Multi-objective approaches for the open-pit mining operational planning problem

V.N. Coelho\textsuperscript{a}, M.J.F. Souza\textsuperscript{a}, I.M. Coelho\textsuperscript{b}, F.G. Guimarães\textsuperscript{c}, T. Lust\textsuperscript{d} and R.C. Cruz\textsuperscript{a}

\textsuperscript{a} Federal University of Ouro Preto, Dept. of Computer Science, Ouro Preto, MG
\textsuperscript{b} Fluminense Federal University, Institute of Computing, Niterói, RJ / Inria Lille Nord Europe
\textsuperscript{c} Federal University of Minas Gerais, Department of Electrical Engineering, Belo Horizonte, MG
\textsuperscript{d} LIP6-CNRS, UPMC. 4 Place Jussieu 75252 Cedex 05 Paris

imcoelho@ic.uff.br

November 29, 2013
Summary

Introduction

Heuristic Approach

Proposed Algorithms

Computational Experiments and Analysis

Conclusions
Open-Pit Mining Operational Planning (OPMOP)

Introduction

The open-pit mining operational planning problem (OPMOP) deals with:

• Ore and waste rocks
Open-Pit Mining Operational Planning (OPMOP)

Introduction

The open-pit mining operational planning problem (OPMOP) deals with:

- Ore and waste rocks
- Shovels
Open-Pit Mining Operational Planning (OPMOP)

Introduction

The open-pit mining operational planning problem (OPMOP) deals with:

- Ore and waste rocks
- Shovels
- Trucks
The open-pit mining operational planning problem (OPMOP) deals with:

- Ore and waste rocks
- Shovels
- Trucks

Optimize...
Open-Pit Mining Operational Planning (OPMOP)

Introduction

The open-pit mining operational planning problem (OPMOP) deals with:

- Ore and waste rocks
- Shovels
- Trucks

Optimize...

... the allocation of mining equipment to the pits and determine the number of trips for each truck, while respecting the equipment limits, production goals and the desired mineral composition.
Images from an open-pit mine in Brazil, situated in a region called “Quadriláteros Ferríferos”
Open-Pit Mining Operational Planning (OPMOP)
Characteristics

- Goals for ore and waste rock production
Open-Pit Mining Operational Planning (OPMOP)

Characteristics

• Goals for ore and waste rock production
• Goals for ore parameters
Open-Pit Mining Operational Planning (OPMOP)

Characteristics

- Goals for ore and waste rock production
- Goals for ore parameters
- Limited heterogeneous shovels
Open-Pit Mining Operational Planning (OPMOP)

Characteristics

- Goals for ore and waste rock production
- Goals for ore parameters
- Limited heterogeneous shovels
- Limited heterogeneous trucks with compatibility issues
Open-Pit Mining Operational Planning (OPMOP)

Characteristics

- Goals for ore and waste rock production
- Goals for ore parameters
- Limited heterogeneous shovels
- Limited heterogeneous trucks with compatibility issues
- Dynamic allocation of equipment
Open-Pit Mining Operational Planning

- 60% Fe, 1.5% SiO₂
- 68% Fe, 2% SiO₂
- 50% Fe, 1% SiO₂
- Mixture
- Final Product: 65% Fe, 1.6% SiO₂
Open-Pit Mining Operational Planning

Motivation

- Production costs involved are high
Open-Pit Mining Operational Planning

Motivation

- Production costs involved are high
- OPMOP is a problem classified as $\mathcal{NP}$-hard
Open-Pit Mining Operational Planning

Motivation

- Production costs involved are high
- OPMOP is a problem classified as $\mathcal{NP}$-hard
- The decision must be fast
Open-Pit Mining Operational Planning

Motivation

- Production costs involved are high
- OPMOP is a problem classified as $\mathcal{NP}$-hard
- The decision must be fast
- Successful mono-objective heuristic approaches in literature
Open-Pit Mining Operational Planning

Motivation

- Production costs involved are high
- OPMOP is a problem classified as $NP$-hard
- The decision must be fast
- Successful mono-objective heuristic approaches in literature
- Novel multi-objective approach
Multi-objective Model

Conflicting objectives

Minimize each part of the function \( z(s) = (z_1(s), z_2(s), z_3(s)) \), composed of three conflicting objectives.
Multi-objective Model

Conflicting objectives

Minimize each part of the function \( z(s) = (z_1(s), z_2(s), z_3(s)) \), composed of three conflicting objectives.

