

# MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS FOR THE TRUCK DISPATCH PROBLEM IN OPEN-PIT MINING OPERATIONS

**Rafael F. Alexandre** 

Department of Computers and Systems, Federal University of Ouro Preto  
João Monlevade 35931-008, Brazil  
rfalexandre@ufop.edu.br

**Felipe Campelo** 

Department of Computer Science, Aston University, Birmingham, UK  
and  
Department of Electrical Engineering, Universidade Federal de Minas Gerais, Brazil  
f.campelo@aston.ac.uk

**João A. Vasconcelos** 

Department of Electrical Engineering, Universidade Federal de Minas Gerais  
Belo Horizonte 31270-901, Brazil  
jvasconcelos@ufmg.br

**Abstract** – This work is concerned with the efficient allocation of trucks to shovels in operation at open-pit mines. As this problem involves high-value assets, namely mining trucks and shovels, any improvement obtained in terms of operational efficiency can result in considerable financial savings. Thus, this work presents multi-objective strategies for solving the problem of dynamically allocating a heterogeneous fleet of trucks in an open-pit mining operation, aiming at maximizing production and minimizing costs, subject to a set of operational and physical constraints. Two Multi-objective Genetic Algorithms (MOGAs) were specially developed to address this problem: the first uses specialized crossover and mutation operators, while the second employs Path-Relinking as its main variation engine. Four test instances were constructed based on real open-pit mining scenarios, and used to validate the proposed methods. The two MOGAs were compared to each other and against a Greedy Heuristic (GH), suggesting of the MOGAs as a potential strategy for solving the multi-objective truck dispatch problem for open-pit mining operations.

**Keywords** – Multi-Objective Optimization; Evolutionary Algorithms; Open-Pit Mines; Truck Dispatch; Path-Relinking; NSGA-II.

## 1. INTRODUCTION

The investment of mining companies in equipment is a central factor for this industry, mainly due to the high cost of the required machinery, usually of the order of hundreds of millions of dollars, up to billions of dollars for large companies that operate several mines [1, 2]. It is therefore essential to ensure that all equipment available for the operation of a mine is used efficiently.

To this end, the allocation of trucks in a mine is of great importance, as it can result in substantial cost reductions. Trucks operating in the mining industry are among the largest in the world, with capacities up to 400 metric tons, and represent a significant acquisition cost (US\$ 3-5 millions per unit). Additionally, the shovels used in mines must also be used in an efficient manner, to avoid long idle intervals. In recent decades haulage cost savings have been mainly achieved by economies of scale, i.e., by increasing the capacities of trucks and shovels [2]. However, it seems that the limit capabilities of these resources have been reached, and mining companies are now looking for the most efficient use of their equipment, particularly when it comes to truck allocation and routing. As a consequence, researchers have been working towards higher levels of efficiency and effectiveness in the allocation of trucks, which results in an overall improvements in fleet productivity [3–5].

To reduce the risk of not meeting ore demands due to uncertainties, the operational planning of mines usually adds a greater number of trucks than required for operation. Oftentimes this approach leads to an inefficient use of trucks, mainly due to loading queues. On the other hand, operations with fewer trucks can result in shovel idleness or underproduction. Truck allocation models are therefore necessary for determining and justifying an adequate budget for the equipment fleet.

Given a fixed equipment fleet, the operation of a mine consists of allocating trucks to satisfy stakeholder demands. Furthermore, it is important to meet operational constraints, such as compatibility between equipments and ore quality. Figure 1 shows the main components of this operation. A cycle of operation for a given (initially idle) truck (1) consists of: i) dispatch request; ii) moving empty (2); iii) waiting for loading (3); iv) loading (4); v) moving loaded (6); and vi) dumping (7).

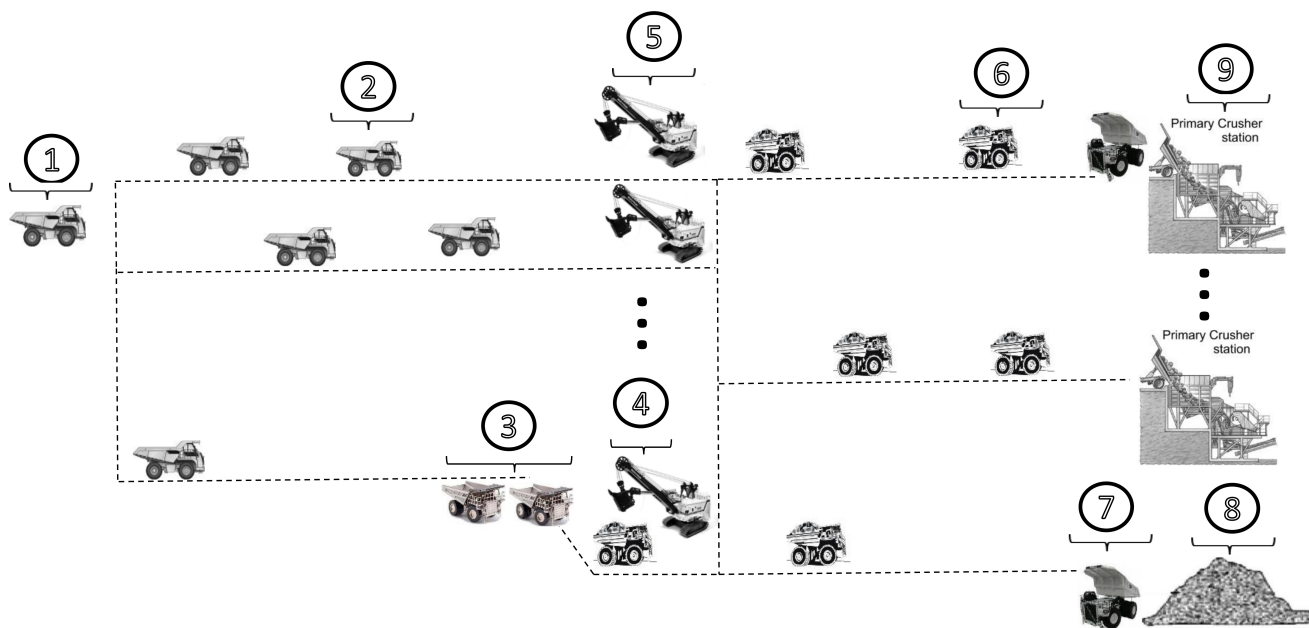


Figure 1: A simple scenario of an open-pit mine and its main features. Circled numbers indicate possible states of the trucks in operation at the mine. The vertical ellipsis indicate the possibility of many shovels as well as several crushers. Dashed lines indicate possible destination locations for trucks. The loading of trucks is carried out by shovels identified as 5. Loaded trucks are dispatched to crushers (identified as 9) or waste repositories (identified as 8).

Figure 1 also shows the shovels (5), waste dumps (8) and crushers (9). Each shovel has a given productivity (in *ton/h*) and is located at a pit. Each pit contains ore with a specific chemical concentration (e.g., %Fe, %SiO<sub>2</sub> and %P). Trucks in operation have sizes that must be compatible with the size of the shovels. Crushers receive ore extracted from the pits, and have minimum and maximum limits to the chemical concentrations of the ore, which must be met during operation. To produce ore which satisfies the chemical concentration constraints, a mixture of ores from different pits is usually required. Simultaneously, to reduce operating costs, queues of trucks and idleness of shovels must be avoided.

This work is focused on two objectives: (i) to minimize the operational costs; and (ii) to maximize the transportation (production) of ore and / or waste, which is a proxy for the mine's productivity. The cost of operation is directly related to the number and capacity of trucks operating in the mine, which on the other hand tend to correlate with the total production of the mine. This leads to a tradeoff scenario, since it is simultaneously desired to reduce the costs of operation (which can involve reducing the number of trucks in operation) and to increase the production, which lends itself to a multi-objective optimisation formulation.

In this paper we present:

- A mathematical model of the constrained multiobjective problem, defined by the minimization of the cost of trucks in operation and maximization of mine production, subject to ore quality and operational constraints, focusing on the truck dispatching problem [5].
- Two multi-objective optimization metaheuristics to solve this problem formulation.
- A comparative evaluation of the proposed methods against a greedy heuristic on four real instances of open-pit mines located in the state of Minas Gerais, Brazil.

In addition, this work also describes the simulator that was used to evaluate the solutions proposed by the optimization algorithms. The remainder of this paper is organized as follows: Section 2 provides an overview of relevant works; Section 3 introduces the main characteristics of the simulation system used to evaluate solutions; and Section 4 presents the mathematical formulation for the problem. The proposed algorithms for the solution of this problem are detailed in Section 5. Section 6 describes the experimental design and the results obtained through the computational tests. Finally, Section 7 presents conclusions and final remarks.

