Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Discrete Mathematics 39 (2012) 233–240

Electronic Notes in DISCRETE MATHEMATICS

www.elsevier.com/locate/endm

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

V. N. Coelho [a,1,2] M. J. F. Souza [a,2] I. M. Coelho [b,2] F. G. Guimaraes [c,2] T. Lust [d,2] R. C. Cruz [a,2]

[a] *Federal University of Ouro Preto, Ouro Preto, MG, 35400-000, Brazil*

[b] *Fluminense Federal University, Niterói, RJ, 24210-240, Brazil*

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

[d] *LIP6-CNRS, UPMC. 4 Place Jussieu 75252 Paris Cedex 05, France*

Abstract

This work presents three multi-objective heuristic algorithms based on Two-phase Pareto Local Search with VNS (2PPLS-VNS), Multi-objective Variable Neighborhood Search (MOVNS) and Non-dominated Sorting Genetic Algorithm II (NSGA-II). 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 and spacing metrics. Computational experiments have shown the superiority of the algorithms based on VNS methods, which were able to find better sets of non-dominated solutions, more diversified and with an improved convergence.

Keywords: Open-pit-mining, Multi-objective optimization, Two-phase Pareto Local Search with VNS, Multi-objective Variable Neighborhood Search, NSGA-II

[1] The authors acknowledge UFOP, FAPEMIG and CNPq for supporting this research

[2] Email:vncoelho@gmail.com, marcone@iceb.ufop.br, imcoelho@ic.uff.br, fredericoguimaraes@ufmg.br, lust.thibaut@gmail.com and phaelcarlo@gmail.com

doi:10.1016/j.endm.2012.10.031

1 Introduction

This work deals with the open-pit mining operational planning problem (OPMOP). In this problem, there is a set of mining pits P, a set of trucks T and a set of load equipments L. It is necessary to blend ores from different mining pits to form a final product, taking into account the minimization of the three conflicting objectives: deviations of the production and quality goals, as well as the number of trucks necessary to the production process.

A solution for OPMOP is represented by a matrix $R = [Y|N]$, where Y is a matrix $|P| \times 1$ and N a matrix $|P| \times |T|$. Each cell y_p of the matrix $Y_{|P|\times 1}$ represents shovel $l \in L$ allocated to the pit $p \in P$. In the matrix $N_{|P|\times|T|}$, each cell n_{pt} represents the number of trips performed by the truck $t \in T$ to the pit $p \in P$. The dynamic truck allocation is considered, therefore trucks can be allocated to different pits after each discharge of material. This allocation system contributes to increasing fleet productivity and, therefore, to reduce the number of trucks needed to the process.

The OPMOP is an $\mathcal{NP}$-hard problem [9] and, for this reason, it is usually solved by heuristic algorithms. Unlike [9], who treated the problem through a mono-objective optimization algorithm, we develop here three multi-objective heuristic algorithms. The first one is based on Multi-objective Variable Neighborhood Search (MOVNS) inspired on previous works ([3], [10]) also due to the good results of VNS [4] in the mono-objective version of this problem [9]. The others are based on Two-phase Pareto Local Search with VNS (2PPLS-VNS) [5] and Non-dominated Sorting Genetic Algorithm II (NSGA-II) [2]. The three algorithms used the construction phase of Greedy Randomized Adaptive Search Procedures (GRASP) [7] to generate good initial populations.

In the multi-objective approach there is no single solution that satisfies all the objectives. What is sought is a set of non-dominated solutions, also called efficient solutions, or Pareto Front, being incumbent upon the decision maker the choice of the most suitable solution.

2 Proposed Algorithms

In this work, three multi-objective algorithms are proposed. The first one, called GMOVNS, combines ideas from Greedy Randomized Adaptive Search Procedure – GRASP [7] and Multi-objective Variable Neighborhood Search – MOVNS [3] procedures. The second one, called G2PPLS-VNS, combines GRASP and Two-phase Pareto Local Search with VNS – 2PPLS-VNS procedures [5]. The third one, so called GNSGAII-PR, combines the procedures GRASP and NSGA-II [2] with Path Relinking – `PR` [8] as a crossover operator.