Let: \( T \) ore parameters; \( V \) vehicles; \( d_t \) deviations of ore parameters from goals; \( P_o \) and \( P_w \) production deviations for ore and waste rock, respectively; \( U_v \) utilization rate for vehicles; cost weights \( \lambda_j, \alpha, \beta \) and \( \omega \).
Multi-objective Model

Conflicting objectives

Minimize each part of the function $z(s) = (z_1(s), z_2(s), z_3(s))$, composed of three conflicting objectives.

Let: $T$ ore parameters; $V$ vehicles; $d_t$ deviations of ore parameters from goals; $P_o$ and $P_w$ production deviations for ore and waste rock, respectively; $U_v$ utilization rate for vehicles; cost weights $\lambda_j$, $\alpha$, $\beta$ and $\omega$.

- Ore quality

$$z_1(s) = \sum_{t \in T} \lambda_t^- d_t^- + \sum_{t \in T} \lambda_t^+ d_t^+$$  \hspace{1cm} (1)
Multi-objective Model

Conflicting objectives

Minimize each part of the function $z(s) = (z_1(s), z_2(s), z_3(s))$, composed of three conflicting objectives.

Let: $T$ ore parameters; $V$ vehicles; $d_t$ deviations of ore parameters from goals; $P_o$ and $P_w$ production deviations for ore and waste rock, respectively; $U_v$ utilization rate for vehicles; cost weights $\lambda_j$, $\alpha$, $\beta$ and $\omega$.

- Ore quality

$$z_1(s) = \sum_{t \in T} \lambda_t^- d_t^- + \sum_{t \in T} \lambda_t^+ d_t^+$$

(1)

- Production quantity

$$z_2(s) = \alpha^- P_o^- + \alpha^+ P_o^+ + \beta^- P_w^- + \beta^+ P_w^+$$

(2)
Multi-objective Model

Conflicting objectives

Minimize each part of the function \( z(s) = (z_1(s), z_2(s), z_3(s)) \), composed of three conflicting objectives.

Let: \( T \) ore parameters; \( V \) vehicles; \( d_t \) deviations of ore parameters from goals; \( P_o \) and \( P_w \) production deviations for ore and waste rock, respectively; \( U_v \) utilization rate for vehicles; cost weights \( \lambda_j \), \( \alpha \), \( \beta \) and \( \omega \).

- **Ore quality**
  \[
  z_1(s) = \sum_{t \in T} \lambda_t^- d_t^- + \sum_{t \in T} \lambda_t^+ d_t^+ \quad (1)
  \]

- **Production quantity**
  \[
  z_2(s) = \alpha^- P_o^- + \alpha^+ P_o^+ + \beta^- P_w^- + \beta^+ P_w^+ \quad (2)
  \]

- **Number of trucks**
  \[
  z_3(s) = \sum_{v \in V} \omega_l U_v \quad (3)
  \]
Heuristic Approach

Representation of a Solution

Table: Representation of a Solution

<table>
<thead>
<tr>
<th>Shovel</th>
<th>Truck₁</th>
<th>Truck₂</th>
<th>...</th>
<th>Truckₜ</th>
</tr>
</thead>
<tbody>
<tr>
<td>F₁</td>
<td>8</td>
<td>X</td>
<td>...</td>
<td>X</td>
</tr>
<tr>
<td>F₂</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>F₃</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>F₆</td>
<td>0</td>
<td>9</td>
<td>...</td>
<td>3</td>
</tr>
</tbody>
</table>
Heuristic Approach

Neighborhood Structures (1/2)
# Heuristic Approach

Neighborhood Structures (2/2)

<table>
<thead>
<tr>
<th>Frentes</th>
<th>Carregadeiras</th>
<th>Cam1</th>
<th>Cam2</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Car₁</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>F2</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>Car₂</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>F4</td>
<td>x</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frentes</th>
<th>Carregadeiras</th>
<th>Cam1</th>
<th>Cam2</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Car₂</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>F2</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>Car₃</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>F4</td>
<td>x</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frentes</th>
<th>Carregadeiras</th>
<th>Cam1</th>
<th>Cam2</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Car₃</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>F2</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>Car₁</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>F4</td>
<td>x</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
Multi-objective Open-Pit Mining Operational Planning

Proposal

Heuristics

- Greedy Randomized Adaptative Search Procedure (GRASP)
  1. Two-phase Pareto Local Search with VNS (2PPLS-VNS)
  2. Multi-objective Variable Neighborhood Search (MOVNS)
  3. Non-dominated Sorting Genetic Algorithm II (NSGA-II)

Comparison metrics:

1. Spacing
2. Hypervolume
3. Coverage
Two-Phase Pareto Local Search

Basic principle

Composed of two phases:

1. In the first phase, an initial population, diversified and with a good approximation of the efficient set is generated.

