ALGORITMOS MULTIOBJETIVOS 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], Thibaut Lust[1]

[1]Departamento de Computação - Universidade Federal de Ouro Preto (UFOP)
vncoelho@gmail.com, marcone@iceb.ufop.br
[2]Departamento de Engenharia Elétrica - Universidade Federal de Minas Gerais (UFMG)
frederico.g.guimaraes@gmail.com
[3]Instituto de Computação - Universidade Federal Fluminense (UFF)
igor.machado@gmail.com
[4]LIP6-CNRS, UPMC. 4 Place Jussieu 75252 Paris Cedex 05, França
lust.thibaut@gmail.com

Resumo

Este trabalho apresenta três algoritmos heurísticos multiobjetivos baseados nos procedimentos *Multiobjetive Variable Neighborhood Search* (MOVNS), *Two-phase Pareto Local Search* com VNS (2PPLs) e *Nondominated Sorting Genetic Algorithm* II (NSGA-II). Os algoritmos foram aplicados ao 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, espaçamento, cobertura e cardinalidade. Os experimentos computacionais realizados mostraram a superioridade do algoritmo baseado no método 2PPLs, que foi capaz de encontrar conjuntos de soluções não-dominadas mais diversificados e com uma melhor convergência. O algoritmo 2PPLS 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-Chaves: Planejamento Operacional de Lavra, Otimização Multiobjetivo, MOVNS, Two-phase Pareto local search, NSGA-II

Abstract

This work presents three multiobjective heuristic algorithms based on Multiobjetive Variable Neighborhood Search (MOVNS), Two-phase Pareto Local Search with VNS (2PPLs) and Nondominated Sorting Genetic Algorithm II (NSGA-II) procedures. The algorithms were applied to 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, spacing and cardinality metrics. Computational experiments have shown the superiority of the algorithm based on the method 2PPLs, which was able to find better sets of non-dominated solutions, more diversified and with an improved convergence. 2PPLs 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, Two-phase Pareto local search, 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 para o produto requerido sejam satisfeitas. Além disso, procura-se minimizar o número de caminhões necessários ao processo produtivo. O POLAD tem sido tratado na literatura como um problema de otimização mono-objetivo, sendo a função objetivo a 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.

Para a resolução do POLAD, a literatura tem mostrado várias abordagens baseadas em procedimentos heurísticos, uma vez que métodos exatos possuem uma aplicabilidade restrita (SOUZA *ET AL.*, 2010). Dentre esses trabalhos, destacamos: Costa (2005), que desenvolveu um algoritmo heurístico baseado em *Greedy Randomized Adaptive Search Procedures* - GRASP (FEO & RESENDE, 1995; RESENDE & RIBEIRO, 2010) e *Variable Neighborhood Search* - VNS (HANSEN & MLADENOVIC, 2001) para o POLAD usando seis tipos diferentes de movimentos para explorar o espaço de soluções; Coelho *et al.* (2008), que aplicaram o algoritmo heurístico GVILS, que combina os procedimentos GRASP, VND (MLADENOVIC & HANSEN, 1997) e *Iterated Local Search* - ILS (LOURENÇO *ET AL.*, 2003) e desenvolveram mais dois movimentos além daqueles de Costa (2005); Souza *et al.* (2010), que propuseram 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 & 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).

No presente trabalho são desenvolvidos três algoritmos multiobjetivos, sendo dois deles baseados em busca local e um em busca populacional. O primeiro tem como base o método *Multiobjective Variable Neighborhood Search* - MOVNS (GEIGER, 2004), o segundo o método *Two-phase Pareto Local Search* com VNS - 2PPLs (LUST *ET AL.*, 2011) e o terceiro, o método NSGA-II. Esses métodos têm sido aplicados, com sucesso, na solução de diversos problemas, motivando seu uso na resolução do POLAD. O terceiro, inclusive, conforme comentado anteriormente, já foi objeto de estudo em Pantuza (2011) para a solução do POLAD. Tem, assim, este 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 PROPOSTOS

2.1. REPRESENTAÇÃO DE UMA SOLUÇÃO

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 $R = [Y \mid N]$, sendo Y a matriz $|F| \times 1$ e N uma 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*.

