

A comparative study of algorithms for solving the Multiobjective Open-Pit Mining Operational Planning Problems

Rafael Frederico Alexandre^{1,2,3}, Felipe Campelo¹,
Carlos M. Fonseca⁴, and João Antônio de Vasconcelos^{1,2}

¹ Graduate Program in Electrical Engineering, Federal University of Minas Gerais
Av. Antônio Carlos, 6627, Pampulha, 31270-901, Belo Horizonte, MG, Brazil

² Evolutionary Computation Laboratory, Federal University of Minas Gerais
Av. Antônio Carlos, 6627, Pampulha, 31270-901, Belo Horizonte, MG, Brazil

³ Department of Computer and Systems, Federal University of Ouro Preto
Rua 37, no 115, Loanda, 35931-008, João Monlevade, MG, Brazil

⁴ Department of Informatics Engineering, University of Coimbra
Pólo II, 3030-290 Coimbra, Portugal

Abstract. This work presents a comparison of results obtained by different methods for the Multiobjective Open-Pit Mining Operational Planning Problem, which consists of dynamically and efficiently allocating a fleet of trucks with the goal of maximizing the production while reducing the number of trucks in operation, subject to a set of constraints defined by a mathematical model. Three algorithms were used to tackle instances of this problem: NSGA-II, SPEA2 and an ILS-based multiobjective optimizer called MILS. An expert system for computational simulation of open pit mines was employed for evaluating solutions generated by the algorithms. These methods were compared in terms of the quality of the solution sets returned, measured in terms of hypervolume and empirical attainment function (EAF). The results are presented and discussed.

Keywords: Open pit mines, dispatch, multiobjective optimization, performance comparison

1 Introduction

The efficient use of available resources by companies is a requirement in any highly competitive market. For mining companies, using the fleet of trucks and shovels in the best possible way can enable a significant reduction in operational costs and a considerable improvement in productivity. According to Nel *et al.* [21] the cost of operating trucks and shovels in an open pit mine corresponds to between 50 to 60 percent of the total cost of operation. Moreover, trucks ranging from 100 to 240 tonnes of transport capacity usually cost from \$1.8 to \$4.7 million dollars, respectively [8]. Therefore, investment in efficient usage of available equipments can result in significant reductions in the total costs of a mining operation.

The solution to the problem of truck dispatching in open pit mines consists basically of answering the question: where each truck should go after leaving each place? Any answer must be provided with the aim of satisfying the needs of the mine using the available resources in the best possible way. Thus, the answer to this question must consider issues such as what should be produced, what is the expected quality, travel time to the next location, and even possible queues that may occur on the way to a given destination. When requesting a new dispatch, a truck moves to a pit, which must have a shovel compatible with that particular truck. The material removed from each pit has a certain quality that is associated with the proportion of chemical elements such as Iron, Silicon, Manganese, among others. If there is no queue at the place of loading, the truck is loaded and moves up to a crusher. Each crusher has quality requirements that the material produced must meet. Material that has no commercial value (that is, waste) is conducted by the trucks to mine sites reserved for storage of this type of material (rock piles).

The objectives of this work are twofold: first, to present a multiobjective model that defines, for a given fleet of trucks, a sequence of dispatches for the efficient use of equipment, minimizing the occurrence of queues and idle shovels. The proposed multiobjective model for the open-pit mining operational planning problem (OPMOPP) additionally includes the modeling of possible queues for truck loading operations as well as different speeds for loaded and empty trucks. The second objective is to propose and compare the performance of three approaches for the solution of the proposed model: two multiobjective evolutionary algorithms (MOEAs) and a metaheuristic based on the Pareto Iterated Local Search (PILS). A specific solution encoding and operators for generating candidate solutions are proposed for the evolutionary approaches, in order to generate feasible solutions given the operational constraints of the problem, therefore enabling a more effective search for the solution of this class of problems. The algorithms are compared using standard quality indicators: hypervolume and empirical attainment function (EAF).

2 Previous Works

The work of Doig and Kizil [8] studied the impact of the truck cycle time differences in mine productivity. The authors conclude in their work that the cycle time and the subutilization of the truck fleet impacts significantly on productivity in a mine, thus justifying the efficient use of available equipment. Additionally, roads in good condition for transportation were also found to be relevant. The work of Topal [26] asserts that proper planning of maintenance of trucks is essential to minimize its costs. That is, assuming availability of the entire fleet of trucks when looking for a solution may lead to oversensitive solutions, as units may be unavailable due to the preventive maintenance schedule. A case study of a large-scale gold mine showed a significant reduction (10%) of annual maintenance costs and more than 16% of overall reduction in maintenance costs

over 10 years of operation, in comparison with the baseline spreadsheet used in operation [25].