2. In the second phase, the method Pareto Local Search (PLS) is applied to each individual in the population.

PLS
This method is the generalization in the multiobjective case of a basic metaheuristic: the hill-climbing method.
Heuristic Approach

Initial Population Generation

Greedy Randomized Initial Solution

- Allocation of the shovels
- Distribution of trips
Heuristic Approach

Initial Population Generation

Greedy Randomized Initial Solution

- Allocation of the shovels
- Distribution of trips
- Waste pits

Pits, the best is the one with the greatest mass
Shovels, the best is the one with the greatest production
Trucks, the largest one is the best.
Heuristic Approach

Initial Population Generation

Greedy Randomized Initial Solution

- Allocation of the shovels
- Distribution of trips

- Waste pits
  - Pits, the best is the one with the greatest mass
  - Shovels, the best is the one with the greatest production
  - Trucks, the largest one is the best.

- Ore pits
  - Pits, the best pit is the one that has the least deviation of the control parameters
  - Shovels, the best is the one with the greatest production
  - Trucks, the smallest one is the best.
Algorithm 1: G2PPLs-VNS

**Input:** graspMax; Neighborhoods $\mathcal{N}_k(x)$

**Output:** Approximation of the efficient set $X_e$

$P_0 \leftarrow \text{BuildInicialSet}(\text{graspMax})$;

$X_e \leftarrow \text{2PPLs-VNS}(P_0, \mathcal{N}_k(x))$;

return $X_e$
Algorithm 2: BuildInitialSet

Input: $\text{graspMax}$;

Output: Approximation of the efficient set $X_e$

for $i \leftarrow 1$ until $\text{graspMax}$ do

\[ s_w \leftarrow \text{BuildWasteSolution}() \]

Generate a random number $\gamma \in [0, 1]$ 

\[ s_i \leftarrow \text{BuildOreSolution}(s_w, \gamma) \]

addSolution($X_e, s_i, f(s_i)$)

end

return $X_e$
G2PPLS-VNS

**Algorithm 3: 2PPLs com VNS**

**Entrada:** Aproximação inicial de um conjunto eficiente $P_0$; Vizinhanças $N_k(x)$

**Saída:** Conjunto Eficiente $X_e$

1. $X_e \leftarrow P_0; P \leftarrow P_0; P_a \leftarrow \emptyset$
2. $k \leftarrow 1 \{\text{Tipo de estrutura de vizinhança corrente}\}$
3. **enquanto** $k \leq r$ **faça**
   4. **para todo** $p \in P$ **faça**
      5. **para todo** $p' \in N_k(p)$ **faça**
         6. **se** $f(p) \nleq f(p')$ **então**
            7. addSolution($X_e, p', f(p'), Added$)
            8. **se** $Added = \text{verdadeiro}$ **então**
               9. addSolution($P_a, p', f(p')$)
               10. **fim**
         11. **fim**
      12. **fim**
   13. **fim**
   14. **se** $P_a \neq \emptyset$ **então**
      15. $P \leftarrow P_a; P_a \leftarrow \emptyset; k \leftarrow 1$
      16. **senão**
      17. $k \leftarrow k + 1$
      18. $P \leftarrow X_e \setminus \{x \in X_e \mid \text{Pareto Local ótimo em relação } N_k(x)\}$
      19. **fim**
   20. **fim**
21. **retorna** $X_e$
Algorithm 4: GMOVNS

Input: Neighborhoods $N_k(x)$; graspMax; levelMax
Output: Approximation of the efficient set $X_e$

$X_e \leftarrow \text{BuildInitialSet}(\text{graspMax})$; $\text{level} \leftarrow 1$; $\text{shaking} \leftarrow 1$

while stop criterion not satisfied do
    Select a not “visited” solution $s \in X_e$ and check it as “visited”
    $s' \leftarrow s$
    for $i \leftarrow 1$ until $\text{shaking}$ do
        Select one neighborhood $N_k(.)$ at random
        $s' \leftarrow \text{Shake}(s', k)$
    end
    Let $k_{ult} \leftarrow k$; $\text{changeLevel} \leftarrow true$
    forall $s'' \in N_{k_{ult}}(s')$ do
        addSolution($X_e, s'', f(s''), \text{Added}$)
        if $\text{Added} = true$ then
            $\text{changeLevel} \leftarrow false$
        end
    end
    if $\text{changeLevel} = true$ then
        $\text{level} \leftarrow \text{level} + 1$
    else
        $\text{level} \leftarrow 1$; $\text{shaking} \leftarrow 1$
    end
    if $\text{level} = \text{levelMax}$ then
        $\text{level} \leftarrow 1$; $\text{shaking} \leftarrow \text{shaking} + 1$
    end
    if all $s \in X_e$ are “visited” then
        check all $s \in X_e$ as “non-visited” solutions
    end
end
return $X_e$
GNSGAII-PR

