-objetivo.

Dado que a tomada de decisão no problema em pauta tem que ser rápida, os resultados encontrados validam a utilização dos algoritmos propostos enquanto ferramenta de apoio à decisão.

Agradecimentos

Os autores agradecem à FAPEMIG e ao CNPq pelo apoio ao presente trabalho.

Referências

Beyer, H. G. e Schwefel, H. P. (2002). Evolution strategies - a comprehensive introduction. *Natural Computing*, v. 1, p. 3–52.

Coelho, I. M.; Munhoz, P. L. A.; Haddad, M. N.; Coelho, V. N.; Silva, M. M.; Souza, M. J. F. e Ochi, L. S. (2011)a. A computational framework for combinatorial optimization problems. *VII ALIO/EURO Workshop on Applied Combinatorial Optimization*, p. 51–54, Porto.

Coelho, I. M.; Ribas, S.; Perche, M. H. P.; Munhoz, P. L. A.; Souza, M.J. F. e Ochi, L. S. (2010). Optframe: a computational framework for combinatorial optimization problems. *XLII Simpósio Brasileiro de Pesquisa Operacional*, p. 1887 – 1898, Bento Gonçalves, RS.

Coelho, I. M.; Ribas, S. e Souza, M. J. F. (2008). Um algoritmo baseado em grasp, vnd e iterated local search para a resolução do planejamento operacional de lavra. *XV Simpósio de Engenharia de Produção*, Bauru/SP.

Coelho, V. N.; Souza, M. J. F.; Coelho, I. M.; Guimarães, F. G. e Coelho, B. N. (2011)b. Estratégias evolutivas aplicadas a um problema de programação inteira mista. *Anais X Congresso Brasileiro de Inteligência Computaciona (CBIC)*, volume 1, p. 1–8, Fortaleza/CE.

Coelho, V. N.; Souza, M. J. F.; Coelho, I. M.; Ribas, S. e Oliveira, T. A. (2011)c. PGGVNS: Um algoritmo paralelo para o problema de planejamento operacional de lavra. *Anais XVIII Simpósio de Engenharia de Produção (SIMPEP)*, volume 1, p. 1–14, Bauru/SP.

Costa, F. P. (2005). Aplicações de técnicas de otimização a problemas de planejamento operacional de lavras em mina a céu aberto. Dissertação, Programa de Pós-Graduação em Engenharia Mineral, Escola de Minas, UFOP, Ouro Preto.

Deb, K.; Pratap, A.; Agarwal, S. e Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, v. 6, n. 2, p. 182–197.

Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, p. 109–133.

Geiger, Martin Josef. (2004). Randomised variable neighbourhood search for multi objective optimisation. *In Proceedings of the 4th EUME Workshop Design and Evaluation of Advanced Hybrid MetaHeuristics*, v. November 4, n. Nottingham, United Kingdom,, p. 34–42.

Glover, F. (1996). Tabu Search and adaptive memory programming - advances, applications and challenges. Barr, R.; Helgason, R. e Kennington, J., editors, *Interfaces in Computer Sciences and Operations Research*, p. 1–75. Kluwer Academic Publishers.

Hansen, P. e Mladenovic, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, v. 130, p. 449–467.

Hansen, P.; Mladenovic, N. e Pérez, J. A. M. (2008). Variable neighborhood search: methods and applications. *4OR: Quarterly journal of the Belgian, French and Italian operations research societies*, v. 6, p. 319–360.

Lourenço, H. R.; Martin, O. C. e Stützle, T. (2003). Iterated local search. Glover, F. e Kochenberger, G., editors, *Handbook of Metaheuristics*. Kluwer Academic Publishers, Boston.

Mladenovic, N. e Hansen, P. (1997). A variable neighborhood search. *Computers and Operations Research*, v. 24, p. 1097–1100.

Pantuza, G. (2011). Métodos de otimização multiobjetivo e de simulação aplicados ao problema de planejamento operacional de lavra em minas a céu aberto. Dissertação de mestrado, Programa de Pós-Graduação em Engenharia Mineral - PPGEM da Universidade Federal de Ouro Preto, Ouro Preto.

Ribeiro, C. e Resende, M. (2012). Path-relinking intensification methods for stochastic local search algorithms. *Journal of Heuristics*, v. 18, p. 193–214.

Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. Dissertação de mestrado, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Souza, M. J. F.; Coelho, I. M.; Ribas, S.; Santos, H. G. e Merschmann, L. H. C. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research, EJOR*, v. 207, p. 1041–1051.

Zitzler, E. e Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms - a comparative case study. Eiben, Agoston; Back, Thomas; Schoenauer, Marc e Schwefel, Hans-Paul, editors, *Parallel Problem Solving from Nature - PPSN V*, volume 1498 of *Lecture Notes in Computer Science*, p. 292–301. Springer Berlin / Heidelberg.