Les Cahiers du GERAD ISSN: 0711–2440

Location of Water Depots in Open-Pit Mine Networks

J.-P. Riquelme-Rodríguez,
M. Gamache, A. Langevin

G–2014–12

March 2014

Location of Water Depots in Open-Pit Mine Networks

Juan-Pablo Riquelme-Rodríguez
GERAD, CIRRELT & Department of Mathematics and Industrial Engineering
Polytechnique Montréal
Montréal (Québec) Canada, H3C 3A7
juan-pablo.riquelme@polymtl.ca

Michel Gamache
GERAD & Department of Mathematics and Industrial Engineering
Polytechnique Montréal
Montréal (Québec) Canada, H3C 3A7
michel.gamache@polymtl.ca

André Langevin
CIRRELT & Department of Mathematics and Industrial Engineering
Polytechnique Montréal
Montréal (Québec) Canada, H3C 3A7
andre.langevin@polymtl.ca

March 2014

Les Cahiers du GERAD

G–2014–12

Abstract: The dust suppression of hauling roads in open-pit mines is done by periodically spraying water from a water truck. The objective of this article is to present a method for locating water depots in the road network so that penalty costs for the lack of humidity in roads and routing costs are minimized. Because the demands are located on the arcs of the network and the arcs require service more than once in a time horizon, this is a periodic capacitated arc routing problem. We compare three methods for finding the initial depot location using an L-A-R (location, allocation, and routing) approach combined with an adaptive large-neighborhood search to improve the solution. This method is the first for depot location in periodic arc routing problems.

Key Words: Location arc routing problem, Adaptive large-neighborhood search, Periodic capacitated arc routing problem.

Acknowledgments: This project was partially funded by the National Council of Science and Technology (CONACYT) of the Government of Mexico.

1 Introduction

Arc routing problems find the routes that satisfy a set of customers located on the arcs of a network while minimizing the associated routing cost. When the objective is to determine the best location of a set of depots so that routing costs are minimized, the problem becomes a location arc routing problem (LARP). The two tasks, locating the depots and finding routes to serve the customers, are performed simultaneously [6].

A classification of location routing problems can be found in [8]. Although many examples refer to location problems in the context of node routing problems, i.e., the demands are located on the nodes of the network, some applications can be found in the arc routing domain. A review of the methods used in LARPs can be found in [9].

The first LARP application was presented in 1989 by Levy and Bodin [6] and involved finding the best locations for postal carriers to park their vehicles in preparation for mail delivery. The authors introduced a location, allocation, and routing (L-A-R) approach in which the locations for the depots are selected, then the edges to be served are assigned to each depot, and finally routes are built to serve those edges. Ghiani and Laporte [3] approached the location rural postman problem (LRRP) by transforming it into a rural postman problem (RPP) where there are no bounds on the number of depots and using a branch and cut method to solve it. Ghiani and Laporte [4] reviewed the common applications of LARP, such as mail delivery, garbage collection, and street maintenance. The review covered common heuristics used to solve LARP problems including the abovementioned L-A-R and the A-R-L, in which customers are first assigned to a vehicle route, then the route is formed, and finally the depot locations are determined.

Other applications where location decisions are made in the arc routing domain include garbage collection using mobile depots [2]. Small-capacity trucks move along the streets, collecting garbage and delivering their contents to larger trucks used as temporary depots. The authors use a variable neighborhood descent to schedule meetings of the two types of vehicles so that the use of small trucks reduces the number of returns to the main depot. A similar application was presented in [1], where one type of vehicle is used to paint street lines while a second type is used to refill at specific points in the network.

We consider the problem of locating water depots to facilitate road watering in open-pit mines. This problem differs from the previously described LARP in that it is a capacitated problem (unlike the RPP applications), and the vehicles traversing the network have a limited capacity. It is also a periodic problem. The edges must be visited more than once in the time horizon; a solution that serves each edge only once is not useful for this problem. Given these characteristics, we treat the routing decisions as a periodic capacitated arc routing problem (PCARP).

The PCARP was introduced in [5] for a garbage collection problem in which the demands on the arcs varied from one period to the next, and a solution was needed for the whole time horizon instead of for individual periods. The PCARP has been applied to road watering in open-pit mines. Li et al. [7] presented two heuristic algorithms to minimize the routing costs of water trucks while mantaining the watering frequency on the edges. Riquelme-Rodríguez et al. [15] proposed a mathematical model to find an optimal route for one water truck, minimizing the routing costs and imposing penalty costs for having less humidity than the level required to ensure dust-particle retention. Riquelme-Rodríguez et al. [16] proposed an adaptive large-neighborhood search (ALNS) heuristic for the same problem for any number of water trucks. The only studies combining location and routing decisions for periodic applications are presented in [13] and [14]. Both refer to node routing problems. The contribution of this article is a mathematical model as well as a heuristic algorithm to deal with a location problem in the periodic arc routing domain.

The article is organized as follows. In Section 2, we present the definition of the problem of road watering in open-pit mines and the mathematical model. Our algorithm is presented in Section 3, and the results are discussed in Section 4. Finally, Section 5 presents concluding remarks.