## 2. RELATED WORKS

The problem of truck dispatching is not exclusive to the mining sector, with several industries that use transportation fleets dealing with similar problems. Although this paper focuses on a fleet of trucks for operation in open-pit mines, the methodology developed here can in principle be adapted for any application that involves large fleets of trucks requiring optimal allocation, such as the operational planning of city waste collection. Newman *et al.* [6] present a literature review with emphasis on recent

works in mine planning. Several techniques that address the dispatch of trucks in mines are discussed, which suggests a strong interest by the scientific community in solving problems related to this issue. Techniques for truck dispatching in open-pit mines can be classified in two groups: i) *fixed allocation*, where the truck operates on the same route during the entire operation; and ii) *dynamic allocation*, in which the best route is assigned to the truck on each dispatch [7]. This paper focuses on the latter, as it commonly yields better results [3].

Studies available in the literature model the problem of dispatching of trucks emphasizing different aspects of the problem. Some focus on defining the optimal fleet size for a given mine [2, 5, 8], and afterwards define the routes for each truck. However, the order in which the trips must occur is not defined. Moreover, not all works consider the quality of the ore produced as part of their models [4, 8–10].

Some studies work exclusively with homogeneous fleets [10, 11], i.e., do not consider the need for compatibility constraints between trucks and shovels. As large mines generally involve heterogeneous truck fleets [2, 4, 5, 7–9], specialist algorithms for truck dispatching must consider equipment size to perform the correct allocation.

Most studies employ single-objective approaches to solve the truck dispatch problem, with mathematical models that generally seek to maximize production. Some use deterministic heuristics to solve the optimization problem, as discussed by Alarie and Gamache [3]. Others apply non-deterministic heuristics to solve the problem of identifying the number of trips that each truck must perform. For instance, Souza *et al.* [5] present a single-objective hybrid algorithm that combines characteristics of two meta-heuristics: Greedy Randomized Adaptive Search Procedure (GRASP) and General Variable Neighborhood Search (GVNS). Lin *et al.* [11] use a Genetic Algorithm (GA) to solve a problem of managing truck fleets in earthmoving operations, a similar problem to the truck dispatch problem in open-pit mines. Bastos *et al.* [4] present an algorithm that uses probability functions to select the destination of the truck and the probability value can vary over time. The algorithm is compared with both a greedy heuristic and another method called Minimization of Truck Cycle Times (MTCT).

Some recent works address the problem using multi-objective approaches, to simultaneously optimize several performance criteria. For instance, Mendes *et al.* [9] present a Multi-objective Genetic Algorithm to solve the problem of truck dispatching in open-pit mines. The authors considered two objectives to be minimized, namely the distance travelled and the queue time of transport equipments. While interesting, the algorithm proposed in that work required the application of a repair operator to correct infeasible solutions, which resulted in additional computational costs and, possibly, in reduced efficacy of the search mechanisms of the optimization approach.

A direct comparison between the problem formulation used in this work and other similar problems found in the literature, reported in this section, is presented in a consolidated form in Table 1 of Section 4.3.

### 3. SIMULATION SYSTEM

Most studies dealing with operational planning for open-pit mines employ simulation techniques for evaluating solutions. Some use simulation software such as ARENA [2], Truck Dispatch Simulator [9], CYCLONE [11] or SimEvents [4]. Others have developed specific simulation software [8]. Despite being widely used by researchers, these approaches either do not allow control of all problem variables, or are not easily integrated with other computer systems. Thus, a system developed for simulating the operation of a mine, with a well-defined interface for optimization routines, is desired.

To this end, an expert system, based on discrete event simulation, was built<sup>1</sup> to evaluate the candidate solutions generated by the optimization algorithms. This system is similar to existing ones [2, 9, 12]. An interesting aspect of an open-pit mining truck dispatch simulator is that it can reproduce the mine characteristics as it executes the vehicle dispatch plan [13]. The simulation system developed has an interface that receives the mining scenario to be simulated, the set of trucks in operation, and the sequence of dispatches that will be received by each truck. Based on the scenario – which defines the mine characteristics, such as ore and waste pits, shovels, distances, shovel productivity, among others – the truck dispatches are processed and calculated during a given operating time, and returned at the end of the simulation, together with the variables of interest for decision making and evaluation of constraints. Returned values include the volume of mineral (ore or waste) produced, queue times, ore quality, and distance travelled by each truck, which are then used to calculate the relevant quantities for the optimization problem.

Truck dispatches consider the distance between sites of the mine and truck speeds to calculate the time required for them to reach their destinations. The simulator also considers the possibility of queues as trucks arrive at a pit. The loading time of each truck depends on its own capacity, and on the productivity of shovels. As mentioned in Section 1, loaded trucks are dispatched to crusher or waste piles, according to the dispatch defined by the optimization algorithm. The stopping criterion of the simulation is the operation time set by the user.

Unlike Tan and Takakuwa's approach [12], which works specifically with Virtual Basic for Applications (VBA), the simulator developed for this work has a well defined interface that allows communication with optimization methods implemented in other programming languages, therefore allowing the test of specialist algorithms for the solution of the optimization problem.

The simulator receives as an input parameter an XML file that defines the characteristics of the mine to be simulated. This file contains the following information: (i) distances between mining pits; (ii) quality characteristics of the ore; (iii) trucks available for operation, their respective average speeds and capacities; (iv) shovels allocated to each mining front, their respective productivities and compatibility with the trucks in operation; and v) crushers, with their expected productivity and desired quality characteristics during the operation.

<sup>1</sup>Using Java JDK version 1.7.

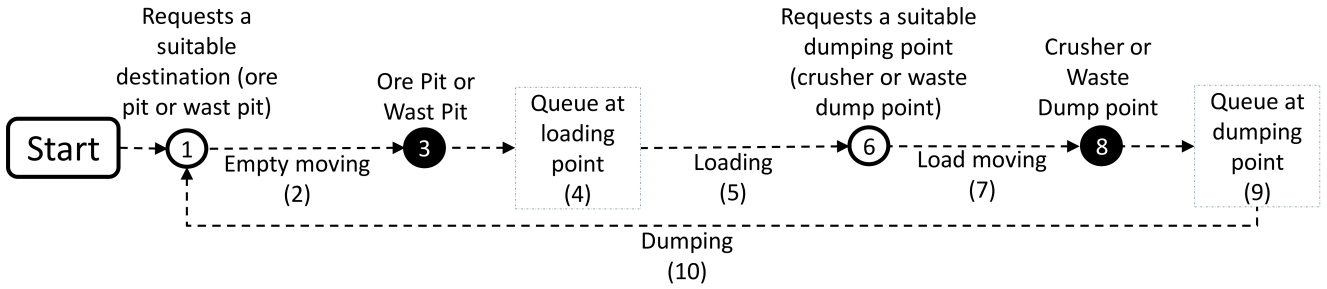


Figure 2: A typical truck loading and haulage cycle in an open-pit mining operation, as simulated in this work. The work cycle of the trucks starts the moment it receives the dispatch (1). The truck moves to the pit (3), is loaded and requests the dispatch to the unloading location (6). It moves to the crusher or waste dump, discharges the material (ore or waste) and returns to the beginning of the cycle.

After loading the mine definitions, the simulation system works according to a sequence of discrete events that change the status of the equipments in operation. These events occur as illustrated in Figure 2. Each cycle of a truck in operation is initiated by receiving a target location (ore or waste pit) to carry out its loading. The truck then starts a new route in which it is moving empty. The duration of this route is calculated based on the average speed of the truck on that specific route, and on the distance between its current location and target destination. Upon arriving at the destination, the truck may need to wait a certain amount of time if there are other trucks in the same location (queue at loading point). Waiting queues consider the order of arrival of the trucks, without other priority definitions. The loading time of the truck is calculated based on the productivity of the shovel in operation (tonnes/hour) and the capacity of the truck. After the shovel finishes loading the truck, a new target destination is defined - crusher or waste dump - and the truck starts moving loaded towards it. The truck must be dispatched to a crusher if it is loaded with ore, and to a waste dump otherwise, which must be ensured by the dispatch algorithm. After arriving at its destination, the truck can again be subject to a waiting queue until the transported material can be unloaded. Finally, the material is dumped and the truck cycle is terminated by returning to step (1) of Figure 2, freeing the truck to start of a new cycle.