Basic idea

- The methods BuildInitialSet changes a little;
- Generate a population of solutions of good quality and well diversified. We do not warrant that this set is composed of non-dominated solutions;
- The structure of the algorithm follows the same literature. Changing only the genetic operators: Path Relinking (Ribeiro and Resende, 2012) and mutation.
**GNSGAII-PR**

**Basic pseudocode**

---

**Algorithm 5: GNSGAII-PR**

**Input:** Population size $N$; Neighborhoods $N_k(x)$  
**Output:** Approximation of the efficient set $X_e$  
Initial population $P_0$  

while $|P_0| \leq N$ do  
  $s_w \leftarrow \text{BuildWasteSolution}()$  
  Generate a random number $\gamma \in [0, 1]$  
  $s_i \leftarrow \text{BuildOreSolution}(s_w, \gamma)$  
  $P_0 \leftarrow s_i$  
end  

$Q_0 \leftarrow \text{SelectionPRCrossoverMutation}(P_0, \text{Neighborhoods } N^{(k)}(.))$  

$X_e \leftarrow \text{NSGA-II}(P_0, Q_0, N, \text{SelectionPRCrossoverMutation}(.))$  

return $X_e$
GNSGAII-PR

**Algorithm 6: selectionPRCrossoverMutation**

**Input:** mutationRate; localSearchRate

**Input:** Population P; Neighborhoods \( N^{(k)}(.) \)

**Output:** Offspring population \( Q \)

```
while |Q| ≤ N do
    Select two random individuals \( s_1 \) and \( s_2 \) ∈ P
    \( s ← \text{best}(\text{Path Relinking}(s_1, s_2), \text{Path Relinking}(s_2, s_1)) \)
    addSolution(\( Q, s, f(s) \))
    Generate a random number \( ap_{\text{mutation}} \in [0, 1] \)
    if \( ap_{\text{mutation}} < \text{mutationRate} \) then
        Select one neighborhood \( N_k(.) \) at random
        \( s' ← N^{(k)}(s) \)
    else
        \( s' ← s \)
    end
    Generate a random number \( ap_{\text{localSearch}} \in [0, 1] \)
    if \( ap_{\text{localSearch}} < \text{localSearchRate} \) then
        \( s'' ← \text{VND}(s') \)
        addSolution(\( Q, s'', f(s'') \))
    else
        addSolution(\( Q, s', f(s') \))
    end
end
turn Q
```
Experiments

- The code was implemented using the computational framework OptFrame\(^1\).
- The battery of tests was composed of 30 runs for each algorithm with a computational time limited to 2 minutes since this runtime is suitable for real applications. The instances used for testing the algorithms were those of Souza \textit{et al.} (2010)\(^2\)

\(^1\) [http://optframe.sourceforge.net](http://optframe.sourceforge.net)
# Spacing and Hypervolume metrics

**Table:** G2PPLS-VNS × GMOVNS × GSGAII-PR: Spacing and Hypervolume

<table>
<thead>
<tr>
<th>Instance</th>
<th>Spacing</th>
<th>Hypervolume (10^6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G2PPLS-VNS</td>
<td>GMOVNS</td>
</tr>
<tr>
<td>opm1</td>
<td>4566.51</td>
<td>2868.68</td>
</tr>
<tr>
<td>opm2</td>
<td>1166.88</td>
<td>2041.59</td>
</tr>
<tr>
<td>opm3</td>
<td>3824.96</td>
<td>6188.99</td>
</tr>
<tr>
<td>opm4</td>
<td>2243.74</td>
<td>8232.88</td>
</tr>
<tr>
<td>opm5</td>
<td>4795.68</td>
<td>2563.72</td>
</tr>
<tr>
<td>opm6</td>
<td>1212.87</td>
<td>2025.84</td>
</tr>
<tr>
<td>opm7</td>
<td>6676.54</td>
<td>8311.12</td>
</tr>
<tr>
<td>opm8</td>
<td>6381.39</td>
<td>8045.67</td>
</tr>
</tbody>
</table>
# Coverage metric