2 Mathematical model

2.1 Problem definition

The most cost-effective method for suppressing dust in temporary roads is to spray water over them [12]. Because of evaporation and traffic volume, the humidity needs to be replenished periodically. A penalty cost is assigned for a level of humidity lower than that required to ensure dust-particle retention. Because the water trucks have a limited capacity, they reload at the depot before starting a new route. The objective is to find the location of water depots in the mine's road network that minimizes the penalty cost for lack of humidity and the routing cost.

This problem combines strategic and operational decisions. The placement of the depots is a long-term decision, so the performance of the vehicles is tested under different scenarios. A new scenario is created when the values of some of the parameters change. For example, at the beginning of the time horizon the roads may have different levels of humidity. Each initial humidity level represents a scenario.

2.2 Mathematical model

The model presented in this section is based on the model presented in [16] that aims to minimize operational costs such as the penalty and routing costs. We explore these costs under different scenarios to minimize the long-term costs such as vehicle and depot placement.

Consider a time horizon that corresponds to one working shift divided into T time periods. A time period is the amount of time it takes a water truck to cover a constant distance D at a constant speed. For example, a truck traveling at 20 km/h can cover a distance $D = 300\,\text{m}$ in approximately 1 minute (54 seconds). We assume that the service and deadheading speeds for the truck are the same. The truck may travel faster during deadheading, but there are several factors affecting this such as the presence of other trucks on the same road [7] and the road condition [18]. An analysis of different truck speeds for a model with one depot and one vehicle is presented in [15].

Consider a mixed network $G(N, E \cup A)$, where N is the set of nodes, and E is the set of edges that correspond to the roads of the mine network. A is a set of arcs that indicate the direction of traversal of each edge. $D \subseteq A$ is a set of arcs such that for each edge $[i, j] \in E$ there are two artificial arcs $(i, j), (j, i) \in D$. $B \subseteq A$ is a set of artificial loops located at each node $i \in N$ where it is possible to place a depot. A loop $(i, i) \in B$ is used to simulate the filling of a vehicle. Note that $B \cap D = \emptyset$.

Consider the following parameters of the model:

- C is the number of scenarios that simulate different conditions.
- K is the number of trucks of capacity Q^{max}.
- T is the number of time periods in the time horizon.
- A penalty cost P_{ij} is given to edge $[i, j] \in E$ if the humidity level is below the minimum level, $\overline{h_{ij}}$, required to ensure particle retention.
- hc_{ij} is the cost of watering arc $(i, j) \in D$ or filling at loop $(i, i) \in B$.
- tr_{ij} is the cost of traversing arc $(i, j) \in D$ without service or waiting at $(i, i) \in B$.
- cv is the cost of purchasing a vehicle and cd_b is the cost of establishing a depot at location $b \in B$.
- g_{ij} is the percentage of humidity loss due to the traffic volume in edge $[i, j] \in E$.
- e^{tc} is the evaporation factor for time period $t \in \{0, \dots, T\}$ and scenario $c \in \{1, \dots, C\}$.
- I_{ij}^{c} represents the initial humidity level of edge $[i, j] \in E$ in scenario c.
- H_{ij}^{max} is the maximum humidity level allowed for edge $[i, j]$.
- $MinDep$ and $MaxDep$ are, respectively, the minimum and maximum number of depots allowed in the network. In an open-pit mine, the number of potential location sites is limited by the fact that the mine's topology is in perpetual evolution.

- d_{ij} is the number of time periods required to travel along arc $(i,j) \in A$.
- $\Omega \subseteq A$ is the set of arcs that require more than one time period to be traversed, i.e., $d_{ij} > 1$.

Consider the following variables of the model:

- H_{ij}^{tc} is the humidity level of edge $[i,j] \in E$ at time t of scenario c.
- $Y_{ij}^{ktc} = 1$ if vehicle $k \in \{1,\ldots,K\}$ traverses arc $(i,j) \in D$ without service or waits at $(i,i) \in B$ at the beginning of time t of scenario c, and 0 otherwise.
- $X_{ij}^{ktc} = 1$ if vehicle k waters arc $(i,j) \in D$ or refills at $(i,i) \in B$ at the beginning of time t of scenario c, and 0 otherwise.
- $Z_b = 1$ if arc $b \in B$ is chosen as a location for a depot, and 0 otherwise.
- $R_k = 1$ if vehicle k is used, and 0 otherwise.
- Q^{ktc} is the quantity of water in vehicle k at the beginning of time t of scenario c.
- q_{ij}^{ktc} is the quantity of water delivered to edge $[i,j] \in E$ by truck k at the beginning of time t of scenario c.
- q_b^{ktc} is the quantity of water to be added to vehicle k at depot $b \in B$, at the beginning of time t of scenario c.
- $S_{uv}^{ktc} = 1$ if vehicle k is between nodes u and v of arc $(u,v) \in \Omega$ at the beginning of time t of scenario c, and 0 otherwise.
- $w_{ij}^{tc} = \max\{0, \overline{h_{ij}} - H_{ij}^{tc}\}$, $\forall [i,j] \in E$.