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

Na Tabela 1, tem-se um exemplo de uma solução para o POLAD. Observa-se que na coluna CARGA, linha F_1, a dupla $\langle Car_1, 1\rangle$, indicando que o equipamento de carga Car_1 está alocado à frente F_1 e em operação. Na coluna CARGA, linha F_3, 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. Observa-se, ainda, na coluna CARGA, linha F_2, o valor $\langle D, 0\rangle$, informando que não existe equipamento de carga alocado à frente F_2 e que, portanto, esta frente está disponível. As demais colunas representam o número de viagens a serem realizadas por um caminhão a uma frente, considerando a compatibilidade entre o caminhão e o equipamento de carga alocado à frente. As células com os valores X indicam incompatibilidade entre um caminhão e o respectivo equipamento de carga.

Tabela 1 – Representação de uma solução

	Carga	*Cam 1*	*Cam 2*	...	*CamV V*
F_1	$\langle Car_1, 1\rangle$	8	X	...	X
F_2	$\langle D, 0\rangle$	0	0	...	0
F_3	$\langle Car_8, 0\rangle$	0	0	...	0
...	...	...	...	...	...
F_F	$\langle Car_5, 1\rangle$	0	9	...	3

2.2. ESTRUTURAS DE VIZINHANÇA

Para explorar o espaço de soluções foram utilizados os oito movimentos de Souza *et al*. (2010), brevemente apresentados a seguir.

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 na qual 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, 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}. Isto é, 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, 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, isto é, 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 unidade de uma célula passa para a outra, isto é, uma viagem de um caminhão associado a uma frente *i* passa para outro caminhão associado outra frente.

Movimento Troca de Carregadeiras - N^{CT}(s): Duas células distintas y_i e y_k da matriz *Y* tem seus valores permutados, ou seja, os equipamentos de carga que operam nas frentes *i* e *k* são trocados. Analogamente ao movimento *CG*, os equipamentos de carga 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.

2.3. AVALIAÇÃO DE UMA SOLUÇÃO

Uma solução *s* para o POLAD é avaliada por um vetor *z* de objetivos, dado pela Eq. (1), os quais devem ser minimizados:

$$z(s) = \left(z_1(s), z_2(s), z_3(s)\right) \tag{1}$$

Na Eq. (1), as parcelas $z_1(s)$, $z_2(s)$ e $z_3(s)$ mensuram, respectivamente, os desvios dos parâmetros de qualidade da mistura, os desvios de produção e a quantidade de caminhões usados. Tais funções objetivo e seus respectivos parâmetros são definidos em Souza *et al.* (2010).

2.4. ALGORITMOS PROPOSTOS

São propostos três algoritmos multiobjetivo. O primeiro deles, nomeado GRASP-MOVNS, consiste na combinação de procedimentos heurísticos *Greedy Randomized Adaptative Search Procedure* – GRASP (RESENDE & RIBEIRO, 2010) e *Multiobjective Variable Neighborhood Search* - MOVNS (GEIGER, 2004). O segundo, denominado GRASP-2PPLS, combina os procedimentos GRASP e *Two-phase Pareto Local Search* com VNS – 2PPLs (LUST *ET AL.*, 2011). Já o terceiro, denominado GRASP-NSGAII-PR, combina os procedimentos GRASP, NSGA-II e Reconexão por Caminhos – PR, do inglês *Path Relinking* (RIBEIRO, 2012).

O pseudocódigo do algoritmo GRASP-MOVNS está esquematizado na Figura 1.