Tan *et al.* [24] presented a procedure for obtaining the optimal number of trucks in operation at the mine and also to estimate the capacity of the fleet. For the simulation of mine the software *Arena* [17] was employed. The data used for the simulations were collected using a GPS system and used weekly average values as the reference. Souza *et al.* [22] proposed a solution to an open-pit mining planning problem with dynamic truck allocation. The objective considered in their work was the minimization of the number of trucks used in the mine, and determination of the extraction rate at each pit to fulfill production and quality goals. They developed a heuristic called GGVNS, which combines ideas from both the Greedy Randomized Adaptive Search Procedure (GRASP) [9] and General Variable Neighborhood Search (GVNS) [19]. The GGVNS was successfully applied to solve the 8 distinct testing scenarios, with results validated using the commercial optimization software CPLEX [16]. More recent work presented three heuristics to solve the same problem, considering a multiobjective approach. Moreover, the work does not consider a possible queue to load and unload the trucks and also does not define the order of dispatches [3].

Subtil *et al.* [23] proposed a multi-stage approach for dynamic allocation of trucks in real environments for open pit mines. The proposed approach was validated through a simulation model based on discrete events. The authors report significant results using the algorithm, yielding increased production and also reduced operational delays of equipments. The work also states that, although the model is able to predict ore quality, this ability was not studied due to lack of relevant data for analysis. He *et al.* [14] sought to reduce the number of vehicles used in a mine by minimizing transportation costs and maintenance using GAs. Although satisfactory results were achieved, the model employed does not consider multiple constraints (compatibility between vehicles, production equipment and shovels, among others) found in dispatching problems in mines.

Given the many works in the literature, one realizes that they each have a different mathematical model and treat different objectives using techniques such as weighted sum of functions or goal programming. None of these works directly address the multiobjective nature of the problem by using multicriteria optimization techniques. Moreover, a large portion of these works aims at optimizing functions related to production, but fail to consider the quality of material produced or even operational constraints such as compatibility between shovels and trucks. In the next section we propose a multiobjective model to address these issues.

3 The Multiobjective Open-Pit Mining Operational Planning Problem

This section presents a new multiobjective mathematical model that includes two objectives: the first one is to maximize production at the mine, be it ore or waste rock. The second one is to minimize the number of trucks in operation.

However, as there are trucks with different capacities it is necessary to take their size into consideration. To facilitate understanding of the model, the parameters and variables are first presented. The parameters are defined by the test instances discussed in section 6.1. Let the parameters be:

- C is the set of crushers;
- O is the set of active ore pits;
- P is the set of pits formed by $O \cup W$;
- Q is the set of chemical elements of the ore;
- S is the set of shovels;
- T represents the set of trucks available;
- W is the set of active waste pits;
- Cap_t is the payload (in tonnes) of the truck t ;
- f_{ts} is a flag variable. 1, if truck t is compatible with shovel s and 0, otherwise.
- $Limc_p$ is the number of shovels that can be allocated to pit p ;
- Ql_{qc} is the lower limit of the amount of concentration (in percent) of the q^{th} chemical element to the crusher c ;
- Qu_{qc} is the upper limit of the amount of concentration (in percent) of the q^{th} chemical element to the crusher c ;
- q_{qo} is the content of chemical concentration (in percent) of the element q in the o^{th} pits of the ore;
- $y_{sp} \in \{0, 1\}$ is a flag variable. 1, if shovel s operates in pit p and 0, otherwise;
- y_{tp} is a flag variable. 1, if truck t can operate in pit p and 0, otherwise;

Let the variables be:

- $\bar{v} \in \{0, 1\}^{|T|}$ is the vector of optimization variables responsible for representing the availability of the trucks, with the t^{th} position of the vector (v_t) indicating whether the truck is in operation ($v_t = 1$) or not ($v_t = 0$);
- \widetilde{M} defines the sequence of dispatches received for each truck in operation inside the mine;
- x_o is the production (in tonnes) of the ore pit o ;
- x_w is the production (in tonnes) of the waste pit w ;
- x_{oc} is the production of the o^{th} ore pit, crusher c (in tonnes).