The complete model is as follows:

$$\min \frac{1}{C}\left[\sum_{c=1}^{C}\sum_{[i,j]\in E}\sum_{t=0}^{T} P_{ij} w_{ij}^{tc} + \sum_{c=1}^{C}\sum_{(i,j)\in A}\sum_{k=1}^{K}\sum_{t=0}^{T}(hc_{ij}X_{ij}^{ktc} + tc_{ij}Y_{ij}^{ktc})\right] + \sum_{k=1}^{K} cvR_k + \sum_{b\in B} cd_b Z_b \tag{1}$$

subject to:

$$w_{ij}^{tc} \geq \overline{h_{ij}} - H_{ij}^{tc} \qquad \forall [i,j] \in E,\ t \in \{0,\ldots,T\},\ c \in \{1,\ldots,C\} \tag{2}$$

$$H_{ij}^{(t+1)c} = (1 - (e^{tc} g_{ij}))H_{ij}^{tc} + \sum_{k=1}^{K} q_{ij}^{k(t+1)c} \qquad \forall [i,j] \in E,\ t \in \{0,\ldots,T-1\},\ c \in \{1,\ldots,C\} \tag{3}$$

$$H_{ij}^{0c} = I_{ij}^{c} \qquad \forall [i,j] \in E,\ c \in \{1,\ldots,C\} \tag{4}$$

$$q_{ij}^{kt} \leq H_{ij}^{max}(X_{uv}^{kt} + X_{vu}^{kt}) \qquad \forall \{[i,j] \in E | i = u, j = v; (u,v),(v,u) \in A\},\ t \in \{0,\ldots,T\},\ k \in \{1,\ldots,K\},\ c \in \{1,\ldots,C\} \tag{5}$$

$$H_{ij}^{tc} \leq H_{ij}^{max} \qquad \forall [i,j] \in E,\ t \in \{0,\ldots,T\},\ c \in \{1,\ldots,C\} \tag{6}$$

$$\sum_{[i,j]\in E} q_{ij}^{ktc} \leq Q^{ktc} \qquad \forall t \in \{0,\ldots,T\},\ k \in \{1,\ldots,K\},\ c \in \{1,\ldots,C\} \tag{7}$$

$$Q^{k0c} = Q^{max} \qquad \forall\, k \in \{1,\ldots,K\},\ c \in \{1,\ldots,C\} \tag{8}$$

$$Q^{ktc} \leq Q^{max} \qquad \forall t \in \{0,\ldots,T\},\ k \in \{1,\ldots,K\},\ c \in \{1,\ldots,C\} \tag{9}$$

$$Q^{k(t+1)c} = Q^{ktc} - \sum_{[i,j]\in E} q_{ij}^{ktc} - \sum_{b\in B} q_b^{ktc} \qquad \forall t \in \{0,\ldots,T-1\},\ k \in \{1,\ldots,K\},\ c \in \{1,\ldots,C\} \tag{10}$$

$$q_b^{ktc} = -Q^{max} X_b^{ktc} \qquad \forall t \in \{0,\ldots,T\},\ k \in \{1,\ldots,K\},\ b \in B,\ c \in \{1,\ldots,C\} \tag{11}$$

$$\sum_{c=1}^{C}\sum_{(i,j)\in A}\sum_{T_0}^{T} X_{ij}^{ktc} \leq R_k TC \qquad \forall k \in \{1,\ldots,K\} \tag{12}$$

$$\sum_{c=1}^{C}\sum_{t=0}^{T}\sum_{k=1}^{K}(X_b^{ktc}+Y_b^{ktc}) \leq Z_b KTC \qquad \forall b \in B \tag{13}$$

$$MinDep \leq \sum_{b\in B} Z_b \leq MaxDep \tag{14}$$

$$Z_{dep} = 1 \tag{15}$$

$$\sum_{c=1}^{C}\sum_{t=0}^{T}\sum_{k=1}^{K} X_b^{ktc} \leq Z_b \qquad \forall b \in B \tag{16}$$

$$X_{ij}^{ktc} + Y_{ij}^{ktc} \leq 1 \qquad \forall (i,j) \in A,\ t \in \{0,\dots,T\},\ k \in \{1,\dots,K\},\ c \in \{1,\dots,C\} \tag{17}$$

$$\sum_{k=1}^{K} X_{ij}^{ktc} \leq 1 \qquad \forall (i,j) \in A,\ t \in \{0,\dots,T\},\ c \in \{1,\dots,C\} \tag{18}$$

$$\sum_{(i,j)\in A|i=dep} X_{ij}^{k0c} + Y_{ijc}^{k0} = 1 \qquad \forall k \in \{1,\dots,K\},\ c \in \{1,\dots,C\} \tag{19}$$

$$\sum_{(i,j)\in A|j=dep} X_{ij}^{kT} + Y_{ij}^{kT} = 1 \qquad \forall k \in \{1,\dots,K\},\ c \in \{1,\dots,C\} \tag{20}$$

$$X_{ij}^{ktc} + Y_{ij}^{ktc} \leq \sum_{l|(j,l)\in A} X_{jl}^{k(t+d_{ij})c} + Y_{jl}^{k(t+d_{ij})c} \qquad \forall (i,j) \in A,\ t \in \{0,\dots,T\},\ k \in \{1,\dots,K\},\ c \in \{1,\dots,C\} \tag{21}$$

$$X_{ij}^{ktc} + Y_{ij}^{ktc} \leq S_{ij}^{k(t+m)c} \qquad \forall (i,j) \in \Omega,\ m \in \{0,\dots,d_{ij}-1\},\ t \in \{0,\dots,T-m\},\ k \in \{1,\dots,K\},\ c \in \{1,\dots,C\} \tag{22}$$