It is clear that a correct definition of the number of trucks in operation, as well as of the dispatch sequences given to each truck, may result in several improvements to the efficiency of the mine, such as: i) increasing productivity due to reducing both queues and shovel idleness; and ii) guaranteeing the quantity of material removed from each pit, in compliance with the operating constraints. In view of these considerations, the next Section will present a mathematical formulation of the problem as a multi-objective optimization task, and Section 5 will describe the proposed algorithms for solving this problem, which must return efficient sequences for each truck in operation to execute.

## 4. THE MULTI-OBJECTIVE TRUCK DISPATCH PROBLEM IN OPEN-PIT MINING OPERATIONS

### 4.1 Optimization Variables

In this work the dispatch of trucks in an open-pit mining operation considers a heterogeneous fleet of trucks  $\mathbb{V}$ , such that each truck  $v_i \in \mathbb{V}$ ,  $i = \{1, \dots, |\mathbb{V}|\}$  has, associated to it, a transport capacity and an average speed. Each truck in operation must perform a set of cycles during the simulation, with the time of each cycle as well as the production of the truck depending on the dispatches received by it. Solving the truck dispatch problem means defining the fleet of trucks that must operate and the sequence of dispatches for each truck.

The variables that define the truck fleet as well as the sequence of dispatches received are presented below. A candidate solution for the truck dispatch problem is represented by matrix  $S = [v|D]$ , where  $v \in \{0, 1\}^{|\mathbb{V}|}$  is a binary vector where each position represents the availability (1) or not (0) of a truck; and  $D$  is a  $|\mathbb{V}| \times m$  matrix, with  $m$  representing the number of dispatches. Each cell  $d_{ij}$  of  $D$  is composed of a tuple  $(\alpha, \beta)$ , which represents the identifiers of the next two destinations of the truck (loading and discharge positions). Each place in the mine has a unique identifier. Thus, being  $\alpha$  the next destination of the truck, the route to be travelled is identified by its current place and the  $\alpha$  destination ( $route(currentPlace, \alpha)$ ). Furthermore, it is necessary to ensure that (i) if  $\alpha$  is a ore pit,  $\beta$  must be a crusher, and (ii) if  $\alpha$  is a waste pit,  $\beta$  must be a waste dump.

Figure 3 provides an example. From the scenario shown on the left portion of this figure, a hypothetical solution  $S^1$  was created, which has 5 trucks (3 active and 2 inactive) and a set of 4 dispatches for each truck ( $m = 4$ ).

For understanding the structure of  $S^1$ , let us consider the third truck (row 3), which is available for operation (a value of one in the leftmost column). Suppose that this truck is currently at site 5. So, it must move from site 5 to site  $\alpha = 2$ , taking the route  $5 \rightarrow 2$ . The sequence of routes taken by this truck is given in the third row of the matrix:  $\{5 \rightarrow 2, 2 \rightarrow 6, 6 \rightarrow 3, 3 \rightarrow 7, \dots, 1 \rightarrow 6\}$ .

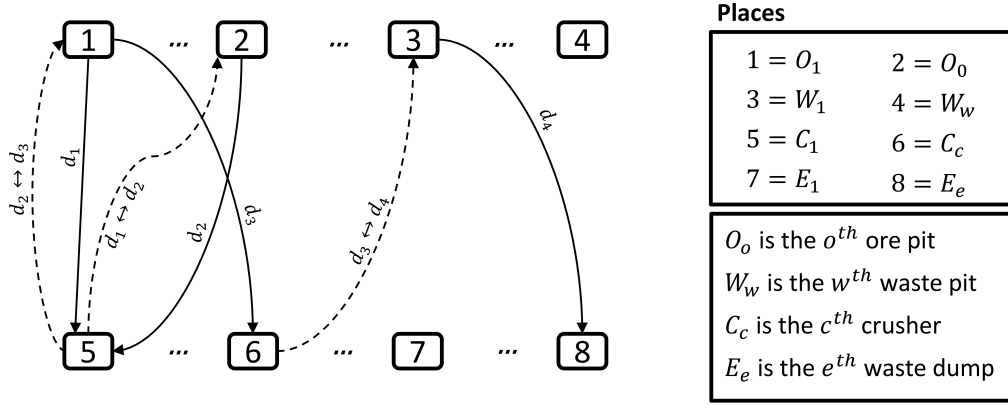


Figure 3: Hypothetical mine scenario and truck routes. At the top we show loading and discharge sites at a hypothetical mine. Sites 1 and 2 represent ore pits, 3 and 4 are waste pits, 5 and 6 are crushers, and 7 and 8 are waste dumps. At the bottom we illustrate an example candidate solution for this particular scenario, composed of 5 (possible) trucks and 4 dispatches for each truck. Arrows in the top graph indicate the routes of Truck 1 defined by the example solution, with solid lines indicating “moving loaded” routes, while dashed lines indicate “moving empty” ones.

#### 4.2 Mathematical Model

The mathematical formulation for this constrained optimization problem is expressed in its condensed form by equations (1) - (3), where  $\vec{F}$  is the vector of objective functions,  $g_i(S)$ ,  $i = 1, \dots, 6$  are inequality constraints, and  $h(S)$  is an equality constraint.

$$\begin{aligned} \text{Minimize: } \vec{F}(S) &= [f_1(S); -f_2(S)] & (1) \\ \text{subject to: } g_i(S) &\leq 0; \quad i = 1, \dots, 6 & (2) \\ h(S) &= 0 & (3) \end{aligned}$$

Before presenting the complete definition of the functions in equations (1) - (3), the following definitions are provided:

- $\mathbb{C}$ : set of crushers;
- $\mathbb{E}$ : set of waste dumps;
- $\mathbb{O}$ : set of active ore pits;
- $\mathbb{W}$ : set of active waste pits;
- $\mathbb{P}$ : set of pits formed by  $\mathbb{O} \cup \mathbb{W}$ ;
- $\mathbb{Q}$  set of chemical elements of the ore;
- $\mathbb{A}$ : set of shovels;
- $\mathbb{V}$ : set of trucks;
- $c_a^l$ : lower limit production (in tonnes) of  $a^{th}$  shovel;
- $c_a^u$ : upper limit production (in tonnes) of  $a^{th}$  shovel;
- $q_{qc}^l$ : lower limit for the concentration (%) of the  $q^{th}$  chemical element in the  $c^{th}$  crusher;
- $q_{qc}^u$ : upper limit for the concentration (%) of the  $q^{th}$  chemical element in the  $c^{th}$  crusher;
- $q_{qo}$ : average chemical concentration (%) of  $q^{th}$  element in the  $o^{th}$  ore pit;
- $row^l$ : minimum ore/waste ratio;
- $row^u$ : maximum ore/waste ratio;

- $s_{vj}$  : element ( $v$ -th row,  $j$ -th column) of  $S$ ;
- $u_v$ : operation cost of the  $v^{th}$  truck;
- $x_{oc}$ : ore received in  $c^{th}$  crusher from the  $o^{th}$  pit (in tonnes);
- $x_{we}$ : waste received in  $e^{th}$  waste dump from the  $w^{th}$  pit (in tonnes);
- $x_p$ : production (in tonnes) of the  $p^{th}$  pit;
- $y_{ap} \in \{0, 1\}$ : binary flag indicating whether the  $a^{th}$  shovel operates in the  $p^{th}$  pit;
- $y_{vp} \in \{0, 1\}$ : binary flag indicating whether the  $v^{th}$  truck can operate in the  $p^{th}$  pit;
- $y_{va} \in \{0, 1\}$ : binary flag indicating whether the  $v^{th}$  truck is compatible with the  $a^{th}$  shovel.

Considering these definitions, the objectives are formulated as: i) minimization of the total cost of the trucks in operation, considering a heterogeneous fleet defined by equation (4); and ii) maximization of the production, calculated as the total supply of ore to the crushers or sterile to waste dumps defined by equation (5).

$$f_1(S) = \sum_{\forall v \in \mathbb{V}} s_{v1} u_v \quad (4)$$

$$f_2(S) = \sum_{\forall o \in \mathbb{O}} \sum_{\forall c \in \mathbb{C}} x_{oc} + \sum_{\forall w \in \mathbb{W}} \sum_{\forall e \in \mathbb{E}} x_{we} \quad (5)$$

It should be noted that the volume of mineral transported to crushers and waste dumps is related to the number and capacity of the trucks in operation. Trucks with greater capacity tend to carry more volume per time interval. It is also important to highlight that  $x_{oc}$ ,  $x_{we}$  and  $x_p$  are calculated as a function of  $S$ , using the simulation model discussed in Section 3.