Next, the Eqs. (1)-(9) present the mathematical model for the problem under consideration. It is important to highlight at this point that x_o , x_w , and x_{oc} are calculated as a function of optimization variables \bar{v} and \widetilde{M} . The objective functions are given as:

$$\text{Maximize: } \sum_{\forall o \in O} x_o(\bar{v}, \widetilde{M}) + \sum_{\forall w \in W} x_w(\bar{v}, \widetilde{M}) \quad (1)$$

$$\text{Minimize: } \sum_{\forall t \in T} \bar{v}_t \times Cap_t \quad (2)$$

subject to a number of operational constraints, that define key aspects of the operating environment of a mine:

$$\frac{\sum_{\forall o \in O} q_{qo} x_{oc}(\bar{v}, \widetilde{M})}{\sum_{\forall o \in O} x_{oc}(\bar{v}, \widetilde{M})} \geq Ql_{qc}, \quad \forall q \in Q; c \in C \quad (3)$$

$$\frac{\sum_{\forall o \in O} q_{qo} x_{oc}(\bar{v}, \widetilde{M})}{\sum_{\forall o \in O} x_{oc}(\bar{v}, \widetilde{M})} \leq Qu_{qc}, \quad \forall q \in Q; c \in C \quad (4)$$

$$\sum_{\forall s \in S} y_{sp} \leq Limc_p, \quad \forall p \in P \quad (5)$$

$$\sum_{\forall p \in P} y_{sp} \leq 1, \quad \forall c \in C \quad (6)$$

$$y_{sp} + y_{tp} - 2f_{ts} = 0 \quad (7)$$

$$|C|, |S|, |P|, |Q|, |T| > 0 \quad (8)$$

$$Ql_{qc}, q_{qo}, h, u_t > 0, \quad \forall q \in Q; c \in C; t \in T; o \in O \quad (9)$$

The optimization variables \bar{v} and \widetilde{M} are discussed in detail in section 4.1. The constraints of the model represent the *limits of chemical quality deviation* (3)–(4); the *shovel allocation constraints* (5)–(6); the *compatibility between shovel and trucks constraint* (7); and the *ensures that the variables are greater than zero* (8)–(9).

4 Multiobjective Evolutionary Algorithms

The optimization problem presented in the previous section can be solved using evolutionary algorithms. Evolutionary algorithms (EAs) [5] represent a family of metaheuristics that perform an adaptive iterative sampling of the design space by means of a population of candidate solutions. EAs generally work by iteratively updating the current population to create a new population by means of four main operators: selection, crossover, mutation and elite-preservation. Evolutionary methods can be easily designed or adapted to solve multiobjective problems, with or without constraints [7]. Moreover, these algorithms are easily adjusted to handle a diversity of problem domains, which allows for their straightforward adaptation to the multiobjective OPMOPP.

In this work two algorithms were adapted to solve the multiobjective OPMOPP: the NSGA-II [6] and the SPEA2 [27]. A detailed description of these two algorithms can be found in the references, and will not be provided here.

In common with other EAs, successful multiobjective implementations require well-designed representation systems for individual problems and also genetic operators that are appropriate for the task. Recombination (crossover) operators can be particularly problematic. In the following sections a new representation to allow the dispatch fleet of trucks in a open pit mine is presented, together with operators to perform the crossover and mutation of candidate solutions coded according to this representation. Moreover, it is important to note that in this work the initial populations of all algorithms were randomly initialized. Additionally, binary tournament selection [5] was used in all cases.

4.1 Representation

The proposed codification initially builds a matrix \tilde{P} wherein each column j represents a location of the mine. For each location, a subset of the possible places to where a truck can be dispatched is defined. Therefore, each cell p_{ij} of the matrix indicates a possible target location for a truck that is in location p .

From the initial matrix \tilde{P} , the candidate solutions can be created without the need for additional information from the mine, ensuring that the constraint (7) is satisfied. For the generation of individuals it is necessary to inform the value of k which aims to define the number of rows (i) of the matrix \tilde{M} of the solution $s = [\bar{v}|\tilde{M}]$. The number of columns (j) is the same as in matrix \tilde{P} . For each cell of column j a random place p in column j of the matrix \tilde{P} is chosen. The vector $\bar{v} \in \{0, 1\}^{|\mathcal{T}|}$ is randomly constructed, indicating whether the truck is in operation ($v_t = 1$) or not ($v_t = 0$). With this structure, for each request for a new order by a truck in operation the candidate solution informs the next destination for that truck, considering the location of the truck at the time of the request.