$$\sum_{(i,j)\in A\setminus\{(u,v)\}} (X_{ij}^{ktc} + Y_{ij}^{ktc}) \leq 1 - S_{uv}^{ktc} \qquad \forall t \in \{0,\dots,T\},\ (u,v) \in \Omega,\ k \in \{1,\dots,K\},\ c \in \{1,\dots,C\} \tag{23}$$

$$H_{ij}^{tc},\ w_{ij}^{tc} \geq 0 \qquad \forall [i,j] \in E, t \in \{0,\dots,T\},\ c \in \{1,\dots,C\} \tag{24}$$

$$q_{ij}^{ktc} \geq 0 \qquad \forall [i,j] \in E, t \in \{0,\dots,T\},\ k \in \{1,\dots,K\},\ c \in \{1,\dots,C\} \tag{25}$$

$$q_b^{ktc} \leq 0 \qquad \forall t \in \{0,\dots,T\},\ k \in \{1,\dots,K\},\ b \in B,\ c \in \{1,\dots,C\} \tag{26}$$

$$X_{ij}^{ktc},\ Y_{ij}^{ktc} \in \{0,1\} \qquad \forall (i,j) \in A, t \in \{0,\dots,T\},\ k \in \{1,\dots,K\},\ c \in \{1,\dots,C\} \tag{27}$$

$$Q^{tc} \geq 0 \qquad \forall t \in \{0,\dots,T\},\ c \in \{1,\dots,C\} \tag{28}$$

$$S_{uv}^{ktc} \in \{0,1\} \qquad \forall (u,v) \in \Omega, t \in \{0,\dots,T\},\ c \in \{1,\dots,C\} \tag{29}$$

$$Z_b \in \{0,1\} \qquad \forall b \in B \tag{30}$$

$$R_k \in \{0,1\} \qquad \forall k \in \{1,\dots,K\} \tag{31}$$

The objective function (1) minimizes the total cost. The penalty cost for a shortage of humidity and the routing cost of watering and deadheading are the result of the average cost of all scenarios. It is added to the vehicle cost and the depot-location cost.

Constraints (2) define variable w_{ij}^{tc} as the difference between the required and actual humidity levels at a time t for each scenario. If positive, this difference is penalized in the objective function. Constraints (3) establish the humidity level for the next time period, for each scenario. Constraints (4) establish the initial humidity level of each edge for each scenario. Constraints (5) and (6) limit, respectively, the amount of water delivered and the humidity level to the maximum humidity level. Constraints (7) are vehicle capacity constraints. Constraints (8) determine the initial quantity of water available in each vehicle. Constraints (9) limit the vehicle capacity at each time period to the maximum capacity. Constraints (10) determine the quantity of water in the vehicle at each time period. Constraints (11) allow the truck refilling at each depot.

Constraints (12) allow the use of trucks. Constraints (13) allow the use of a selected depot. Constraint (14) establishes the limits on the number of depots. Constraint (15) determines the main depot, $dep \in N$. This is the depot from which the vehicles depart at the beginning of the time horizon using constraints (19) and to which they return at the end of it according to constraints (20). Constraints (16) allow a vehicle to be refilled only if loop $b \in B$ has been selected as a depot. Constraints (17) limit the vehicle to either water or traverse an edge at a specific time period. Constraints (18) state that an edge can be serviced by only one vehicle in a given time period. Constraints (21) are flow conservation constraints for all the arcs in the network. Constraints (22) and (23) ensure that the vehicle stays on the same edge for the number of time periods it takes to service or traverse it. Finally, constraints (24) to (31) define the variables of the model.

We added the following constraints to break the symmetry in the solution tree:

$$Y_{00}^{ktc} = 1 \qquad \forall k \in \{1, \dots, K\},\ t \in \{0, \dots, k-1\},\ c \in \{1, \dots, C\} \tag{32}$$

$$\sum_{(0,j) \in A} X_{0j}^{ktc} + Y_{0j}^{ktc} = 1 \qquad \forall k \in \{1, \dots, K\},\ t = k,\ c \in \{1, \dots, C\} \tag{33}$$

Restrictions (32) ensure that the vehicles stay at the initial depot for an increasing number of periods. The first vehicle will stay at the depot until the first time period, the second vehicle will stay until the second period, and so on. Constraints (33) ensure that one vehicle leaves the depot at each period.

This model was coded in Cplex OPL and solved for a network of 8 nodes and 11 edges, $K = 5$ vehicles, and 10 scenarios. The scenarios included 10 different values for I_{ij}^{c}. The stopping criterion was 2 hours or a relative gap of 2%, and the program met this criterion after fewer than 18 time periods. For larger instances and a larger time horizon, a heuristic method is required.

3 Location and routing algorithms

We follow the L-A-R approach proposed by Levy and Bodin [6]. In the location phase, we make an initial placement of the depots. The allocation phase partitions the network so that a set of edges can be serviced by one truck. This is an adaptation from [6] since the edges are assigned to vehicles, not depots. The routing phase builds a route for each vehicle so that the penalty cost for having less humidity than required can be minimized. We add an improvement phase at the end of the last step by using an ALNS heuristic that changes the initial allocation and routing by means of a series of destroy-repair operators.

3.1 Location algorithms

We use three methods for the initial location of the depots.