Na linha 1 da Figura 1 é acionado o procedimento ConstroiConjuntoInicial (Figura 3) para gerar o conjunto inicial de soluções não-dominadas. Esse conjunto de soluções não-dominadas é gerado pelo procedimento parcialmente guloso GRASP (linhas 2, 3 e 4 da Figura 3), utilizando a mesma estratégia de Souza *et al.* (2010). O procedimento addSolution (linha 5 da Figura 3) encontra-se detalhado na Figura 2, esse procedimento adiciona na população potencialmente eficiente *Xe* as soluções geradas pelo GRASP. Nas linhas 4 e 5 mostra-se a seleção de um indivíduo da população de soluções potencialmente eficientes, marcando-o como "visitado". Quando todos os indivíduos estão com este marcador, a linha 28 retira tais marcadores.

```
Entrada: Vizinhanças N_k(x); graspMax; levelsMax
Saída: Aproximação de um conjunto eficiente Xe
1  Xe ← ConstroiConjuntoInicial(graspMax)
2  level ← 1 ; shaking ← 1
3  enquanto Critério de parado não satisfeito faça
4  |   Seleciona uma solução não visitada s ∈ Xe
5  |   Marque s como visitada
6  |   s' ← s
7  |   para i ← 1 até shaking faça
8  |   |   Selecione aleatoriamente uma vizinhança N_k(.)
9  |   |   s' ← Perturbação(s', k)
10 |   fim
11 |   Seja k_ult ← k
12 |   incrementa ← verdadeiro
13 |   para todo s'' ∈ N_{k_ult}(s') faça
14 |   |   addSolution(Xe, s'', f(s''), Added)
15 |   |   se Added = verdadeiro então
16 |   |   |   incrementa ← falso;
17 |   |   fim
18 |   fim
19 |   se incrementa = verdadeiro então
20 |   |   level ← level + 1
21 |   senão
22 |   |   level ← 1; shaking ← 1
23 |   fim
24 |   se level >= levelMax então
25 |   |   shaking ← shaking + 1 ; level ← 1
26 |   fim
27 |   se todo s ∈ Xe estão marcadas como visitadas então
28 |   |   Marque todos s ∈ Xe como não-visitado
29 |   fim
30 fim
31 retorna Xe
```

Figura 1 – Algoritmo GRASP-MOVNS

As variáveis *shaking* e *level*, linha 2 da Figura 1, regulam a perturbação utilizada no algoritmo. Esta 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 7 à 10 da Figura 1, ocorre o laço de perturbação do algoritmo. No mecanismo utilizado, quanto maior o valor da variável *shaking*, maior será a intensidade da perturbação na solução. Para cada unidade dessa variável aplica-se um movimento aleatório dentre as seis vizinhanças: N^{NV}, N^{CG}, N^{VC}, N^{VF}, N^{VT}, N^{CT}. A variável *level* regula quando a variável *shaking* será incrementada. A linha 22 retorna os valores das variáveis *level* e *shaking* para uma unidade, quando, pelo menos, uma solução é adicionada ao conjunto eficiente *Xe*.

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

```
Added ← verdadeiro
para todo x ∈ Xe faça
    se z(x) ⪯ z(s) então
        Added ← falso
        Break
    fim
    se z(s) ≺ z(x) então
        Xe ← Xe \ x
    fim
fim
se Added = verdadeiro então
    Xe ← Xe ∪ s
fim
retorna Xe
```

Figura 2 – Algoritmo addSolution

Entrada: $graspMax$;
Saída: Aproximação de um conjunto eficiente Xe

```
para i ← 1 até graspMax faça
    s_w ← ConstróiSoluçãoEstéril()
    Gere um número aleatório γ ∈ [0, 1]
    s_i ← ConstróiSoluçãoMinério(s_w, γ)
    addSolution(Xe, s_i, f(s_i))
fim
retorna Xe
```

Figura 3 – Algoritmo ConstroiConjuntoInicial

O algoritmo GRASP-2PPLS é apresentado na Figura 4. O algoritmo segue a mesma estrutura proposta por Lust *et al.* (2011), na qual o procedimento 2PPLS é utilizado em conjunto com um mecanismo de troca de vizinhanças, espelhado no método VNS.

O método 2PPLS é composto de duas fases. Na primeira, um conjunto inicial diversificado e com uma boa aproximação dos extremos de um conjunto eficiente é gerado (linha 1 da Figura 4). Já na segunda fase, aplica-se o método *Pareto Local Search* (PLS) a cada um dos indivíduos da população (linhas 4 até 21 da Figura 4). O método PLS é visto como uma generalização multiobjetivo do método da descida (LUST & TEGHEM, 2010).

Quando pelo menos uma solução não-dominada é adicionada ao conjunto *Xe*, linha 16 da Figura 4, a vizinhança corrente retorna à primeira vizinhança do conjunto de vizinhanças $N_k(x)$, passado como parâmetro do algoritmo. Caso não haja nenhum vizinho a ser adicionado ao conjunto de soluções não-dominadas *Xe*, a próxima estrutura de vizinhança é acionada (linha 18). O algoritmo termina sua execução quando todas as vizinhanças forem percorridas. A linha 19 assegura que o algoritmo não volta a soluções já visitadas.