The constraints are expressed in equations (6) - (15).

$$g_1(S) = \sum_{\forall o \in \mathbb{C}} x_{oc} (q_{qc}^l - q_{qo}) \leq 0 \quad \forall q \in \mathbb{Q}; c \in \mathbb{C} \quad (6)$$

$$g_2(S) = \sum_{\forall o \in \mathbb{C}} x_{oc} (q_{qo} - q_{qc}^u) \leq 0 \quad \forall q \in \mathbb{Q}; c \in \mathbb{C} \quad (7)$$

$$g_3(S) = \sum_{\forall a \in \mathbb{A}} c_a^l y_{ap} - x_p \leq 0 \quad \forall p \in \mathbb{P} \quad (8)$$

$$g_4(S) = x_p - \sum_{\forall a \in \mathbb{A}} c_a^u y_{ap} \leq 0 \quad \forall p \in \mathbb{P} \quad (9)$$

$$g_5(S) = \frac{\sum_{\forall o \in \mathbb{O}} \sum_{\forall c \in \mathbb{C}} x_{oc}}{\sum_{\forall w \in \mathbb{W}} \sum_{\forall e \in \mathbb{E}} x_{we}} - row^u \leq 0 \quad (10)$$

$$g_6(S) = row^l - \frac{\sum_{\forall o \in \mathbb{O}} \sum_{\forall c \in \mathbb{C}} x_{oc}}{\sum_{\forall w \in \mathbb{W}} \sum_{\forall e \in \mathbb{E}} x_{we}} \leq 0 \quad (11)$$

$$h(S) = y_{ap} + y_{vp} - 2y_{va} = 0 \quad \forall a \in \mathbb{A}; v \in \mathbb{V}; p \in \mathbb{P} \quad (12)$$

$$|\mathbb{C}|, |\mathbb{E}|, |\mathbb{O}|, |\mathbb{W}|, |\mathbb{Q}|, |\mathbb{A}|, |\mathbb{V}| > 0 \quad (13)$$

$$c_a^l, c_a^u, q_{qc}^l, q_{qc}^u, u_v > 0 \quad \forall a \in \mathbb{A}; q \in \mathbb{Q}; c \in \mathbb{C}; v \in \mathbb{V} \quad (14)$$

$$c_a^u \geq c_a^l; q_{qc}^u \geq q_{qc}^l; row^u \geq row^l \quad \forall a \in \mathbb{A}; q \in \mathbb{Q}; c \in \mathbb{C} \quad (15)$$

These constraints define key aspects of the operating environment. These constraints represent the limits of chemical quality deviation defined by equations (6) - (7); production limits for each pit defined by equations (8) - (9); ore/waste ratio limits defined by equations (10) - (11); compatibility between shovel and truck defined by equation (12); and non-negativity defined by equations (13) - (14) and model validity constraints defined by equation (15).

Note that at each available pit there is at least one shovel. A shovel has a non-zero productivity only if a truck is available for it to load. Therefore, if during the simulation there are no trucks dispatched to the mining pit where a shovel is operating, its productivity will be zero, leading to the violation of constraint defined by equation (8). On the other hand, the maximum production limit of the shovel - constraint defined by equation (9) - cannot be exceeded, and there is therefore a limit to the number of trucks that can be loaded by a shovel. The compatibility between shovels and trucks - constraint defined by equation (12) - must be met to avoid dispatching a truck to a mining pit where there is no compatible shovel.

### 4.3 Literature Placement of the Proposed Mathematical Model

Several works dealing with the problem of operational planning in open-pit mines are present in the literature. To adequately contrast the mathematical model proposed in the present work with existing ones, we selected papers that have focus on the dynamic dispatch of trucks, which is the emphasis of this work. Table 1 presents a summary of the main characteristics of the models. Columns of the table refer to the reference number of the reviewed works, with this paper identified as [PM], in the last column.

Table 1 allows some interesting considerations to be made. First, notice that the vast majority of works consider ore and waste production in some way, either directly in their objective functions (60% of the works) or indirectly through production deviation (20% of the works). All works present some kind of constraint, although not all of them present their mathematical formulation. Finally, it is also interesting to observe that only references [14] and [9] address the problem using a multi-objective methodology: all others, despite considering more than one objective, use other techniques such as goal programming [15] or convex scalarization.

Table 1: Summary of reviewed works in terms of objective functions considered, types of constraints and other problem characteristics.

Criteria evaluated in comparison	Contemplated mathematical models									
Objective functions	[16]	[17]	[13]	[10]	[5]	[18]	[4]	[14]	[9]	[PM]
Max. quality of ore	✓	✓	✓		✓			✓		
Min. fleet size of trucks	✓			✓	✓			✓		
Max. production (ore and waste)		✓	✓	✓		✓	✓			✓
Min. trucks queue time		✓				✓			✓	
Min. cost of shovel allocation			✓							
Min. production deviations					✓			✓		
Min. average cost of operation of the truck fleet										✓
Min. idle shovels						✓				
Min. distance traveled by trucks									✓	
<b>Number of objectives</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>2</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>3</b>	<b>2</b>	<b>2</b>
Constraint functions	[16]	[17]	[13]	[10]	[5]	[18]	[4]	[14]	[9]	[PM]
Relationship waste vs. ore	✓		✓		✓	✓		✓	✓	✓
Ore quality limits	✓		✓		✓			✓	✓	✓
Limits of production of pits	✓	✓	✓		✓	✓		✓	✓	✓
Shovels productivity limits			✓	✓	✓	✓		✓	✓	✓
Allocation of shovels by pit			✓		✓			✓	✓	
Compatibility between equipment			✓		✓	✓	✓	✓	✓	✓
Size limit of the truck fleet						✓	✓			
Allocation of several shovels per pit									✓	✓
Operation with several crushers						✓			✓	✓
Number of truck trips			✓							
<b>Number of constraints</b>	<b>3</b>	<b>1</b>	<b>7</b>	<b>1</b>	<b>6</b>	<b>6</b>	<b>2</b>	<b>6</b>	<b>8</b>	<b>7</b>

## 5. PROPOSED ALGORITHMS

In this work we investigate two algorithms to solve the truck dispatch problem. Since candidate solutions are represented as matrix tuples (see Section 4.1) encapsulating truck availability information together with the dispatch sequences, considering also the compatibility between trucks and shovels, specific operators were developed to work with these variables. The proposed operators guarantee the feasibility of new candidate solutions generated, therefore removing the need of costly repair operators.

The first proposed method, called *Multi-Objective Genetic Algorithm - Tuple Representation 1* (MOGA-TR1), includes a constructive heuristic to generate the initial candidate solutions, and uses the usual GA operators, *crossover* and *mutation*, adapted to work with the solution representation described in Section 4.1. The second method, *Multi-Objective Genetic Algorithm - Tuple Representation 2* (MOGA-TR2), replaces the GA operators of MOGA-TR1 by an approach based on *Path-Relinking* [19], also adapted for the solution representation proposed in this work.

A third approach was also developed here, in the form of a *Greedy Heuristic* (GH) [3,4], to be used as a comparison baseline. This algorithm uses only the size of the queues at each shovel to dispatch trucks to those with the smallest queues. Information such as distances between places, equipment productivity, and the quality of the material are not used by this method, and the performance of this algorithm is intended to represent the bare minimum required for any method attempting to solve the truck dispatch problem in open-pit mines.

## 5.1 MOGA-TR1

The first specially tailored evolutionary algorithm developed in this work, called MOGA-TR1, combines the principles of the *Non-dominated Sorting Genetic Algorithm-II* (NSGA-II) [20] with a procedure to create the initial population. As mentioned earlier, the crossover and mutation operators are adapted to deal with the solution encoding described in Section 4.1. The pseudo-code of MOGA-TR1 is shown in Algorithm 1. The algorithm starts by building a set of candidate solutions, and then iterating over custom-tailored variation and selection operators until a user-defined stop criterion (in this work, a predefined number of candidate solutions visited) is reached. Selection of candidate solutions to compose the updated population is based on nondominated sorting, with crowding distance used as a secondary selection criterion.