1. Levy and Bodin location method

The first step in Levy and Bodin's L-A-R method is to select the initial location for a number of depots. We adapt this approach for our problem as follows:

- Determine the nodes that can be used as depots. For this problem we assume that every node is a potential location for a depot.
- Arrange the nodes in decreasing order of their *attractiveness measure* (AM). A node has a higher AM if it has more incident nodes and the priority of those nodes is higher:

  $$\text{AM} = A_1 * \text{Number of incident nodes} + A_2 * \text{Sum of the priorities of incident nodes}$$

  where A_1 and A_2 are the weights given to each contributing factor of the AM.

- Choose the nodes from the list using a separation criterion. The next node must have a minimum distance from the previous one; otherwise, it is discarded (removed from the AM list) and the next node is tested. The distance between the nodes is calculated using Dijkstra's algorithm. The selected node is removed from the AM list and added to the depot list. The minimum distance is a percentage (D_1) of the total distance the truck is able to water when at full capacity.

- The process is repeated until there are no nodes left in the AM list.

2. Random location

This method assigns P nodes randomly as depots, where P is the number of depots obtained by Levy and Bodin's method. Assuming that the more depots there are, the higher the cost, we can make a comparison only if all the methods have the same number of depots.

3. Clustering

This method distributes depots in the network by clustering the edges so as to minimize the distances to their assigned depot. Let $X_{ij} = 1$ if edge j is assigned to depot i, 0 otherwise; and $Y_i = 1$ if a depot is placed at node i, 0 otherwise. C_{ij} represents the shortest path from depot i to edge j, P is the number of depots obtained using Levy and Bodin's method, and E is the number of edges. The formulation is:

$$\min \sum_i \sum_j C_{ij} X_{ij} \tag{34}$$

subject to:

$$\sum_i Y_i = P \tag{35}$$

$$\sum_i X_{ij} = 1 \qquad \forall j \tag{36}$$

$$\sum_j X_{ij} \leq E Y_i \qquad \forall i \tag{37}$$

$$X_{ij} \in \{0,1\} \qquad \forall i,j \tag{38}$$

$$Y_i \in \{0,1\} \qquad \forall i \tag{39}$$

The objective function (34) minimizes the shortest distance between the selected depots and the edges of the network. Constraint (35) establishes the number of depots. Constraints (36) ensure that all the edges are assigned to a depot. Constraints (37) allow an edge to be assigned to node i if that node was selected as a depot. Constraints (38) and (39) define the variables.

The next two steps, allocation and routing, are based on the initial solution of the algorithm proposed by Riquelme-Rodríguez et al. [16] for the PCARP with inventory constraints.

3.2 Allocation

The network is partitioned into K sets of edges, where K is the number of vehicles. This procedure was described in [16] and is based on the *cluster-first-route-second* algorithm used in [11] for the PCARP with irregular services. It can be summarized as follows:

1. Select K edges called *seeds* that are far away from each other. The first seed is the edge at the greatest distance from the depot. From a set of selected seeds $\{s_1, \ldots, s_h\}$, where $h < k$, select seed s_{h+1} so that the product of the distances from s_{h+1} to the rest of the seeds is maximum.
2. Assign the remaining edges of the network by minimizing the sum of the lengths of the shortest paths from the edges to the seeds selected in step 1.

3.3 Routing

A route is calculated for each of the vehicles by means of a constructive algorithm, first proposed in [16]. Each vehicle can serve only the edges assigned to it in the allocation phase, but it can traverse any edge. A route with a starting and ending depot is called a *run*. The main depot represents the start of the first run and the end of the last one. Any vehicle can refill at any depot, including the main depot, before starting a new run. For each run, we calculate the total quantity of water used as well as the time needed to traverse the arcs on that run. Our routing procedure can be sumarized for each of the vehicles as follows:

1. Start with a list L_1 of available edges, i.e., edges that need service because their humidity level is below the required level. After an edge is serviced in the first run, it will not be available for service (removed from L_1) for subsequent runs until the inventory level is reduced enough so that it is available again.
2. From L_1, select the edge, $[i, j]$, with the highest priority and add it to a list of selected edges. Let L_2 be the list of edges from L_1 that are selected for servicing.
3. Assign a quantity of water to be delivered equal to $\alpha D d_{ij}$, where α is the water rate in number of liters per meter and D is the distance, in meters, that a truck travels in one time period at a constant speed.
4. Order the available neighbors, i.e., the edges adjacent to those in L_2, in increasing order of length of their shortest paths to the departure depot, and select the first edge to add to L_2. If no neighbor is available, select the closest available edge to the edges in L_2.
5. Repeat step 4 until the quantity of water in the truck is less than $\alpha D d_{ij}, \forall [i, j] \notin L_1$ or $L_1 = \emptyset$.
6. Order the edges in L_2 in increasing order of their shortest paths to the departure depot.
7. Connect the first edge to the departure depot, calculating the shortest path between them. Connect the rest of the edges in L_2 using the shortest paths between them. The direction in which the edges are traversed is important. An edge $[i, j]$ that is traversed either in the direction of the arc (i, j) or (j, i) is removed from L_2. The process is repeated until $L_2 = \emptyset$.
8. Find the closest depot to the last arc (i, j) in the sequence and find the shortest path to connect them. The set of arcs starting and ending in a depot is called a *run*.
9. Create as many runs as possible within the time horizon, repeating steps 1–8. If the time to complete a run ends after the end of the time horizon, reorganize the path to end the run at the main depot at the end of the time horizon.