Por fim, o pseudocódigo do GRASP-NSGAII-PR está esquematizado na Figura 5. Analogamente aos algoritmos GRASP-MOVNS e GRASP-2PPLS, a população inicial P_0 do algoritmo é iniciada com a adição de indivíduos gerados pelo procedimento GRASP de Souza *et al.* (linhas 3, 4 e 5 da Figura 5); porém, neste caso, o critério de parada (linha 2) é o tamanho da população P_0, parâmetro de entrada do algoritmo.

Entrada: *graspMax*; Vizinhanças $\mathcal{N}_k(x)$
Saída: Conjunto Eficiente Xe

$P_0 \leftarrow$ ConstroiConjuntoInicial($graspMax$)
$Xe \leftarrow P_0$; $P \leftarrow P_0$; $P_a \leftarrow \emptyset$
$k \leftarrow 1$ {Tipo de estrutura de vizinhança corrente}
enquanto $k <= r$ **faça**
  para todo $p \in P$ **faça**
    para todo $p' \in N_k(p)$ **faça**
      se $f(p) \npreceq f(p')$ **então**
        addSolution($Xe, p', f(p'), Added$)
        se $Added = verdadeiro$ **então**
          addSolution($P_a, p', f(p')$)
        fim
      fim
    fim
  fim
  se $P_a \neq \emptyset$ **então**
    $P \leftarrow P_a$; $P_a \leftarrow \emptyset$; $k \leftarrow 1$
  senão
    $k \leftarrow k+1$
    $P \leftarrow Xe \setminus \{x \in Xe \mid$ Pareto Local ótimo em relação $\mathcal{N}_k(x)\}$
  fim
fim
retorna Xe

Figura 4 – Algoritmo GRASP-2PPLS com VNS

Na linha 8 da Figura 5 é acionado o procedimento de seleção, cruzamento e mutação, SelecaoCruzamentoPRMutacao, cujo pseudocódigo está apresentado na Figura 6. A Reconexão por Caminhos foi utilizada como um operador avançado de cruzamento, tal como em Ribeiro *et al*. (2012).

Entrada: Tamanho da população N; Vizinhanças $N_k(x)$
Saída: Conjunto Eficiente Xe

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

Figura 5 – Algoritmo GRASP-NSGAII-PR

O procedimento *Nondominated Sorting Genetic Algorithm* II (DEB *ET AL*., 2002) é ativado na linha 9 da Figura 5.

O Algoritmo SelecaoCruzamentoPRMutacao (Figura 6) 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 da Figura 6 o indivíduo s recebe o melhor indivíduo, considerando a função de avaliação mono-objetivo de Souza *et. al*. (2010), gerado pelo procedimento de

Reconexão por Caminhos. A Reconexão é aplicada de forma bidirecional, isto é, ela é acionada tanto com o indivíduo s_1 sendo a solução base quanto sendo a solução guia. Na estratégia de Reconexão utilizada, os atributos considerados são as posições que as carregadeiras ocupam na solução guia. A cada passo do procedimento, é verificado 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*, e as viagens associadas a essa carregadeira são mantidas. A seguir, faz-se uma busca local baseada no procedimento VND, utilizando apenas as estruturas que modificam o número de viagens dos caminhões, no caso, as vizinhanças N^{NV}, N^{VC} e N^{VF}, preservando assim as carregadeiras já fixadas. O atributo da solução guia escolhido para inserção na solução base é aquele que gerar o melhor valor em relação à função de avaliação mono-objetivo. Este procedimento é repetido até que todos os atributos sejam fixados, de forma que a carregadeira alocada a cada frente da solução base seja a mesma da solução guia.

```
   Entrada: População de pai P; Vizinhanças N^(k)(.); mutationRate e localSearchRate
   Saída: População de offprings Q
1  enquanto |Q| < N faça
2      Selecione dois indivíduos aleatórios s1 e s2 ∈ P
3      s ← melhor indivíduo (ReconexãoPorCaminhos(s1, s2),
       ReconexãoPorCaminhos(s2, s1))
4      addSolution(Q, s, f(s))
5      Gere um número aleatório ap_mutation ∈ [0, 1]
6      se ap_mutation < mutationRate então
7          Selecione aleatoriamente uma vizinhança N^(k)(.)
8          s' ← N^(k)(s)
9      senão
10         s' ← s
11     fim
12     Gere um número aleatório ap_localSearch ∈ [0, 1]
13     se ap_localSearch < localSearchRate então
14         s'' ← VND(s')
15         addSolution(Q, s'', f(s''))
16     senão
17         addSolution(Q, s', f(s'))
18     fim
19 fim
20 retorna Q
```