4.2 Crossover Operator

The crossover operators proposed for this representation are based on cutting operators, as discussed in several studies of the literature [12] [4]. Cutoff crossing (1PX) considers two candidate solutions x' and x'' represented by matrices \tilde{M} of dimension $I \times J$. An integer cutoff value $c \in [1, J]$ is randomly drawn from a discrete uniform random variable, and a new candidate solution y' is generated by combining the first c columns from x' and the final $J - c$ columns from x'' . A second candidate solution y'' is also generated with the c first columns of x'' and the last $J - c$ columns from x' , as is the case of the usual 1-point vector crossover employed in the EAs. Vector \bar{v} uses binary crossover [5].

4.3 Mutation Operator

The mutation proposed for this representation is known as flip mutation [2]. In this case, each cell of the \tilde{M} of the solution s selected for mutation receives a new value obtained from the random matrix \tilde{P} . This operator is applied, with a

certain probability of occurrence p_m , to the candidate solutions generated by the crossover operator. For the vector \bar{v} , the bits are changed by turning individual trucks on or off.

5 Multiobjective Iterated Local Search

To provide a comparison baseline for the evolutionary approaches NSGA-II [6] and SPEA2 [27] using the operators defined in the previous section, and to evaluate the potential of the specific operators proposed for the multiobjective OPMOPP, we employ a method based on the Pareto Iterated Local Search (PILS) [11], which is an adaptation of the Iterated Local Search (ILS) [18] for multiobjective problems.

Algorithm 1: MULTIOBJECTIVE ITERATED LOCAL SEARCH (MILS)

```

Input:  $maxIter$ 
Input:  $maxCount$ 
Output:  $Front$ 
1  $Front \leftarrow makeInitialSolutions()$ 
2  $iter \leftarrow 1$ 
3 while  $iter \leq maxIter$  do
4    $s' \leftarrow selection(Front)$ 
5    $labeled(s')$ 
6    $count \leftarrow 1$ 
7   while  $count \leq maxCount$  do
8      $s'' \leftarrow perturbation(s')$ 
9      $s'' \leftarrow localSearch(s'')$ 
10     $inserted \leftarrow refresh(Front, s'')$ 
11    if  $inserted$  then
12       $count \leftarrow 1$ 
13       $s' \leftarrow s''$ 
14    else
15       $count \leftarrow count + 1$ 
16    end
17  end
18   $iter \leftarrow iter + 1$ 
19 end
20 return  $Front$ 

```

The operation of the MILS is illustrated in Algorithm 1. It starts by generating an initial population and extracting the nondominated set, which gets stored in the $Front$ set (line 1). After this initial step, the iterative cycle is started. For $maxIter$ iterations, a solution from $Front$ is selected and the iteration of the main algorithm (lines 7-17) is executed. In this step, the procedures of perturbation (line 8) and local search (line 9), similar to those existing in PILS, are

executed. The front set is updated (line 10) with the refined solution obtained after local search. If the solution generated after the procedures of perturbation and local search is nondominated, it is inserted into the set *Front*, and the *count* variable is reset (line 12). Otherwise, count is incremented by one (line 15). The *maxCount* variable indicating the maximum number of times the solution is operated without inserting a non-dominated solution in the set front.

The procedure defined as perturbation (line 8) is responsible for generating the solutions known as neighbors. For the problem addressed in this work, the neighboring solutions are constructed as follows: two random integers p_1 and p_2 are generated such that $0 \leq p_1 \leq (J - j_d)$ and $p_2 = p_1 + j_d$, where J represents the number of columns of the matrix \tilde{M} and j_d is the number of columns to be changed. All values in the interval $[p_1, p_2]$ of the matrix \tilde{M} are changed, creating a new solution.

The other procedure used by Algorithm 1 is responsible for performing a local search (line 9) with the objective of exploring neighboring regions of the search space. To accomplish this task we use an algorithm known as reduced VNS (RVNS) [13]. The RVNS is a simplified version of the Variable Neighbourhood Search (VNS), where the deterministic local search procedure (the most time-consuming part of VNS) is removed in order to reduce the computational cost. This algorithm receives the solution to be perturbed and uses the mutation operator (line 3) defined in this work. If the solution changed (s'') dominates the current solution (s'), it is replaced (line 4-5) and the variable *iter* is reset (line 6). The procedure for generating neighboring solutions is performed N times, where N is an input of the algorithm.