The initial population ( $\mathcal{P}_a$ ) is initialized using a constructive heuristic (see Section 5.1.1). The outer loop iterates until a stop criterion is reached. The first inner loop (lines 6–13) is responsible for creating new solutions and including them into a population  $\mathcal{P}_b$ . Function *BinaryTournament*( $\mathcal{P}_a$ ) selects two candidate solutions from population  $\mathcal{P}_a$  according to the *Crowded-Comparison* operator [20], which are then subject to the crossover and mutation operators described in Section 5.1.2. The newly created candidate solutions  $S^{3'}$  and  $S^{4'}$  are evaluated and included into population  $\mathcal{P}_b$ . From line 14 onwards, the proposed algorithm uses the usual selection procedure of NSGA-II [20]: the current and new candidate solutions are combined into a new population  $\mathcal{U}$ , and an efficient non-dominated sorting procedure (line 15) [21] is applied to rank the candidate solutions in terms of dominance relations. Lines 16–26 reconstitute the population  $\mathcal{P}_a$  by sequentially adding solutions contained in successive fronts of  $\mathcal{F}$  until a front  $\mathcal{F}_f$  is reached that would make  $|\mathcal{P}_a| > pSize$ . The candidate solutions of this last front are then sorted according to their crowding distance [20], and added sequentially to  $\mathcal{P}_a$  until the predefined population size  $pSize$  is reached. Finally, Lines 27–29 ensure that only non-dominated solutions are returned at the end of the algorithm.

---

### Algorithm 1: MOGA-TR1

---

```

Input: Mine: mine characteristics; pSize: population size; m: number of dispatches; maxEvaluations: stop criterion; pc: crossover rate; pm: mutation rate; t: operation time
Output:  $\mathcal{P}_{nd}$ : non-dominated solutions set
1  $\mathcal{P}_a \leftarrow CreateInitialPopulation(Mine, pSize, m, t)$  // Sec. 5.1.1
2  $n_{eval} \leftarrow pSize$ 
3 while  $n_{eval} \leq maxEvaluations$  do
4    $\mathcal{P}_b \leftarrow \emptyset$ 
5    $c \leftarrow 0$ 
6   while  $c < (pSize/2)$  do
7      $[S^1, S^2] \leftarrow BinaryTournament(\mathcal{P}_a)$  // See [20]
8      $[S^3, S^4] \leftarrow Crossover(S^1, S^2, p_c)$  // Sec. 5.1.2
9      $[S^{3'}, S^{4'}] \leftarrow Mutation(S^3, S^4, p_m)$  // Sec. 5.1.2
10    Evaluate( $S^{3'}$ , t, Mine)
11    Evaluate( $S^{4'}$ , t, Mine)
12     $n_{eval} \leftarrow n_{eval} + 2$ 
13     $\mathcal{P}_b \leftarrow \mathcal{P}_b \cup S^{3'} \cup S^{4'}$ 
14   $\mathcal{U} \leftarrow \mathcal{P}_a \cup \mathcal{P}_b$ 
15   $\mathcal{F} \leftarrow NonDominatedSorting(\mathcal{U}, 2)$  // See [21]
16   $\mathcal{P}_a \leftarrow \emptyset$ 
17  for  $f \in \{1, \dots, NumberOfFronts(\mathcal{F})\}$  do
18    if  $NumberOfSolutions(\mathcal{F}_f) \leq (pSize - |\mathcal{P}_a|)$  then
19       $\mathcal{P}_a \leftarrow \mathcal{P}_a \cup \mathcal{F}_f$ 
20    else
21      break
22   $\mathcal{C} \leftarrow CrowdingDistanceSort(\mathcal{F}_f)$ 
23   $c \leftarrow 1$ 
24  while  $|\mathcal{P}_a| \leq pSize$  do
25     $\mathcal{P}_a \leftarrow \mathcal{P}_a \cup \mathcal{C}_c$ 
26     $c \leftarrow c + 1$ 
27   $\mathcal{F} \leftarrow NonDominatedSorting(\mathcal{P}_a, 2)$  // See [21]
28   $\mathcal{P}_{nd} \leftarrow \mathcal{F}_1$ 
29  return  $\mathcal{P}_{nd}$ 

```

---

### 5.1.1 Generating the Initial Population

The initial population is created based on the set of available trucks and shovels. Algorithm 2 illustrates the constructive heuristic, which ensures that constraint defined by equation (12) is satisfied. In this algorithm, the *NumberOfTrucks*(*Mine*) function returns the number of available trucks in the mine and stores this value in variable  $sT$ . Each iteration of the outer loop builds one candidate solution for the initial population. The second loop (over  $i$ ) visits one available truck at a time. At each iteration, *RandBinaryValue*() randomly sets the availability of the  $i$ -th truck, and in the innermost loop *BuildDispatch*(*Mine*) randomly creates the  $j$ -th tuple  $(\alpha, \beta)$ , which defines one dispatch of the truck. This function ensures that the tuples are operationally valid, i.e., that candidate solution  $S^k$  will satisfy constraint defined by equation (12). Thus, if  $\alpha$  is set as an ore pit, its



corresponding  $\beta$  must be a crusher; or if  $\alpha$  is a waste pit, then its  $\beta$  must be a waste dump. Finally,  $S^k$  is evaluated and added to  $\mathcal{P}$ .

---

**Algorithm 2: Generate Initial Population (CIP)**


---

**Input:** *Mine*: mine characteristics; *pSize*: population size; *m*: number of dispatches; *t*: operation time

**Output:**  $\mathcal{P}$ : set of candidate solutions

```

1  $\mathcal{P} \leftarrow \emptyset$ 
2  $sT \leftarrow \text{NumberOfTrucks}(\text{Mine})$ 
3 for  $k \in \{1, \dots, pSize\}$  do
4   for  $i \in \{1, \dots, sT\}$  do
5      $s_{i1}^k \leftarrow \text{RandBinaryValue}()$ 
6     for  $j \in \{2, \dots, (m+1)\}$  do
7        $s_{ij}^k \leftarrow \text{BuildDispatch}(\text{Mine})$ 
8    $\text{Evaluate}(S^k, t, \text{Mine})$ 
9    $\mathcal{P} \leftarrow \mathcal{P} \cup S^k$ 
10 return  $\mathcal{P}$ 
    
```

---

### 5.1.2 Crossover and Mutation

In this work we use a *Cut-Point Operator* (CPO) for crossover. This operator uses two cut points to perform recombination of two candidate solutions. The operator randomly defines two values,  $\kappa \in [1, |\mathbb{V}|]$  to cut vector  $\mathbf{v}^k$ , i.e., the first column of  $S^k$ ; and  $\gamma \in [1, m]$ , a column-wise cut point for matrix  $D^k$  (which represent the leftmost  $m$  columns of  $S^k$ ). This crossover procedure is illustrated in Figure 4.

$$\begin{array}{c}
 S^1 = \left[ \begin{array}{c|ccc|c}
 \mathbf{v} & d_1 & d_2 & d_3 & d_4 \\
 \hline
 1 & (1, 5) & (2, 5) & (1, 6) & (3, 8) \\
 0 & (3, 8) & (2, 5) & (1, 5) & (3, 7) \\
 1 & (2, 6) & (3, 7) & (4, 8) & (1, 6) \\
 1 & (2, 5) & (1, 6) & (3, 8) & (2, 6)
 \end{array} \right] \quad S^2 = \left[ \begin{array}{c|ccc|c}
 \mathbf{v} & d_1 & d_2 & d_3 & d_4 \\
 \hline
 1 & (3, 7) & (1, 6) & (1, 6) & (2, 6) \\
 1 & (1, 5) & (1, 5) & (2, 6) & (2, 6) \\
 0 & (1, 5) & (2, 5) & (3, 8) & (3, 7) \\
 1 & (2, 6) & (1, 6) & (4, 8) & (1, 5)
 \end{array} \right] \\
 \\
 S^3 = \left[ \begin{array}{c|ccc|c}
 \mathbf{v} & d_1 & d_2 & d_3 & d_4 \\
 \hline
 1 & (1, 5) & (1, 6) & (1, 6) & (2, 6) \\
 0 & (3, 8) & (1, 5) & (2, 6) & (2, 6) \\
 1 & (2, 6) & (2, 5) & (3, 8) & (3, 7) \\
 1 & (2, 5) & (1, 6) & (4, 8) & (1, 5)
 \end{array} \right] \quad S^4 = \left[ \begin{array}{c|ccc|c}
 \mathbf{v} & d_1 & d_2 & d_3 & d_4 \\
 \hline
 1 & (3, 7) & (2, 5) & (1, 6) & (3, 8) \\
 1 & (1, 5) & (2, 5) & (1, 5) & (3, 7) \\
 0 & (1, 5) & (3, 7) & (4, 8) & (1, 6) \\
 1 & (2, 6) & (1, 6) & (3, 8) & (2, 6)
 \end{array} \right]
 \end{array}$$

Figure 4: Example of crossover operator, for  $\kappa = 3, \gamma = 1$ . Candidate solutions  $S^3$  and  $S^4$  are the result of applying the operator to the pair  $(S^1, S^2)$ .