Figura 6 – Algoritmo SelecaoCruzamentoPRMutacao

Na linha 8 da Figura 6 aplica-se um movimento aleatório no indivíduo *s* (operador de mutação), caso a variável $ap_{mutation}$ seja menor que a taxa de mutação *mutationRate*. O mesmo acontece na linha 14, na qual é aplicado o procedimento VND caso uma condição análoga seja satisfeita. Finalmente, nas linhas 4, 15 e 17 é verificado se os indivíduos *s*, *s'* e *s''* devem ser adicionados à população de *offprings Q*.

3. EXPERIMENTOS COMPUTACIONAIS E ANÁLISES

Os algoritmos propostos foram implementados em C++ com auxílio do *framework* OptFrame (COELHO *ET AL*., 2011a).

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 o algoritmo foi usado o conjunto de 8 instâncias de Souza *et al*. (2010), disponível em http://www.iceb.ufop.br/decom/prof/marcone/projects/mining.html.

Após uma bateria preliminar de testes, os seguintes valores para os parâmetros dos algoritmos desenvolvidos foram utilizados: *graspMax* = 400, *levelMax* = 10, *N* = 35, *mutationRate* = 8% e *localSearchRate* = 20%. A variável *shaking* teve seu valor máximo fixado no valor cinco. Cada algoritmo foi executado 30 vezes, cada qual por 120 segundos, assim como em Souza *et al.* (2010).

Para verificação do desempenho dos algoritmos, foram usadas as métricas de Hipervolume (ZITZLER & THIELE, 1998), Espaçamento (SCHOTT, 1995), Cobertura (ZITZLER & THIELE, 1998) e Cardinalidade (ZITZLER *ET AL.*, 2003). As Tabelas 2 a 5 mostram os resultados obtidos. A coluna "Instância" indica a instância utilizada; a coluna "Melhor" indica o melhor valor da métrica analisada obtido nas 30 execuções e as colunas "Média" e "Desv. Pad." indicam, respectivamente, a média e desvio padrão da amostra.

A Tabela 2 mostra os valores da métrica de hipervolume para os três algoritmos desenvolvidos. O cálculo dessa métrica foi realizado com o auxílio da ferramenta computacional de Beume *et al.* (2009). Analisando os resultados, verifica-se que o algoritmo GRASP-2PPLS foi capaz de obter os conjuntos de soluções não-dominadas com as melhores convergências em seis instâncias, além de apresentar as melhores médias em cinco delas. Ressalta-se, ainda, que o algoritmo GRASP-MOVNS obteve boas frentes de Pareto nas instâncias opm1, opm2, opm5 e opm6.

Tabela 2 – GRASP-2PPLS × GRASP-MOVNS × GRASP-NSGAII-PR: Hipervolume

	GRASP-2PPLS			GRASP-MOVNS			GRASP-NSGAII-PR		
Instância	Melhor	Média	Desv. Pad.	Melhor	Média	Desv. Pad.	Melhor	Média	Desv. Pad.
opm1	**81,71**	78,13	1,54	80,37	**78,23**	1,12	78,43	75,68	0,89
opm2	76,47	75,15	0,56	**79,49**	**76,12**	1,49	75,26	73,15	1,05
opm3	**77,94**	**71,17**	4,18	76,97	70,42	2,18	69,95	64,98	2,38
opm4	**76,80**	**67,45**	4,67	75,61	66,94	3,00	65,09	59,75	2,89
opm5	**81,88**	78,04	1,41	80,91	**78,67**	1,37	77,36	75,68	0,72
opm6	80,00	**77,97**	0,91	**80,43**	77,23	1,20	77,19	74,32	0,83
opm7	**78,78**	**75,28**	1,88	78,48	73,39	3,54	76,15	73,69	2,28
opm8	**78,88**	**73,77**	2,26	78,57	72,96	3,90	76,05	73,62	1,57

A Tabela 3 apresenta a comparação dos resultados utilizando a métrica de Espaçamento. Percebe-se que a variante GRASP-2PPLS obteve os melhores resultados, pois foi capaz de gerar conjuntos mais uniformemente espaçados.