Algorithm 2: REDUCED VARIABLE NEIGHBOURHOOD SEARCH (RVNS)

Input: s'
Output: s'

```

1 iter  $\leftarrow$  1
2 while iter  $\leq$   $N$  do
3    $s'' \leftarrow \text{MakeNeighborhood}(s')$ 
4   if  $s'' \prec s'$  then
5      $s' \leftarrow s''$ 
6     iter  $\leftarrow$  1
7   else
8     iter  $\leftarrow$  iter + 1
9   end
10 end
11 return  $s'$ 

```

6 Experimental Setup

In this section we define the test problems and the experimental design of the computational experiments employed to verify the ability of the NSGA-II and SPEA2 heuristics to obtain a good set of tradeoff solutions for the multiobjective OPMOPP. This experiment has essentially two goals: to evaluate whether any of the algorithms will be able to find feasible, interesting solution tradeoffs for the multiobjective OPMOPP instances considered, and to check whether the algorithms will yield significantly different performances.

First we describe the test scenarios employed and the configurations of the algorithms. Afterwards the performance metrics and experimental design are provided.

6.1 Test problems

In this study we considered benchmark instances of problems based on those proposed by Souza *et al.* [22]⁵. Table 1 describes the main characteristics of the test instances. Columns **# Pits**, **# Shovels**, **# Trucks** and **# Par** indicate the number of pits, shovels, trucks and control parameters (chemical), respectively. The column **Details** provides the number and capacity (in case of trucks), or the productivity (in case of shovels). For example, the pair (15;50t) means there are 15 shovels (or trucks) of 50 tonnes of capacity (or maximum productivity). The difference between Mines 1 and 2 are the levels of quality of chemical elements.

Table 1. Test Instances.

Instance	Details						
	# Pits	# Shovels, capacity		# Trucks, capacity		# Par	
Mine1	8	(4,900t)	(2,1000t)	(2,1100t)	(15,56t)	(15,90t)	10
Mine2	8	(4,900t)	(2,1000t)	(2,1100t)	(15,56t)	(15,90t)	10
Mine3	7	(2,500t)	(2,400t)	(1,600t)	(30,56t)		5
		(1,800t)	(1,900t)				
		(2,400t)	(2,500t)	(1,600t)			
Mine4	10	(1,800t)	(1,900t)	(3,1000t)	(22,56t)	(7,90t)	5
			(3,2600t)				

6.2 Evaluation of the solutions

An expert simulation system, based on discrete events, was built to evaluate the solutions generated by the optimization algorithms. This system has an interface with these algorithms, in which candidate solutions are processed and returned by the simulator to the algorithms, including the values of the objectives and

⁵ The definitions of the test instances used can be retrieved online [1].

constraints. Dispatches for mining fronts consider the distance and the average speed of trucks to calculate the time required for the trucks reaching their destination. In addition, the simulator considers the possibility of queue occur when loading trucks. The load time of each truck depends on the productivity of the shovels and truck capacity. The trucks are then dispatched to the crusher or waste piles, according to the quality of material produced. The stopping criterion of the simulation is the operation time of the mine.. This simulator was built using a programming language Java JDK 1.7.

6.3 Algorithm Setup

All the experiments considered the following (arbitrarily set) parameters: Population size = 200; Maximum number of evaluations = 20,000; Crossover rate = 0.9; and Mutation rate = 0.4. The dispatch matrices (\widehat{M}) have $J=20$, i.e., twenty columns. The selection operator employed was the Binary Tournament [5]. Initial populations were generated randomly, and all trucks were considered as starting their operation in the crusher. The MILS used $maxIter = 100$ and $maxCount = 20$, and $N = 10$ for the RVND. All runs consider one hour of operation of the mine. All algorithms were coded in Java and compiled with JDK 1.7, and were tested in a PC Intel(R) Core(TM) i7-3632, 2.2 GHz, with 8 GB of RAM, running Windows 8.1.

6.4 Quality Indicators

Evolutionary multiobjective optimization techniques usually need to consider complementary goals, namely the acquisition of a set of tradeoff solutions that are at the same time near the true (oftentimes unknown) Pareto-optimal front, and to have this set evenly covering the whole extension of the Pareto-optimal front - dual objectives usually referred to as *convergence* and *diversity*. To consider this multi-criterion nature in the evaluation of multiobjective algorithms, regarding the convergence and diversity of the solutions, the following quality indicator is used in this work.