For the mutation procedure we employ a *Binary Mutation Operator* (BMO) [15], which can randomly change any element of  $S^k$  with a user-defined probability  $p_m$ . In the case of the truck availability vector this change represents a bit flip, while for the tuples this change represents a randomization of an  $(\alpha, \beta)$  pair.

## 5.2 MOGA-TR2

The second algorithm proposed here is similar to MOGA-TR1, but instead of the crossover and mutation operators it features a Path-Relinking (PR) method as its variation engine. The idea of employing PR to generate new candidate solutions have already been explored by Ribeiro and Vianna [22], which provided the original motivation for this version of the algorithm. However, in that work Path-Relinking was applied in solutions with a structure that is different from the presently proposed one, and in the context of a Simple Genetic Algorithm.

The pseudo-code of MOGA-TR2 is shown in Algorithm 3. The initial population ( $\mathcal{P}_a$ ) is created using the constructive heuristic (line 1) of Section 5.1.1. A population  $\mathcal{P}_\chi$  is composed of candidate solutions created using the proposed Path-Relinking operator (line 6), described in Section 5.2.1. Each newly created candidate solution ( $\mathcal{P}_\chi^i$ ) is evaluated and included in  $\mathcal{P}_b$  (lines 8 – 12), a procedure that is repeated until either  $\mathcal{P}_b$  is fully populated, or all solutions of the  $\mathcal{P}_\chi$  population are included in  $\mathcal{P}_b$ . After line 13 the remainder of the algorithm is exactly the same as lines 14 – 26 of Algorithm 1.

**Algorithm 3: MOGA-TR2**


---

**Input:** *Mine*: mine characteristics; *pSize*: population size; *m*: number of dispatches; *maxEvaluations*: stop criterion; *t*: operation time  
**Output:**  $\mathcal{P}_{nd}$ : non-dominated solutions set

```

1  $\mathcal{P}_a \leftarrow \text{CreateInitialPopulation}(\text{Mine}, \text{pSize}, m, t)$  // Sec. 5.1.1
2  $n_{eval} \leftarrow \text{pSize}$ 
3 while  $n_{eval} \leq \text{maxEvaluations}$  do
4    $\mathcal{P}_b \leftarrow \emptyset$ 
5   while  $\text{NumberOfSolutions}(\mathcal{P}_b) \leq \text{pSize}$  do
6      $\mathcal{P}_\chi \leftarrow \text{PathRelinking}(\mathcal{P}_a, m, \text{NumberOfTrucks}(\text{Mine}))$  // Sec. 5.2.1
7      $i \leftarrow 1$ 
8     while [ $\text{NumberOfSolutions}(\mathcal{P}_b) \leq \text{pSize}$  AND  $i \leq \text{NumberOfSolutions}(\mathcal{P}_\chi)$ ] do
9        $\text{Evaluate}(\mathcal{P}_\chi^i, t, \text{Mine})$ 
10       $n_{eval} \leftarrow n_{eval} + 1$ 
11       $\mathcal{P}_b \leftarrow \mathcal{P}_b \cup \mathcal{P}_\chi^i$ 
12       $i \leftarrow i + 1$ 
13 ..... lines 14–26 of Algorithm 1 .....
14  $\mathcal{F} \leftarrow \text{NonDominatedSorting}(\mathcal{P}_a, 2)$  // See [21]
15  $\mathcal{P}_{nd} \leftarrow \mathcal{F}_1$ 
16 return  $\mathcal{P}_{nd}$ 

```

---

**5.2.1 Path-Relinking Operator (PRO)**

In MOGA-TR2 the crossover and mutation operators are replaced by a procedure based on *Path-Relinking* (PR) [19, 23], which has previously been suggested as an approach to integrate intensification and diversification strategies in the context of *Tabu Search* [19]. This approach explores the neighborhood of a starting solution  $S^a$ . The goal is to build new solutions from  $S^a$  towards a *guide* solution ( $S^b$ ). The proposed procedure, called *Path-Relinking Operator*, is given in Algorithm 4.

**Algorithm 4: Path-Relinking Operator (PRO)**


---

**Input:**  $\mathcal{P}$ : solution set; *m*: number of dispatches; *sT*: number of trucks in the mine  
**Output:**  $\mathcal{P}'$ : solution set constructed by Path-Relinking Operator

```

1  $[S^a, S^b] \leftarrow \text{BinaryTournament}(\mathcal{P})$  // See [20]
2  $\mathcal{P}' \leftarrow \emptyset$ 
3  $k \leftarrow 1$ 
4  $S^k \leftarrow S^a$ 
5 for  $i \in \{2, \dots, sT\}$  do
6   if  $s_{i1}^b = 1$  then
7      $k \leftarrow k + 1$ 
8      $S^k \leftarrow S^{k-1}$ 
9      $s_{i1}^k \leftarrow s_{i1}^b$ 
10    for  $j \leftarrow 2$  to  $(m + 1)$  do
11       $s_{ij}^k \leftarrow s_{ij}^b$ 
12     $\mathcal{P}' \leftarrow \mathcal{P}' \cup S^k$ 
13 return  $\mathcal{P}'$ 

```

---

The *BinaryTournament*( $\cdot$ ) function selects the *starting* ( $S^a$ ) and *guide* ( $S^b$ ) solutions from population  $\mathcal{P}$ , and the outer loop iteratively creates new candidate solutions and adds them into solution set  $\mathcal{P}'$ . A new solution will be created only if  $s_{i1}^b$  indicates that the  $i^{th}$  truck is active, i.e., if  $s_{i1}^b = 1$ . In this case, a new candidate solution  $S^k$  is created from  $S^{k-1}$ , and the  $i^{th}$  line of the  $S^k$  solution is updated with the elements of the  $i^{th}$  line of  $S^b$ . The resulting solution  $S^k$  is added into the set  $\mathcal{P}'$ , which is returned at the end of the procedure.

**6. COMPUTATIONAL EXPERIMENTS****6.1 Test Instances**

In this work we employ four problem instances based on real data of an iron mining company located in the state of Minas Gerais, Brazil. These instances were previously used in the literature [5], and their definitions are available online [24]. The complexity of the problem is related not only to the mine parameters (number of trucks, pits etc.), but also to the number of cycles used for the mine simulation system. In this work we simulate the operation of each mine for one hour, allowing each truck to perform several cycles. Table 2 describes the main characteristics of the test instances: columns **Pits**, **Shovels**, **Trucks** and **Params** indicate the number of pits, shovels, trucks, and number of controlled chemical quality parameters, respectively, as well as the shovel productivities and truck capacities. Mines 1 and 2 differ only in the levels of quality of chemical elements. Truck costs were artificially generated as \$1 (for 56-tonne trucks) and \$3 (for 90-tonne trucks).

Table 2: Test instances used for the experimental validation.

Instance	Pits	Shovels (product., ton/h)	Trucks (capac., ton)	Params
Mines 1, 2	8	4 (900); 2 (1000); 2 (1100)	15 (56); 15 (90)	10
Mine 3	7	2 (500); 2 (400); 1 (600); 1 (800); 1 (900)	30 (56)	5
Mine 4	10	2 (400); 2 (500); 1 (600); 1 (800); 1 (900); 3 (1000); 3 (2600)	22 (56); 8 (90)	5

## 6.2 Optimization Results

The purpose of this section is to evaluate and compare the performance of the algorithms presented in section 5. The battery of tests was composed of 33 runs for each algorithm on each test instance. Experiments were run using an Intel Core i7-3632, 2.2GHz CPU with 8GB RAM machine. All the experiments considered the following (arbitrarily set) parameters: *population size* ( $pSize$ ): 200; *stopping criterion* ( $maxEvaluations$ ): 20,000 evaluations; *crossover rate* ( $p_c$ ): 0.9; *mutation rate* ( $p_m$ ): 0.4; *number of the dispatches for each truck* ( $m$ ): 20; *operating time of the mine* ( $t$ ): 1 hour. The parameters that define each instance are defined in the definition files available online [24].

The algorithms were compared using two quality indices:

- The difference in *Coverage of Two Sets* [15], i.e.,  $\mathcal{C}(A, B) - \mathcal{C}(B, A)$ . This indicator quantifies how much the first algorithm dominates the second. Since this is not a symmetric measure, we opted for using the difference as a measure of relative quality of algorithm A in relation to B.
- The normalized Hypervolume (HV) indicator [25], which is a unary indicator that is jointly determined by convergence and diversity.