Tabela 3 – GRASP-2PPLS × GRASP-MOVNS × GRASP-NSGAII-PR: Espaçamento

	GRASP-2PPLS			GRASP-MOVNS			GRASP-NSGAII-PR		
Instância	Melhor	Média	Desv. Pad.	Melhor	Média	Desv. Pad.	Melhor	Média	Desv. Pad.
opm1	**1177,93**	4566,51	5058,15	1662,54	**2868,67**	1099,72	2589,54	6924,42	5007,23
opm2	**520,91**	**1166,88**	389,56	1066,13	2041,59	970,75	1812,14	6557,20	5030,65
opm3	**4,00**	**3824,96**	3443,70	2250,96	6188,99	3331,59	4850,50	14368,62	5173,60
opm4	**3,77**	**2243,74**	1235,66	2411,89	8232,88	3957,29	4865,89	18295,27	7046,64
opm5	**1251,93**	4795,68	3882,54	1400,84	**2563,71**	1057,14	2578,39	7544,48	5529,35
opm6	**697,34**	**1212,87**	344,61	1087,95	2025,84	695,00	2218,05	5963,77	4650,80
opm7	**0,00**	**6676,54**	8042,17	2917,66	8311,12	3437,74	4924,61	14673,57	4646,02
opm8	**0,00**	**6381,38**	7678,35	2258,14	8045,67	4684,92	4974,85	14874,73	5897,70

Os algoritmos baseados em procedimentos de busca local, GRASP-2PPLS e GRASP-MOVNS, apresentaram os melhores resultados em relação às métricas de hipervolume e espaçamento. Deste modo, as métricas de Cobertura e Cardinalidade foram aplicadas, apenas, entre esses dois algoritmos. A Tabela 4 mostra o resultado da comparação entre esses dois algoritmos com relação à Cobertura e a Tabela 5, em relação à Cardinalidade.

Para utilizar a métrica de cardinalidade, foi criado um conjunto Pareto Referência ($|REF|$) obtido em duas etapas. Primeiramente, foram realizadas 30 execuções de duas horas com cada uma das instâncias. Essa bateria de teste consistiu na execução dos algoritmos GRASP-2PPLS e GRASP-MOVNS, alternadamente. Dado um conjunto inicial de soluções não-dominadas, o algoritmo GRASP-2PPLS teve como critério de parada o ótimo local em relação às estruturas de vizinhanças utilizadas; em seguida, o conjunto de soluções não-dominadas obtido era entregue ao algoritmo GRASP-MOVNS que refinava essa fronteira de Pareto com um tempo máximo de processamento de 5 minutos e, novamente, o algoritmo GRASP-2PPLS era acionado. Por fim, o conjunto Pareto Referência foi obtido com a união de todas as fronteiras de Pareto encontradas nas 30 execuções, mais as fronteiras obtidas em todos os experimentos executados neste trabalho. Os conjuntos $|D1|$ e $|D2|$, representam as cardinalidades dos conjuntos de Pareto obtidos com a união das soluções não-dominadas encontradas pelos algoritmos GRASP-2PPLS e GRASP-MOVNS, respectivamente.