Hypervolume or S-Metric Proposed by Zitzler and Thiele [28], returns the hypervolume of the region covered between the points present in the frontier and a P_{ref} point. This point (P_{ref}) is used as a reference and is dominated by all solutions presented on this frontier. For each solution $i \in \mathcal{PF}$ is constructed a hyperrectangle (c_i) with reference to P_{ref} . The result of this metric can be calculated as:

$$HV(\mathcal{PF}) = \sum_{i \in \mathcal{PF}}^{|\mathcal{PF}|} v_i \quad (10)$$

where v_i provided by c_i . The higher the value of HV better the quality of the solution indicating that there was a better spread and also a better convergence

although the metric is more sensitive to convergence of solutions in relation to the real Pareto frontier. For all the test problems we considered a reference point 10% higher than the upper limits of the Pareto optimal frontier.

Empirical Attainment Function In the face of random Pareto-set approximations, unary quality indicators provide a convenient transformation from random sets to random variables. To prevent the transformation of sets of solutions in a unary indicator and allow at the same time, a statistical analysis of the set of solutions obtained by multiobjective algorithm was proposed calls Empirical Attainment Function (EAF) [10]. Furthermore, an analysis using EAF allows one to identify in which regions of the objective space one algorithm is better than another, and to visualize this difference. The attainment function gives the probability of a particular point in the objective space vector being attained by (dominated by or equal to) the outcome of a single run of an algorithm. This probability can be estimated from several runs of an algorithm, in order to calculate the EAF of an algorithm. The EAF from to is defined as:

$$\alpha_n(z) = \frac{1}{N} \cdot \sum_{i=1}^n b_i(z) \quad (11)$$

where $b_1(z), \dots, b_n(z)$ be n realizations of the attainment indicator $b_x(z)$, $z \in \mathbb{R}^d$. Then, the function defined as $\alpha_n : \mathbb{R}^d \rightarrow [0, 1]$.

In the case of bi-objective optimization problems, the empirical attainment function (EAF) is fast to compute, and its graphical representation provides more intuitive information about the distribution of the output of an algorithm than unary (or binary) quality indicators. A tool for graphical analysis of the EAF is proposed on the work of Ibáñez et al. [15].

6.5 Experimental Design

The algorithms NSGA-II, SPEA2, and MILS were applied for the solution of the four test instance on 33 independent runs, after which each quality metric described in the previous section was calculated. The experimental model used was a 2-way factorial design, with both the algorithms and instances as factors [20]. Since our main interest is on the effects of the algorithms, only their effects were analyzed.

We first assessed the convergence of the three algorithms used considering the hypervolume for the four scenarios considered. Figure 1 considers the average of these metrics. The estimated Pareto frontier of the problem was constructed assessing 10^6 solutions that aim to cover the search space of the problem.

The results presented by Figure 1 suggest that NSGAI and SPEA2 algorithms have similar behavior except for instance 2, wherein the NSGAI has a relatively better performance. Additionally, it is important to note that MILS has worse performance for all test instances.

Tables 2 to 5 shows the results obtained by comparing the algorithm used in the experiments for the four scenarios mine. The tests considered as null

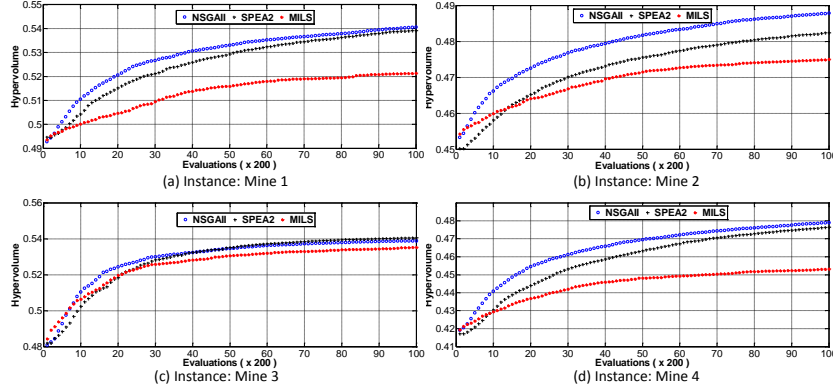


Fig. 1. Average hypervolume for the algorithms on each test case considered.

hypothesis (\mathcal{H}_0) that the two proposed algorithms have the same performance. Otherwise, there is a statistical difference between the algorithms. We consider first-and second tests similar to those proposed in order Fonseca et al. [10].

Table 2. Hypothesis test results for Mine 1 ($\alpha = .05$).