The pseudo-code of the algorithm `GMOVNS` is outlined in Algorithm 1.

Algorithm 1 `GMOVNS`

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

1: $Xe \leftarrow$ BuildInitialSet($graspMax$)
2: $level \leftarrow 1$; $shaking \leftarrow 1$
3: **while** stop criterion not satisfied **do**
4:     Select a not "visited" solution $s \in Xe$ and check it as "visited"
5:     $s' \leftarrow s$
6:     **for** $i \leftarrow 1$ **until** $shaking$ **do**
7:         Select one neighborhood $N_k(.)$ at random
8:         $s' \leftarrow$ Shake(s', k)
9:     **end for**
10:     Let $k_{ult} \leftarrow k$; $changeLevel \leftarrow true$
11:     **for all** $s'' \in N_{k_{ult}}(s')$ **do**
12:         `addSolution`$(Xe, s'', f(s''), Added)$
13:         **if** Added = true **then**
14:             $changeLevel \leftarrow false$;
15:         **end if**
16:     **end for**
17:     **if** $changeLevel = true$ **then**
18:         $level \leftarrow level + 1$
19:     **else**
20:         $level \leftarrow 1$; $shaking \leftarrow 1$
21:     **end if**
22:     **if** $level \geq levelMax$ **then**
23:         $level \leftarrow 1$; $shaking \leftarrow shaking + 1$
24:     **end if**
25:     **if** all $s \in Xe$ are "visited" **then**
26:         check all $s \in Xe$ as "non-visited" solutions
27:     **end if**
28: **end while**
29: return Xe

A set of non-dominated initial solutions (line 1 of Algorithm 1) are generated by the construction phase of the `GRASP` procedure, as detailed in Algorithm 2. The steps of this algorithm were proposed by [9]. The `addSolution` procedure (line 5 of Algorithm 2) [5], adds the solutions created by the `GRASP` procedure to the efficient set X_e.

Algorithm 2 `BuildInitialSet`

Input: $graspMax$;
Output: Approximation of the efficient set Xe

```
for i ← 1 until graspMax do
    s_w ← BuildWasteSolution()
    Generate a random number γ ∈ [0, 1]
    s_i ← BuildWasteSolution(s_w, γ)
    addSolution(Xe, s_i, f(s_i))
end for
return Xe
```

As it can be seen, the `GMOVNS` algorithm has an iterative mechanism regulating the intensity of the perturbation phase. This strategy, proposed in this work, makes the algorithm perform more distant searches after *iterMax* iterations without any improvement in the current solution. For each unit of the variable *shaking*, a random movement is applied, among six previously developed neighborhoods: N^{NT}, N^{L}, N^{TT}, N^{TP}, N^{ST} and N^{SS} (line 7 of Algorithm 1). Line 20 yields the values of the variables *level* and *shaking* to one unit when at least one solution is added to the potentially efficient set Xe.

The `G2PPLS-VNS` (Algorithm 3) follows the same structure proposed by [5], in which `2PPLS` procedure is combined with an exchange neighborhood mechanism, mirrored in the `VNS` [4] method. In the first phase, a diversified initial set is generated (line 1 of Algorithm 3). In the second phase, the Pareto Local Search (`PLS`) [6] is applied to each individual of the population (line 2). The `PLS` can be regarded as a multi-objective generalization of the hill-climbing method.

Algorithm 3 `G2PPLS-VNS`

Input: $graspMax$; Neighborhoods $\mathcal{N}_k(x)$
Output: Approximation of the efficient set Xe

```
1: P_0 ← BuildInitialSet(graspMax);
2: Xe ← 2PPLS-VNS(P_0, N_k(x)); [5]
3: return Xe
```

Finally, the pseudo-code of `GNSGAII-PR` is outlined in Algorithm 4. As in `GMOVNS` and `G2PPLS-VNS`, the initial population P_0 is initialized by adding individuals generated by the `GRASP` procedure of [9] (lines 3 to 6 of Algorithm 4), although, in this case the stop criterion (line 2) is the size of the population P_0, input parameter of the algorithm.