3.4 Adaptive large-neighborhood search

We use an ALNS heuristic to improve the solution found by the L-A-R method. ALNS was introduced in [17]. It is an iterative process in which a set of destroy/repair operators, $\{O_1, \ldots, O_n\}$, modify the initial solution. The process is divided into segments; a segment consists of 250 iterations. At each iteration, an operator O_i is chosen using a *roulette-wheel* mechanism, i.e., it is chosen with probability $\rho_i / \sum_{i=1}^{n} \rho_i$, where ρ_i is the weight of O_i. It is defined as $\rho_i = C_i/U_i$ where C_i is the score of O_i, and U_i is the number of times it was used during one segment. When a segment starts, C_i is set to 0 for all operators, and the weights are updated so that the probability of selecting each operator changes according to the past performance. The operators with better performance have a higher probability of being selected. At the beginning of the procedure, the weights are the same, $\rho_i = 1/n$. After segment j, the weights for segment $j+1$ are calculated using $\rho_{i,j+1} = \rho_{ij}(1-r) + r\frac{C_i}{U_i}$, where $r \in \{0, 1\}$ is called the *reaction factor*. C_i is determined by three scores. The score is σ_1 if the use of O_i results in a better solution overall. The score is σ_2 if the use of O_i results in a solution better than the incumbent one but not better than the best solution, and it has not been explored before. Finally, the score is σ_3 if O_i results in a bad solution, accepted with probability $e^{-(f'-f)/\tau}$, where f is the current solution, f' is the new solution, and the temperature factor τ starts at τ_0 and decreases with each iteration via $\tau = \tau \times c$, where $0 < c < 1$. τ_0 is calculated from the initial solution so that a solution that is $\mu\%$ worse than the current solution is accepted with 50% probability [17].

We use seven destroy/repair operators from [16]: Operators O_1, O_2, and O_3 randomly exchange the edges that were previously assigned to be served by a specific vehicle in the allocation phase. O_1 performs a single edge exchange; O_2 exchanges two adjacent edges; and O_3 exchanges several edges. Operator O_4 randomly selects edges in the first run to change from *serviced* to *not serviced* or vice versa. Operators O_5 and O_6 randomly select a sequence of edges (serviced or not) from one run and exchange it for a sequence in the first run. Operator O_5 exchanges two runs performed by the same vehicle, while operator O_6 exchanges two runs performed by different vehicles. If the total water used with the addition of the inserted run in O_5 or O_6 exceeds the capacity of the vehicle, thus making the solution infeasible, edges are removed starting with the last one until the solution regains feasibility. Operator O_7 randomly changes the amount of water to be delivered.

All the operators destroy the existing solution and repair it in the same iteration. All the repaired solutions are feasible. If an operator results in an infeasible solution, it is discarded.

The inventory constraints (3) and (10) give the level of humidity and the vehicle capacity in period $t+1$ based on the levels in period t. Therefore, every time an operator changes the solution, the rest of the solution must be re-calculated for the rest of the time horizon. For example, if an edge previously not served is served because of operator O_4, it will not be available for the next run, so the list of edges that can be serviced in the second run is different.

4 Test results

4.1 Parameters

We performed the tests on the *mine* instances from [16]. The road networks are taken from 5 real mines, and the number of vehicles used is either 3 or 5.

The location parameters used to compute the attractiveness measure, A_1 and A_2, are set to 0.5 and 0.5. The distance percentage D_1 is set to 0.6. This is the highest percentage for which the smallest network *mine1* resulted in more than one depot.

The ALNS parameters are taken from [16] in order to compare the results. They were tuned by changing the value of one of them and choosing the one that delivered the best objective value. Keeping the first parameter fixed, we tested the second for several values. This process was repeated for all the parameters, resulting in $(r, \mu, \sigma_1, \sigma_2, \sigma_3) = (0.1, 0.8, 25, 10, 5)$.

The network parameters are taken from [16]. The time horizon is $T = 300$ time periods. The traversing cost is the same as the edge distance, i.e., $r_{ij} = d_{ij}$, while the watering cost is set to $c_{ij} = r_{ij} + 2$. To give priority to the penalty costs we set $P_{ij} = \max c_{ij} * priority_{ij}$, where the priority of edge $[i, j]$ is an integer from 1 to 10. The required level of humidity is set to $\overline{h_{ij}} = \alpha D d_{ij}$. I_{ij}^{tc} and H_{ij}^{max} are set to $0.1\overline{h_{ij}}$ and $1.1\overline{h_{ij}}$ respectively; b_{ij} is set to 3% for low-priority edges and 7% for high-priority edges. $e_t = e_{t-1} + 0.1$ if $\lceil 0.55T \rceil \leq t < \lceil 0.75T \rceil$; $e_t = e_{t-1} - 0.1$ if $\lceil 0.75T \rceil \leq t < \lceil 0.95T \rceil$; and $e_t = 1$ otherwise. These parameters are used to approximate the daytime evaporation rate, which is increased in the second half of the time horizon, as shown in the results for hourly evaporation given in [10].