Besides these two quantitative indicators, the performance of the algorithms was also investigated using the *Empirical Attainment Function* (EAF) [26], which provides a graphical (i.e., qualitative) representation of the distribution of the output for each algorithm on each instance.

For the quantitative indicators, the algorithms were compared using statistical tests based on 33 replicates<sup>2</sup> of each algorithm on each problem instance. The differences in the Coverage of Two Sets were tested using paired t-tests [27], with the replicate as the pairing variable. This pairing was induced in this particular case due to  $\mathcal{C}(A, B)$  and  $\mathcal{C}(B, A)$  being computed for the same final solution set at each replicate. The paired t-test was selected due to it requiring relatively few distributional assumptions (which were all observed for the data), as well as having better statistical power than its nonparametric counterparts. The comparisons regarding hypervolume were performed using the Welch t-test for two independent samples (in this case no pairing variable was influential enough to justify the loss of degrees-of-freedom resulting from pairing). This test is also subject to few distributional assumptions and provides adequate statistical power for the within-instances comparison. Due to the limit number of available instances, aggregate tests over all instances (e.g., block ANOVA or Friedman) would unfortunately have very low statistical power, and were therefore not performed.

Regarding the *Coverage of Two Sets*, three paired comparisons (MOGA-TR1 vs. GH, MOGA-TR2 vs. GH, and MOGA-TR1 vs. MOGA-TR2) were performed under each instance. To prevent the accumulation of the type-I errors of the statistical tests the Bonferroni correction was applied [27], to guarantee a joint confidence level of 95%. The results are summarized in Table 3, which reveals some interesting trends. First, in all four instances the mean paired differences in coverage between the MOGA versions and the Greedy Heuristic were in favor of the former, with strongly significant differences and estimated confidence intervals that suggest that in most runs the sets returned by the MOGA versions dominated a large fraction of those returned by the GH method. This was an expected result, given the very low sophistication of the Greedy Heuristic. Another point of note is the fact that MOGA-TR1 and MOGA-TR2 presented statistically significant differences in mean CS performance for all instances, with MOGA-TR1 outperforming MOGA-TR2 in mines 1–3, and MOGA-TR2 presenting a better performance in mine 4.

The second quality index employed was the normalised Hypervolume (HV) indicator [25]. To help calculating the hypervolume we generated a massive dataset of candidate solutions ( $10^6$  for each instance) and used (i) the extreme non-dominated points from these datasets to define the reference point for the calculation of the HV of each algorithm on each instance, and (ii) the resulting nondominated solutions from this brute force exploration as a baseline to normalise the results of each algorithm on each instance. The statistical comparison of the hypervolume values of the algorithms was performed using Welch's t test [27], which does not require equality of variances. Given the strongly significant superiority of both MOGA versions over the CH, we opted for dropping the latter and concentrate the analysis on the comparison of MOGA-TR1 and MOGA-TR2. The results of these comparisons are summarized in Table 4. While in all cases the differences observed were statistically significant, in terms of practical significance the effect sizes suggest a different interpretation: the observed mean differences in hypervolume performance are relatively small, suggesting that the results obtained by both MOGA versions were very similar – i.e., statistically significant but irrelevant for most practical applications.

<sup>2</sup>The number of replicates was arbitrarily chosen.

Table 3: Comparisons of Coverage of Two Sets. The reported p-values and confidence intervals are Bonferroni-corrected for a joint 95% confidence level within each problem instance.

Mine	Algorithms	p-value	95% Confidence Interval
1	MOGA-TR1 – GH	$9.1 \times 10^{-37}$	$0.81 \pm 0.03$
	MOGA-TR1 – MOGA-TR2	$3.2 \times 10^{-14}$	$0.43 \pm 0.08$
	MOGA-TR2 – GH	$9.4 \times 10^{-33}$	$0.84 \pm 0.04$
2	MOGA-TR1 – GH	$6.8 \times 10^{-32}$	$0.79 \pm 0.04$
	MOGA-TR1 – MOGA-TR2	$1.1 \times 10^{-12}$	$0.42 \pm 0.09$
	MOGA-TR2 – GH	$1.9 \times 10^{-30}$	$0.78 \pm 0.04$
3	MOGA-TR1 – GH	$5 \times 10^{-31}$	$0.84 \pm 0.04$
	MOGA-TR1 – MOGA-TR2	$1.7 \times 10^{-21}$	$0.62 \pm 0.07$
	MOGA-TR2 – GH	$8.7 \times 10^{-27}$	$0.57 \pm 0.04$
4	MOGA-TR1 – GH	$9.9 \times 10^{-25}$	$0.64 \pm 0.05$
	MOGA-TR1 – MOGA-TR2	$6.4 \times 10^{-12}$	$-0.40 \pm 0.09$
	MOGA-TR2 – GH	$4.9 \times 10^{-29}$	$0.71 \pm 0.04$

Table 4: Comparison of Hypervolume. Confidence intervals refer to the difference in means (MOGA-TR1 – MOGA-TR2).

Mine	p-value	95% Confidence Interval
1	$8.5 \times 10^{-7}$	$0.0130 \pm 0.005$
2	$4.8 \times 10^{-3}$	$0.0068 \pm 0.005$
3	$1.1 \times 10^{-4}$	$0.0058 \pm 0.003$
4	$5.0 \times 10^{-22}$	$-0.0350 \pm 0.004$

Finally, the algorithms were evaluated in terms of their *Empirical Attainment Function* (EAF) [26]. For two-objective optimization problems, the EAF graphical representation provides information about the distribution of the output of an algorithm that can be more intuitive than quality indicators such as the Hypervolume [28].

The panels in Figure 5 were constructed using this method, and show the location of the differences between the EAFs of the two algorithms. On the left, points denote positive differences between the EAF of MOGA-TR1 over the one of MOGA-TR2, and on the right the differences are in favor of MOGA-TR2 over MOGA-TR1. The magnitude of the differences is encoded in greyscale.

The results shown in Figure 5 is consistent with the other indicators used, suggesting a better performance of MOGA-TR1 for Mines 1–3, with MOGA-TR2 presenting superior results for Mine 4. This suggests the need for a larger set of instances, to more accurately characterize the relative performance of the proposed approaches.

## 7. CONCLUSIONS

This paper presents a mathematical model for the multi-objective truck dispatch problem in the context of open-pit mining operations. The proposed model considers two objectives, namely the minimization of the cost of a (heterogeneous) fleet of trucks for the operation, and the maximization of total production (ore and waste). To solve this problem, two evolutionary metaheuristics were proposed, the first using specially designed crossover and mutation operators, and a second one employing a Path-Relinking strategy as its variation engine.

The proposed algorithms were validated against a greedy heuristic using four test instances. The results suggest that the MOGA algorithms have outperformed the baseline performance set by the greedy heuristic, which was expected given the greater sophistication of the evolutionary approaches. The results also suggest an advantage of MOGA-TR1 over MOGA-TR2, although the differences cannot be fully determined from the limited test set available. To improve on the results presented in this work, validation against solutions currently implemented in actual mining operations should be performed, to compare the quality of algorithmically-generated solutions against those devised by human experts. This is a strong prerequisite for actual adoption of the proposed approaches, and is a central aspect of continuity efforts.

## References

- [1] E. Topal and S. Ramazan. “A new MIP model for mine equipment scheduling by minimizing maintenance cost”. *European Journal of Operational Research*, vol. 207, no. 2, pp. 1065–1071, 2010.
- [2] R. Mena, E. Zio, F. Kristjanpoller and A. Arata. “Availability-based simulation and optimization modeling framework for open-pit mine truck allocation under dynamic constraints”. *International Journal of Mining Science and Technology*, vol. 23, no. 1, pp. 113–119, 2013.
- [3] S. Alarie and M. Gamache. “Overview of Solution Strategies used in Truck Dispatching Systems for Open Pit Mines”. *International Journal of Surface Mining, Reclamation and Environment*, vol. 16, no. 1, pp. 59–76, 2002.