Algorithm 4 `GNSGAII-PR`

Input: Population size N; Neighborhoods $N_k(x)$
Output: Approximation of the efficient set Xe

1: Initial population P_0
2: **while** $|P_0| \leq N$ **do**
3: $s_w \leftarrow$ BuildWasteSolution()
4: Generate a random number $\gamma \in [0, 1]$
5: $s_i \leftarrow$ BuildWasteSolution(s_w, γ)
6: $P_0 \leftarrow s_i$
7: **end while**
8: $Q_0 \leftarrow$ `SelectionPRCrossoverMutation`(P_0, Neighborhoods $N^{(k)}(.)$)
9: $Xe \leftarrow$ `NSGA-II`(P_0, Q_0, N, `SelectionPRCrossoverMutation`(.)) [2]
10: return Xe

In order to achieve the offspring population Q_0 (line 8 of Algorithm 4), the `SelectionPRCrossoverMutation` procedure is triggered. This procedure, which is described in Algorithm 5, enables the genetic operators of selection, crossover and mutation. The Path Relinking method was used as an advanced genetic operator, as in [8].

The `NSGA-II` procedure [2] is activated in line 9 of Algorithm 4. In this procedure, the steps of selection, crossover and mutation are replaced by the `SelectionPRCrossoverMutation` procedure.

Algorithm 5 `SelectionPRCrossoverMutation`

Input: *mutationRate*; *localSearchRate*
Input: Population P; Neighborhoods $N^{(k)}(.)$
Output: Offspring population Q

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

17: addSolution($Q, s', f(s')$)
18: **end if**
19: **end while**
20: return Q

Given two individuals s_1 and s_2, chosen randomly in the population, in line 3 of Algorithm 5 the Path Relinking procedure is applied in a bidirectional manner by exploring the two possible paths connecting the individuals s_1 and s_2. The best individual found s, evaluated by the mono-objective function of [9], is returned. The attributes considered in this strategy are the positions that the shovels hold in the guide solution.

In the line 8 of Algorithm 5 a random movement is applied to the individual s if the variable $ap_{mutation}$ is less than the parameter $mutationRate$. The same happens in the line 14, in which the VNS procedure is applied if a similar condition is satisfied. Finally, in the lines 4, 15 and 17, we check if the individuals s, s' and s'' should be added to the offsprings population Q.

3 Computational Experiments and Conclusions

The proposed algorithms were coded in C++ programming language with the computational framework OptFrame [1]. The algorithms were tested in a PC DELL XPS 8300 i7-2600, 3.4 GHz, with 16 GB of RAM, running Linux Ubuntu 10.10. The instances used for testing the algorithms were those of [9].

First, a comparison was made among the developed algorithms. 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).

Table 1 shows the average values obtained using the hypervolume and spacing metrics. According to the results, G2PPLS-VNS obtained the best average values in the two metrics used. For the spacing metric it was noted that the algorithm achieved a more uniform distribution in the objective space in relation to GMOVNS and GSGAII-PR algorithms. The average values of the hypervolume metric also indicates the superiority of G2PPLS-VNS, which obtained the bests volumes dominated by its approximations of the Pareto fronts.

Due to the superiority of algorithms based on the local search procedure VNS, the algorithms G2PPLS-VNS and GMOVNS were chosen for a further comparison. Table 2 shows the results between these two algorithms with respect to the Coverage metric.