**Table:** G2PPLS-VNS × GMOVNS: Coverage

<table>
<thead>
<tr>
<th>Instance</th>
<th>Coverage C(G2PPLS-VNS,G-MOVNS)</th>
<th>Coverage C(G-MOVNS, G2PPLS-VNS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Average</td>
</tr>
<tr>
<td>opm1</td>
<td>1.00</td>
<td>0.79</td>
</tr>
<tr>
<td>opm2</td>
<td>1.00</td>
<td>0.80</td>
</tr>
<tr>
<td>opm3</td>
<td>1.00</td>
<td>0.65</td>
</tr>
<tr>
<td>opm4</td>
<td>0.95</td>
<td>0.57</td>
</tr>
<tr>
<td>opm5</td>
<td>0.73</td>
<td>0.38</td>
</tr>
<tr>
<td>opm6</td>
<td>0.96</td>
<td>0.79</td>
</tr>
<tr>
<td>opm7</td>
<td>0.90</td>
<td>0.34</td>
</tr>
<tr>
<td>opm8</td>
<td>0.90</td>
<td>0.41</td>
</tr>
</tbody>
</table>
G2PPLS-VNS as a mono-objective optimization tool

**Table:** Comparison of best results: G2PPLS-VNS × GGVNS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>opm1</th>
<th>opm2</th>
<th>opm3</th>
<th>opm4</th>
<th>opm5</th>
<th>opm6</th>
<th>opm7</th>
<th>opm8</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2PPLS-VNS</td>
<td>228.12</td>
<td>256.37</td>
<td>164046.32</td>
<td>164074.32</td>
<td>227.04</td>
<td>236.35</td>
<td>164018.81</td>
<td>164022.63</td>
</tr>
<tr>
<td>GGVNS</td>
<td>230.12</td>
<td>256.37</td>
<td>164039.12</td>
<td>164099.66</td>
<td>228.09</td>
<td>236.58</td>
<td>164021.28</td>
<td>164023.73</td>
</tr>
</tbody>
</table>
Obtaining the Reference Sets

- To obtain the Pareto Reference sets, 30 executions were performed for two hours. This test battery consisted in running GMOVNS and G2PPLS-VNS alternately.
- Reference sets and the solutions belonging to them are available at: http://www.decom.ufop.br/prof/marcone/projects/mining.html
Reference Set

Figure: Pareto Set Reference - opm2 instance - 271 non-dominated solutions
Conclusions

- The proposed multiobjective approach to OPMOP had been validated. The developed algorithms were able to generate good approximations to the Pareto fronts. For instance opm2 a reference set with 271 non-dominated solutions was found.
- The algorithms G2PPLS-VNS and GMOVNS, based on VNS procedures, performed better than algorithm GNSGAII-PR.
- The proposed algorithms seem suitable for practical applications.
• Federal University of Ouro Preto
• Brazilian agency FAPEMIG
• Brazilian agency CNPq
Thank you for your attention!
Questions?
References

Camilo de Lelis Dias Cavalcanti.  
Métodos de pesquisa e lavra ii, 2005.  

Colegio Web.  
Brazilian mining operations, November 2013.  

Prof. Welgeo.  
Comparison of a mining truck with a normal car, November 2013.  
G2PPLS-VNS as a mono-objective optimization tool

**Table:** GRASP–2PPLS × GRASP–MOVNS: Cardinalidade

<table>
<thead>
<tr>
<th>Instância</th>
<th>Ref</th>
<th>GRASP–2PPLS</th>
<th>GRASP–MOVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>D1</td>
</tr>
<tr>
<td>opm1</td>
<td>234</td>
<td>117</td>
<td>18</td>
</tr>
<tr>
<td>opm2</td>
<td>271</td>
<td>175</td>
<td>35</td>
</tr>
<tr>
<td>opm3</td>
<td>151</td>
<td>125</td>
<td>33</td>
</tr>
<tr>
<td>opm4</td>
<td>148</td>
<td>152</td>
<td>9</td>
</tr>
<tr>
<td>opm5</td>
<td>172</td>
<td>99</td>
<td>41</td>
</tr>
<tr>
<td>opm6</td>
<td>209</td>
<td>200</td>
<td>97</td>
</tr>
<tr>
<td>opm7</td>
<td>104</td>
<td>62</td>
<td>17</td>
</tr>
<tr>
<td>opm8</td>
<td>139</td>
<td>104</td>
<td>39</td>
</tr>
<tr>
<td><strong>Total de Soluções</strong></td>
<td></td>
<td>1428</td>
<td>1034</td>
</tr>
</tbody>
</table>