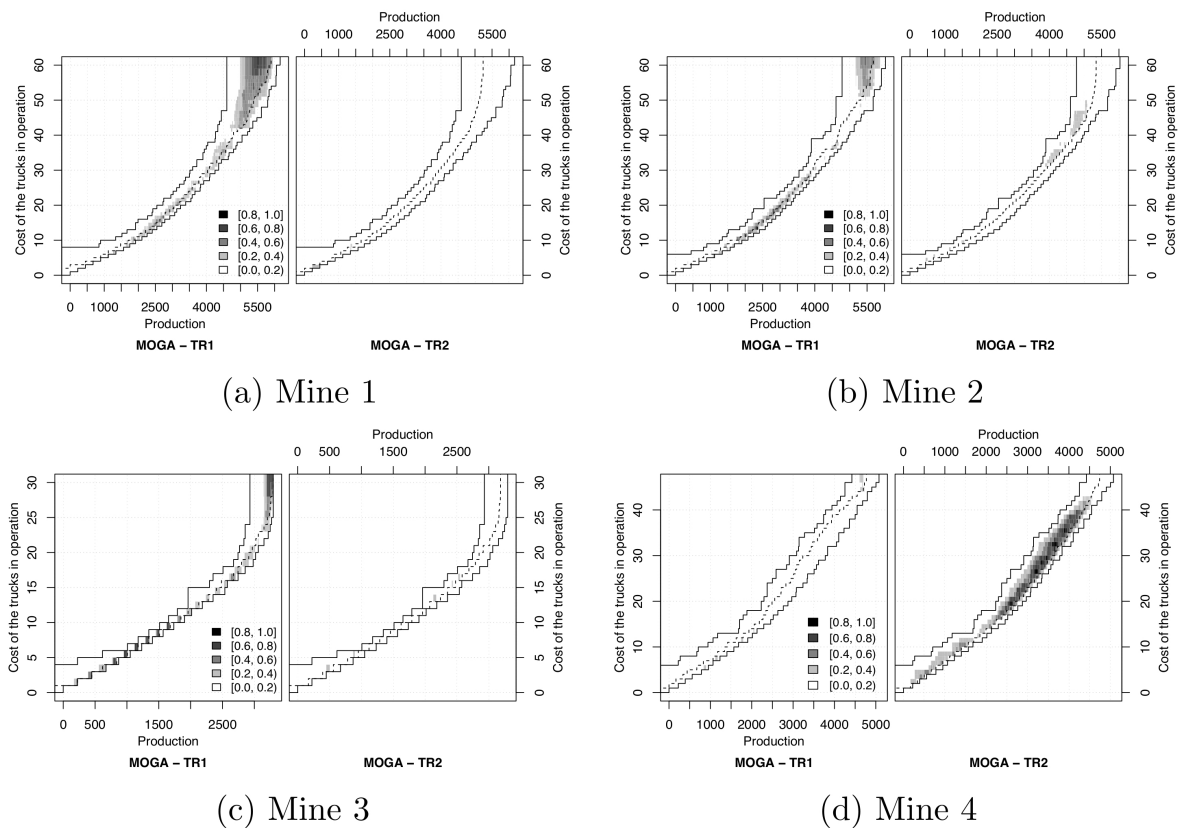


Figure 5: Differences between the EAFs for the four problem instances. The grey scale reflects the magnitude of the difference between the algorithms, with darker tones representing better performance of the algorithm referred at the bottom of each figure.

[4] G. Bastos, L. Souza, F. Ramos and C. Ribeiro. “A single-dependent agent approach for stochastic time-dependent truck dispatching in open-pit mining”. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 1057–1062, Oct 2011.

[5] M. Souza, I. Coelho, S. Ribas, H. Santos and L. Merschmann. “A hybrid heuristic algorithm for the open-pit-mining operational planning problem”. *European Journal of Operational Research*, vol. 207, no. 2, pp. 1041–1051, 2010.

[6] A. Newman, E. Rubio, R. Caro, A. Weintraub and K. Eurek. “A review of operations research in mine planning”. *Interfaces*, vol. 40, no. 3, pp. 222–245, 2010.

[7] D. Ahangaran, A. Yasrebi, A. Wetherelt and P. Foster. “Real-time dispatching modelling for trucks with different capacities in open pit mines”. *Archives of Mining Sciences*, vol. 57, no. 1, pp. 39–52, 2012.

[8] C. Ta, A. Ingolfsson and J. Doucette. “A linear model for surface mining haul truck allocation incorporating shovel idle probabilities”. *European Journal of Operational Research*, vol. 231, no. 3, pp. 770–778, 2013.

[9] J. B. Mendes, M. F. S. V. D’Angelo, N. A. Maia and R. R. Veloso. “A Hybrid Multiobjective Evolutionary Algorithm for Truck Dispatching in Open-Pit-Mining”. *IEEE Latin America Transactions*, vol. 14, no. 3, pp. 1329–1334, March 2016.

[10] S. Ercelebi and A. Bascetin. “Optimization of shovel-truck system for surface mining”. *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 109, no. 7, pp. 433–439, 2009.

[11] C.-T. Lin, M. Hsieh, W.-T. Hsiao, H.-T. Wu and T.-M. Cheng. “Optimizing the schedule of dispatching earthmoving trucks through genetic algorithms and simulation”. *Journal of Performance of Constructed Facilities*, vol. 26, no. 2, pp. 203–211, 2012.

[12] Y. Tan and S. Takakuwa. “A practical simulation approach for an effective truck dispatching system of open pit mines using VBA”. In *2016 Winter Simulation Conference (WSC)*, pp. 2394–2405, Dec 2016.

[13] M. Fioroni, L. Franzese, T. Bianchi, L. Ezawa, L. Pinto and G. Miranda. “Concurrent simulation and optimization models for mining planning”. In *Winter Simulation Conference*, pp. 759–767, 2008.

- [14] V. Coelho, M. Souza, I. Coelho, F. Guimarães, T. Lust and R. C. Cruz. “Multi-objective approaches for the open-pit mining operational planning problem”. *Electronic Notes in Discrete Mathematics*, vol. 39, no. 0, pp. 233–240, 2012.
- [15] C. Coello-Coello, G. Lamont and D. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problem*. Springer, second edition, 2007.
- [16] J. White and J. P. Olson. “Computer based dispatching in mines with concurrent operating objective”. *Mininig Engineering*, vol. 38, no. 11, pp. 1045–1053, 1986.
- [17] G. Alvarenga. “Despacho Ótimo de Caminhões Numa Mineração de Ferro Utilizando o Algoritmo Genético Com Processamento Paralelo”. Master’s thesis, Departamento de Engenharia Elétrica, Faculdade de Engenharia, Universidade Federal de Minas Gerais - UFMG, Belo Horizonte, Brazil, 1997.
- [18] R. Subtil, D. Silva and J. Alves. “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.
- [19] R. Martí, V. Campos, M. Resende and A. Duarte. “Multiobjective GRASP with path relinking”. *European Journal of Operational Research*, vol. 240, no. 1, pp. 54–71, 2015.
- [20] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–187, 2002.
- [21] R. F. Alexandre, C. H. N. R. Barbosa and J. A. Vasconcelos. “LONSA: A labeling-oriented non-dominated sorting algorithm for evolutionary many-objective optimization”. *Swarm and Evolutionary Computation*, 2017.
- [22] C. C. Ribeiro and D. S. Vianna. “A hybrid genetic algorithm for the phylogeny problem using path-relinking as a progressive crossover strategy”. *International Transactions in Operational Research*, vol. 16, no. 5, pp. 641–657, 2009.
- [23] R. Martí, M. Laguna and F. Glover. “Principles of scatter search”. *European Journal of Operational Research*, vol. 169, no. 2, pp. 359–372, 2006.
- [24] R. F. Alexandre, J. A. Vasconcelos and F. Campelo. “Additional Electronic Files”. <https://github.com/ORCSLab/MOPMOPP>, 2018.
- [25] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca and V. da Fonseca. “Performance Assessment of Multiobjective Optimizer: An Analysis and Review”. *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [26] C. Fonseca, V. Fonseca and L. Paquete. “Exploring the Performance of Stochastic Multiobjective Optimisers with the Second-Order Attainment Function”. In *Evolutionary Multi-Criterion Optimization*, edited by C. A. Coello Coello, A. Hernández Aguirre and E. Zitzler, volume 3410 of *Lecture Notes in Computer Science*, pp. 250–264. Springer Berlin Heidelberg, 2005.
- [27] D. Montgomery. *Design and Analysis of Experiments*. Wiley, 7th edition, 2008.
- [28] M. Ibáñez, L. Paquete and T. Stützle. “Exploratory Analysis of Stochastic Local Search Algorithms in Biobjective Optimization”. In *Experimental Methods for the Analysis of Optimization Algorithms*, edited by T. Bartz-Beielstein, M. Chiarandini, L. Paquete and M. Preuss, pp. 209–222. Springer Berlin Heidelberg, 2010.