Table 2 shows that G2PPLS-VNS algorithm was able to generate better sets than GMOVNS algorithm in seven instances. Furthermore, analysing the average

Table 1
G2PPLS-VNS × GMOVNS × GSGAII-PR: Spacing and Hypervolume

Instance	Spacing			Hypervolume (10^6)		
	G2PPLS-VNS	GMOVNS	GNSGAII-PR	G2PPLS-VNS	GMOVNS	GNSGAII-PR
opm1	4566.51	**2868.68**	6924.42	78.13	**78.23**	75.68
opm2	**1166.88**	2041.59	6557.20	75.15	**76.12**	73.15
opm3	**3824.96**	6188.99	14368.62	**71.17**	70.42	64.98
opm4	**2243.74**	8232.88	18295.27	**67.45**	66.94	59.75
opm5	4795.68	**2563.72**	7544.48	78.04	**78.67**	75.68
opm6	**1212.87**	2025.84	5963.77	**77.97**	77.23	74.32
opm7	**6676.54**	8311.12	14673.57	**75.28**	74.39	73.69
opm8	**6381.39**	8045.67	14874.73	**73.77**	73.56	73.22

Table 2
G2PPLS-VNS × GMOVNS: Coverage

Instance	Coverage					
	$\mathcal{C}$(G2PPLS-VNS, GMOVNS)			$\mathcal{C}$(GMOVNS, G2PPLS-VNS)		
	Best	Average	Std. dev.	Best	Average	Std. dev.
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

values, it was also able to obtain sets that covered GMOVNS in eight instances.

Finally, one last battery of tests aimed to verify if this new multi-objective approach could also find a good mono-objective solution. Table 3 shows the best mono-objective solutions obtained in 30 executions of G2PPLS-VNS compared to the results of the GGVNS algorithm [9]. As it can be seen, G2PPLS-VNS proved to be competitive with the literature mono-objective GGVNS algorithm, obtaining better or equal solutions in seven of eight instances.

Table 3
Comparison of best results: G2PPLS-VNS × GGVNS

	Instance							
Algorithm	opm1	opm2	opm3	opm4	opm5	opm6	opm7	opm8
G2PPLS-VNS	**228.12**	**256.37**	164046.32	**164074.32**	**227.04**	**236.35**	**164018.81**	**164022.63**
GGVNS	230.12	**256.37**	164039.12	164099.66	228.09	236.58	164021.28	164023.73

Finally, it is worth mentioning that the combination between G2PPLS-VNS and GMOVNS allows to generate good literature references sets, available at http://www.decom.ufop.br/prof/marcone/projects/mining.html.

References

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

[2] Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation **6** (2002), 182–196.

[3] Geiger, M. J., *Randomised Variable Neighbourhood Search for Multi Objective Optimisation*, Proc. of the 4th EUME Workshop Design and Evaluation of Advanced Hybrid MetaHeuristics (2004), 34–42.

[4] Hansen, P., N. Mladenovic, and J. A. M. Prez, *Variable neighborhood search: methods and applications*, 4OR: Quart. J. of the Belgian, French and Italian operations research societies **6** (2008), 319–360.

[5] Lust, T., J. Teghem, and D. Tuyttens, *Very Large-Scale Neighborhood Search for Solving Multiobjective Combinatorial Optimization Problems*. Evolutionary Multi-Criterion Optimization **207** (2010), 1041–1051.

[6] Paquete L., M. Chiarandini, and T. Stützle, *Pareto Local Optimum Sets in the Biobjective Traveling Salesman Problem: An Experimental Study*, Metaheuristics for Multiobjective Optimisation **535** (2004), 177–199.

[7] Resende, M. G. C., and C. Ribeiro, *Greedy randomized adaptive search procedures: Advances, hybridizations, and applications*, in Gendreau M. and Potvin J. Y. (Eds) "Handbook of Metaheuristics", Springer New York, 283–368, 2010.

[8] Ribeiro, C., and M. Resende, *Path-relinking intensification methods for stochastic local search algorithms*, Journal of Heuristics **18** (2012), 193–214.

[9] Souza, M. J. F., I. M. Coelho, S. Ribas, H. G. Santos, and 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.

[10] Schilde, M., K. Doerner, R. Hartl, and G. Kiechle, *Metaheuristics for the bi-objective orienteering problem*, Swarm Intelligence **3** (2009), 179–201.