Tabela 4 – GRASP-2PPLS × GRASP-MOVNS: Cobertura

	C (GRASP-2PPLS,GRASP-MOVNS)			*C* (GRASP-MOVNS,GRASP-2PPLS)		
Instância	Melhor	**Média**	Desv. Pad.	Melhor	Média	Desv. Pad.
opm1	**1,00**	**0,79**	0,14	0,13	0,02	0,03
opm2	**1,00**	**0,80**	0,03	0,10	0,06	0,11
opm3	**1,00**	**0,65**	0,21	0,25	0,02	0,05
opm4	**0,95**	**0,57**	0,23	0,35	0,04	0,09
opm5	**0,73**	**0,38**	0,14	0,57	0,22	0,14
opm6	**0,96**	**0,79**	0,11	0,18	0,07	0,06
opm7	**0,90**	**0,34**	0,25	0,50	0,11	0,17
opm8	0,90	**0,41**	0,29	**1,00**	0,11	0,25

Os resultados apresentados na Tabela 4 mostram que o algoritmo GRASP-2PPLS foi capaz de gerar melhores conjuntos que o algoritmo GRASP-MOVNS em 7 instâncias.

A Tabela 5 mostra o número de soluções não-dominadas obtidas pelos algoritmos GRASP-2PPLS e GRASP-MOVNS, bem como o número soluções pertencentes aos conjuntos de referência.

Tabela 5 – GRASP-2PPLS × GRASP-MOVNS: Cardinalidade

		GRASP-2PPLS		GRASP-MOVNS	
Instância	$\|REF\|$	$\|D1\|$	$\|REF \cap D1\|$	$\|D2\|$	$\|REF \cap D2\|$
opm1	234	117	**18**	141	4
opm2	271	175	**35**	144	0
opm3	151	125	**33**	74	1
opm4	148	152	**9**	87	1
opm5	172	99	**41**	128	38
opm6	209	200	**97**	121	3
opm7	104	62	**17**	78	2
opm8	139	104	**39**	69	8
Total de Soluções	1428	1034	289	842	57

Analisando a Tabela 5 nota-se que o algoritmo GRASP-2PPLS foi capaz de encontrar um total de 289 soluções pertencentes aos conjuntos de referência. Já o algoritmo GRASP-MOVNS foi capaz de encontrar apenas 57 soluções. Este resultado mostra a robustez do algoritmo GRASP-2PPLS de encontrar, em tempo reduzido, um considerável número de soluções não-dominadas pertencentes às aproximações para os conjuntos Pareto Referência.

Por fim, buscou-se verificar se esta nova proposta multiobjetivo conseguiria gerar uma boa solução mono-objetivo. A Tabela 6 compara o desempenho do algoritmo GRASP-2PPLS frente ao do algoritmo GGVNS de Souza *et al*. (2010), com relação às melhores soluções mono-objetivo obtidas. Analisando-a, percebe-se que o algoritmo GRASP-MOVNS mostrou-se competitivo com o algoritmo mono-objetivo da literatura GGVNS, uma vez que obteve soluções melhores ou iguais em sete instâncias.

Tabela 6 – GRASP-2PPLS × GGVNS: Melhores soluções

	Tempo	GRASP-2PPLS	GGVNS
Instância	(min)	Melhor	Melhor
opm1	2	**228,12**	230,12
opm2	2	**256,37**	**256,37**
opm3	2	164046,32	**164039,12**
opm4	2	**164074,32**	164099,66
opm5	2	**227,04**	228,09
opm6	2	**236,35**	236,58
opm7	2	**164018,81**	164021,28
opm8	2	**164022,63**	164023,73

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.

Em vista da natureza combinatória do problema, foram propostos três algoritmos heurísticos multiobjetivos: GRASP-2PPLS, GRASP-MOVNS e GRASP-NSGAII-PR. Todos usam o GRASP para gerar os conjuntos iniciais de soluções não-dominadas. Os dois primeiros são métodos de busca local inspirados nos métodos 2PPLs (LUST *ET AL*., 2011) e MOVNS (GEIGER, 2004), respectivamente, enquanto o terceiro é baseado no método NSGA-II, utilizando o procedimento *Path Relinking* - PR como operador de cruzamento.