Optimiser	Hypothesis test	Test statistic	Critical value	p-value	Decision
MILS – NSGAII	1st-order EAF	0.696	0.454	0	Reject H_0
MILS – NSGAII	2nd-order EAF	0.848	0.575	0	Reject H_0
MILS – SPEA2	1st-order EAF	0.727	0.454	0	Reject H_0
MILS – SPEA2	2nd-order EAF	0.878	0.575	0	Reject H_0
NSGAII – SPEA2	1st-order EAF	0.424	0.454	> 0.05	Do not Reject H_0
NSGAII – SPEA2	2nd-order EAF	0.606	0.575	0.044	Reject H_0

Table 3. Hypothesis test results for Mine 2 ($\alpha = .05$).

Optimiser	Hypothesis test	Test statistic	Critical value	p-value	Decision
MILS – NSGAII	1st-order EAF	0.636	0.454	0	Reject H_0
MILS – NSGAII	2nd-order EAF	0.787	0.575	0	Reject H_0
MILS – SPEA2	1st-order EAF	0.727	0.454	0	Reject H_0
MILS – SPEA2	2nd-order EAF	0.606	0.575	0.044	Reject H_0
NSGAII – SPEA2	1st-order EAF	0.333	0.454	> 0.05	Do not Reject H_0
NSGAII – SPEA2	2nd-order EAF	0.454	0.575	> 0.05	Do not Reject H_0

Table 4. Hypothesis test results for Mine 3 ($\alpha = .05$).

Optimiser	Hypothesis test	Test statistic	Critical value	p-value	Decision
MILS – NSGAII	1st-order EAF	0.393	0.393	> 0.05	Do not Reject H_0
MILS – NSGAII	2nd-order EAF	0.575	0.484	0.004	Reject H_0
MILS – SPEA2	1st-order EAF	0.393	0.393	> 0.05	Do not Reject H_0
MILS – SPEA2	2nd-order EAF	0.636	0.484	0	Reject H_0
NSGAII – SPEA2	1st-order EAF	0.242	0.393	> 0.05	Do not Reject H_0
NSGAII – SPEA2	2nd-order EAF	0.333	0.484	> 0.05	Do not Reject H_0

Table 5. Hypothesis test results for Mine 4 ($\alpha = .05$).

Optimiser	Hypothesis test	Test statistic	Critical value	p-value	Decision
MILS – NSGAII	1st-order EAF	0.757	0.454	0	Reject H_0
MILS – NSGAII	2nd-order EAF	0.909	0.575	0	Reject H_0
MILS – SPEA2	1st-order EAF	0.727	0.454	0	Reject H_0
MILS – SPEA2	2nd-order EAF	0.909	0.575	0	Reject H_0
NSGAII – SPEA2	1st-order EAF	0.454	0.454	> 0.05	Do not Reject H_0
NSGAII – SPEA2	2nd-order EAF	0.575	0.575	> 0.05	Do not Reject H_0

Tables 2-5 show the comparisons between pairs of algorithms on each scenario, regarding the EAF indicator. The Optimiser column of the tables highlight the algorithms performed better when \mathcal{H}_0 was rejected. Overall, these results suggest that NSGAII and SPEA2 algorithms perform better when compared with MILS algorithm. The comparison between the NSGAII and SPEA2 algorithms does not allow to identify statistical differences between them except for Mine 1 (Table 2).

7 Conclusions

This work presented the definition of a multiobjective formulation for the open-pit mining operational planning problem. This model considers as objectives the maximization of production (ore and waste) and the minimization of the number of trucks in operation. An innovative representation of candidate solutions was proposed and employed by three multiobjective optimization methods: SPEA2, NSGA-II, and MILS. The proposed encoding enables the use of algorithms for heterogeneous fleets and also ensures that the solutions created are operationally feasible.

An experiment to compare the algorithms in terms of hypervolume and empirical attainment function values was performed. The results suggest that NSGA-II and SPEA2 algorithms have a better performance when compared with MILS for the problems considered, with the NSGA-II being marginally better than the SPEA2. As future work, we intend to evaluate the idleness of trucks and

shovels. Moreover, the mathematical model can be extended to consider other variables, such as, operating conditions of the mine.

Acknowledgments

This work has been supported by the Brazilian agencies National Council for Research and Development (CNPq, grants 475763/2012-2 and 306022/2013-3), Research Foundation of the State of Minas Gerais (FAPEMIG, grant CEX APQ-04611-10), and Coordination for the Improvement of Higher Education Personnel (CAPES, grant 012322/2013-00).