Table 1 gives the characteristics of each network; $|N|$ represents the number of nodes and $|E|$ the number of edges. The last two columns show the number of depots obtained using Levy and Bodin's method when there is no restriction on the main depot (column 5), and when the main depot is located at node 0 (column 6).

All the algorithms were coded in Python and executed on a 1.8 GHz Intel Core i5-3337U notebook PC.

4.2 Alternative location of one depot

In the *mine* instances from [16], the depot is located by default at node 0. We tested placing the depot at the node with the highest degree of attractiveness according to Levy and Bodin's method, hereafter referred to as *node n*. Table 2 gives the results. The third column shows the total cost of the initial solution (no ALNS improvement). It is compared with the L-A-R method from Levy and Bodin (column 4), without any ALNS improvement; the better cost is highlighted. Columns 5 and 6 show the cost after ALNS improvement in both cases; the better result is highlighted. The stopping criterion is 25000 iterations or 7200 seconds.

In 8 out of 10 of the solutions with no ALNS improvement, the cost is lower when the depot is located at node n. In all the ALNS solutions, the cost is lower when the depot is located at node n.

Table 1: General information on the *mine* instances tested.

Network	Number of vehicles	$\|N\|$	$\|E\|$	Number of depots $dep = n$	Number of depots $dep = 0$
mine1 A	3	21	22	4	5
mine1 B	5	21	22	4	5
mine2 A	3	22	27	7	7
mine2 B	5	22	27	7	7
mine3 A	3	49	53	10	9
mine3 B	5	49	53	10	9
mine4 A	3	51	60	5	6
mine4 B	5	51	60	5	6
mine5 A	3	30	35	5	6
mine5 B	5	30	35	5	6

Table 2: Value of the objective function with one depot located at node 0 or at node n.

		Initial solution		ALNS	
Network	Number of vehicles	1 depot at node 0	1 depot at node n	1 depot at node 0	1 depot at node n
mine1 A	3	1308.55	**1257.80**	1236.86	**1159.24**
mine1 B	5	1248.35	**1159.82**	1172.15	**1007.01**
mine2 A	3	3325.69	**3317.48**	3153.40	**3107.59**
mine2 B	5	**3245.71**	3274.68	3022.37	**2965.73**
mine3 A	3	5812.03	**5795.60**	5699.44	**5680.91**
mine3 B	5	**5663.04**	5741.13	5661.11	**5604.57**
mine4 A	3	4184.20	**4126.75**	4087.70	**4055.74**
mine4 B	5	4056.81	**3933.53**	4017.61	**3933.53**
mine5 A	3	2487.81	**2448.89**	2391.64	**2184.95**
mine5 B	5	2428.56	**2266.13**	2310.03	**2030.72**

4.3 Comparison of one and several depots

The second test is a direct comparison of one and several depots. We used Levy and Bodin's method to find the depot locations. Table 3 gives the results for the initial solution (no ALNS improvement). Column 3 gives the cost when there is only one depot at node 0 ($dep = 0$). Column 4 gives the cost when there are N depots (N varies according to the network), and the main depot is located at node 0. Column 5 gives the cost when there are N depots, and the main depot is located at node n. Table 4 gives the same comparison of one and several depots after the ALNS improvement.

Table 3: Comparison of the objective function values for one and N depots.

Network	Number of vehicles	1 depot at node 0	N depots $dep = 0$ L-A-R	N depots $dep = n$ L-A-R
mine1 A	3	1308.55	1292.15	**1248.95**
mine1 B	5	1248.35	1176.83	**1170.25**
mine2 A	3	3325.69	3327.21	**3323.06**
mine2 B	5	3245.71	**3115.66**	3157.31
mine3 A	3	5812.03	5752.06	**5739.53**
mine3 B	5	5663.04	5639.34	**5624.56**
mine4 A	3	4184.20	4104.56	**4086.48**
mine4 B	5	4056.81	3972.98	**3918.58**
mine5 A	3	2487.81	2478.83	**2407.66**
mine5 B	5	2428.56	2309.24	**2251.46**

Table 4: Comparison of the objective function values for one and N depots after ALNS.

Network	Number of vehicles	1 depot at node 0	N depots $dep = 0$ L-A-R and ALNS	N depots $dep = n$ L-A-R and ALNS
mine1 A	3	1236.86	1082.80	**1082.29**
mine1 B	5	1172.15	858.28	**855.01**
mine2 A	3	3153.40	**2825.41**	2979.19
mine2 B	5	3022.37	**2539.98**	2781.32
mine3 A	3	5699.44	5624.09	**5553.20**
mine3 B	5	5661.11	**5515.76**	5530.12
mine4 A	3	4087.70	4044.91	**4009.93**
mine4 B	5	4017.61	3944.78	**3896.19**
mine5 A	3	2391.64	2285.00	**2217.75**
mine5 B	5	2310.03	2048.40	**1895.29**

In all the instances, the cost is lower when more depots are included in the network. We note that for one instance, *mine2B*, the solution is better when the main depot is forced to be at node 0.

Based on the ALNS results shown in Table 4, the best location for the main depot for network *mine2* is node 0. For the rest of the networks, except *mine3B*, the best location for the main depot is node n.