Para comparar o desempenho dos algoritmos multiobjetivos desenvolvidos foram usadas oito instâncias da literatura e quatro métricas de avaliação: Hipervolume, Espaçamento, Cobertura e Cardinalidade. Primeiramente, foi feita uma bateria de testes com os três algoritmos utilizando as métricas de Hipervolume e Espaçamento. Os algoritmos GRASP-2PPLS e GRASP-MOVNS mostraram-se superiores frente ao algoritmo GRASP-NSGAII-PR. Em seguida, o algoritmo GRASP-2PPLS foi comparado ao algoritmo GRASP-MOVNS utilizando a métrica de Cobertura. O algoritmo GRASP-2PPLS, que já havia apresentado melhores resultados nas métricas anteriores, com frentes de Pareto mais diversificadas e com uma melhor convergência, obteve, novamente, frentes de Pareto que cobriram as frentes obtidas pelo algoritmo GRASP-MOVNS. Por fim, utilizando uma função mono-objetivo, as soluções obtidas pelo algoritmo GRASP-2PPLS foram comparadas a um algoritmo da literatura mono-objetivo, denominado GGVNS, de Souza *et al*. (2010). O algoritmo GRASP-2PPLS foi capaz de obter soluções melhores ou iguais em sete instâncias, mostrando, assim, seu poderio tanto em aplicações multiobjetivo quanto mono-objetivo. Os resultados obtidos validam, pois, a proposta apresentada.

AGRADECIMENTOS

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

REFERÊNCIAS BIBLIOGRÁFICAS

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

Beume, N.; Fonseca, C. M.; Lopez-Ibanez, M; Paquete, L. & Vahrenhold, J. On the complexity of computing the hypervolume indicator. *Evolutionary Computation, IEEE Transactions on*, v. 13, n. 5, p. 1075 - 1082, 2009.

Coelho, I. M.; Munhoz, P. L. A.; Haddad, M. N.; Coelho, V. N.; Silva, M. M; Souza, M. J. F. & Ochi, L. S.

A computational framework for combinatorial optimization problems. *In: VII ALIO/EURO Workshop on Applied Combinatorial Optimization*, Portugal, p. 51-54, 2011a.

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* - SIMPEP, Bauru/SP, 2008.

Coelho, V. N.; Souza, M. J. F.; Coelho, I. M.; Guimarães, F. G. & Coelho, B. N. Estratégias evolutivas aplicadas a um problema de programação inteira mista. In *Anais X Congresso Brasileiro de Inteligência Computacional* - CBIC, Fortaleza/CE, p.1-8, 2011b.

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

Costa, F. P. *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, Escola de Minas, UFOP, Ouro Preto, 2005.

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

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

Geiger, M. J. Randomised variable neighbourhood search for multi objective optimisation. *In Proceedings of the 4th EUME Workshop Design and Evaluation of Advanced Hybrid MetaHeuristics*, Nottingham, United Kingdom, p. 34-42, 2004.

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

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

Lourenço, H. R., Martin, O. C., & Stützle, T. *Iterated local search*. In Glover, F. and Kochenberger, G., editors, Handbook of Metaheuristics. Kluwer Academic Publishers, Boston, 2003.

Mladenovic, N. & Hansen, P. A variable neighborhood search. *Computers and Operations Research*, v. 24, p. 1097-1100, 1997.

Pantuza, G. Métodos de Otimização Multiobjetivo e de Simulação aplicados ao Problema de Planejamento Operacional de Lavra em Minas a Céu Aberto. Dissertação (Dissertação de Mestrado) — Programa de Pós-Graduação em Engenharia Mineral - PPGEM da Universidade Federal de Ouro Preto, Ouro Preto, 2011.

Ribeiro, C.C. & Resende, M. G. C. Path-relinking intensification methods for stochastic local search algorithms*. Journal of Heuristics*, Springer Netherlands, v. 18, p. 1-22, 2012.

Resende, M. G. C. & Ribeiro, C. C. *Greedy randomized adaptive search procedures: Advances, hybridizations and applications*. In Gendreau, M. and Potvin, J., editors, Handbook of Metaheuristics. Springer, 2 ed., New York: Springer, p. 283-319, 2010.

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

Souza, M. J. F., Coelho, I. M., Ribas, S., Santos, H. G., & Merschmann, L. H. C. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Researc8h*, v. 207, n. 2, p. 1041-1051, 2010.

Zitzler, E.; Thiele, L. Multiobjective optimization using evolutionary algorithms - a comparative case study. *In: EIBEN, A. et al. (Ed.). Parallel Problem Solving from Nature - PPSN V.* Springer Berlin/Heidelberg, v. 1498, 1998.

Zitzler, E.; Thiele, L.; Laumanns, M.; Foneseca, C. M. & Grunert F. V. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 2, p. 117-132, 2003.