References

1. Alexandre, R., Vasconcelos, J., Campelo, F.: Additional electronic files. <http://cpdee.ufmg.br/~fcampelo/files/MOPMOPP/> (2014)
2. Chicano, F., Alba, E.: Exact computation of the expectation curves of the bit-flip mutation using landscapes theory. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'11. pp. 2027–2034. Dublin, Ireland (July 2011)
3. Coelho, V., Souza, M., Coelho, I., Guimarães, F., Lust, T., Cruz, R.C.: Multi-objective approaches for the open-pit mining operational planning problem. *Electronic Notes in Discrete Mathematics* 39(0), 233–240 (2012)
4. Coello, C., Lamont, G., Veldhuizen, D.: Evolutionary multi-objective optimization: A historical view of the field. *Computational Intelligence Magazine, IEEE* 1(1), 28–36 (2006)
5. Coello, C., Lamont, G., Veldhuizen, D.: *Evolutionary Algorithms for Solving Multi-Objective Problem*. Springer, 2nd edn. (2007)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE* 6(2), 182–187 (2002)
7. Dias, A., Vasconcelos, J.: Multiobjective genetic algorithms applied to solve optimization problems. *IEEE Transactions on Magnetics* 38(2), 1133–1136 (2001)
8. Doig, P., Kizil, M.: Improvements in truck requirement estimations using detailed haulage analysis. In: 3th Coal Operators Conference. pp. 368–375. The Australasian Institute of Mining and Metallurgy and Mine Managers Association of Australia (February 2013)
9. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6(2), 109–133 (1995)
10. Fonseca, C., Fonseca, V., Paquete, L.: Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, vol. 3410, pp. 250–264. Springer Berlin Heidelberg (2005)
11. Geiger, M.: The PILS metaheuristic and its application to multi-objective machine scheduling. In: Kfer, K.H., Rommelfanger, H., Tammer, C., Winkler, K. (eds.) *Multicriteria Decision Making and Fuzzy Systems Theory, Methods and Applications*. pp. 43–58. Shaker Verlag, Industriemathematik und Angewandte Mathematik (2006)

12. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1st edn. (1989)
13. Hansen, P., Mladenovic, N., Pérez, J.M.: Variable neighbourhood search: methods and applications. *4OR* 6(4), 319–360 (2008)
14. He, M., Wei, J., Lu, X., Huang, B.: The genetic algorithm for truck dispatching problems in surface mine. *Information Technology Journal* 9, 710–714 (2010)
15. Ibáñez, M., Stützle, T., Paquete, L.: Graphical tools for the analysis of bi-objective optimization algorithms. In: Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation. pp. 1959–1962. GECCO '10, ACM, New York, NY, USA (2010)
16. ILOG: Users Manual. IBM (2008)
17. Kelton, W., Sadowski, R., Sturrock, D.: Simulation with Arena. McGraw-Hill series in industrial engineering and management science, McGraw-Hill Higher Education, Boston, 4. ed., internat. ed. edn. (2007)
18. Lourenço, H., Martin, O., Stützle, T.: Iterated local search. *ArXiv Mathematics e-prints*. (feb 2001), arXiv:math/0102188
19. Mladenovic, N., Hansen, P.: Variable neighborhood search. *Computers & Operations Research* 24(11), 1097–1100 (1997)
20. Montgomery, D.: Design and Analysis of Experiments. Wiley, 7th edn. (2008)
21. Nel, S., Kizil, M., Knights, P.: Improving truck-shovel matching. In: 35TH APCOM Symposium. pp. 381–391. The Australasian Institute of Mining and Metallurgy, Wollongong, NSW (September 2011)
22. Souza, M., Coelho, I., Ribas, S., Santos, H., Merschmann, L.: A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research* 207(2), 1041–1051 (2010)
23. Subtil, R., Silva, D., Alves, J.: A practical approach to truck dispatch for open pit. 35th International Symposium on Application of Computers in the Minerals Industry (35th APCOM) pp. 765–777 (2011)
24. Tan, Y., Chinbat, U., Miwa, K., Takakuwa, S.: Operation modeling and analysis of open pit copper mining using GPS tracking data. In: Proceedings of the 2012 Winter Simulation Conference. pp. 1–12. ieee, Berlin, German (2012)
25. Topal, E., , Ramazan, S.: A new MIP model for mine equipment scheduling by minimizing maintenance cost. *European Journal of Operational Research* 207(2), 1065–1071 (2010)
26. Topal, E., , Ramazan, S.: Mining truck scheduling with stochastic maintenance cost. *Journal of Coal Science and Engineering (China)* 18(3), 313–319 (2012)
27. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Giannakoglou, K., et al. (eds.) *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*. pp. 95–100. International Center for Numerical Methods in Engineering (CIMNE) (2002)
28. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4), 257–271 (1999)