4.4 Comparison of the depot-location methods

We compared the three methods for establishing the initial location of the depots. We report the average cost obtained after testing 12 different scenarios. We tested 4 initial humidity levels, I^c_{ij}: 10%, 30%, 50%, and 70% of H^{max}_{ij}. These 4 scenarios were combined with 3 evaporation scenarios: $c = 1$ is an approximation of the evaporation behavior described in [10]; $c = 2$ is a scenario where there is no evaporation, and the loss of humidity is due to traffic volume only; and $c = 3$ is a situation where the humidity is briefly increased by weather events such as rain. Figure 1 shows the behavior of e^{tc} during the 300 time periods for the three scenarios.

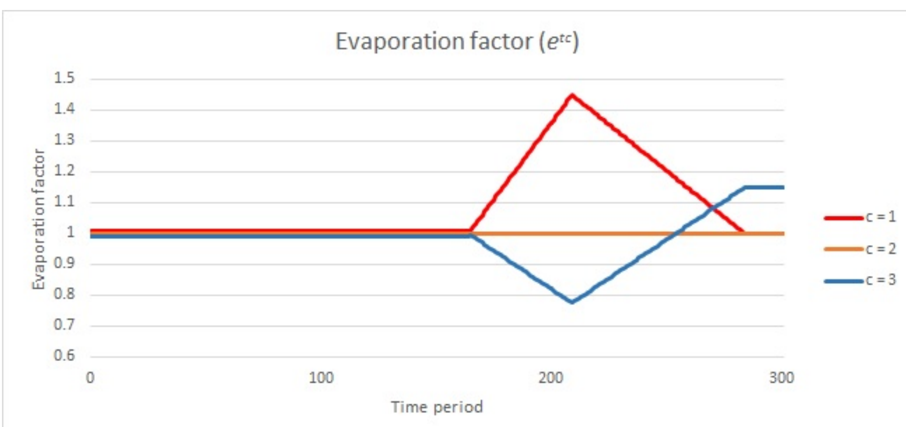

Figure 1: Evaporation factor e^{tc} for the three scenarios.

Table 5 gives the average total cost of the twelve scenarios obtained by each of the three location methods with the main depot at node 0 ($dep = 0$) in columns 3, 4, and 5 and at node n ($dep = n$) in columns 6, 7, and 8. The best solution is highlighted in each case.

The cluster method gives the best results in 8 of the 10 instances when the main depot is located at node 0 and in 5 of the 10 instances when it is located at node n. In 9 of the 10 instances, better results are obtained when the depot is located at node n.

Table 5: Comparison of the objective function values for the three depot-location methods without ALNS improvement.

		$dep = 0$			$dep = n$		
Network	Number of vehicles	LB	Random	Cluster	LB	Random	Cluster
mine1 A	3	1227.53	1229.53	**1200.64**	**1175.97**	1193.47	1213.10
mine1 B	5	1110.36	1112.09	**1098.80**	**1057.09**	1105.54	1076.18
mine2 A	3	3112.55	3144.92	**3078.70**	3138.54	3141.48	**3050.90**
mine2 B	5	2924.49	2967.20	**2909.43**	3018.42	3051.70	**2925.60**
mine3 A	3	5468.26	5465.03	**5456.66**	5440.88	5596.35	**5435.07**
mine3 B	5	**5360.87**	5399.77	5386.89	**5327.36**	5544.08	5344.57
mine4 A	3	3904.26	3912.51	**3884.24**	3877.77	3862.81	**3855.25**
mine4 B	5	3748.94	3768.01	**3742.52**	3704.52	3852.72	**3701.72**
mine5 A	3	2330.84	2310.05	**2302.90**	**2291.42**	2332.02	2326.47
mine5 B	5	2189.67	**2188.90**	2194.57	**2154.91**	2203.68	2197.63

Table 6 gives the average total cost after the ALNS improvement for the twelve scenarios. Columns 3, 4, and 5 compare the three location methods when $dep = 0$, while columns 6, 7, and 8 compare the three methods when $dep = n$. The ALNS performed as many iterations as possible in a time limit of 30 minutes for each scenario. The best solution is highlighted in each case.

Table 6: Comparison of the three depot-location methods after the ALNS improvement.

		$dep = 0$			$dep = n$		
Network	Number of vehicles	LB	Random	Cluster	LB	Random	Cluster
mine1 A	3	**1024.14**	1060.90	1055.96	**1012.73**	1029.74	1055.23
mine1 B	5	**861.73**	890.61	881.58	902.78	**863.09**	909.46
mine2 A	3	**2721.19**	2927.79	2800.79	2815.71	2856.77	**2707.83**
mine2 B	5	**2428.84**	2781.08	2542.39	2582.77	2654.88	**2437.61**
mine3 A	3	5426.61	5421.49	**5415.62**	5363.95	5488.90	**5402.80**
mine3 B	5	**5293.20**	5321.55	5321.39	**5227.82**	5397.96	5287.01
mine4 A	3	3855.63	3888.48	**3854.57**	3833.20	3844.71	**3821.77**
mine4 B	5	**3708.16**	3749.85	3714.81	3695.59	3836.29	**3674.54**
mine5 A	3	2268.94	2287.28	**2239.83**	**2220.35**	2292.81	2265.29
mine5 B	5	2152.69	2177.87	**2141.99**	**2126.60**	2178.76	2163.66

After the ALNS improvement, the results are similar for Levy and Bodin's method and the cluster method. This is likely a result of the randomness of the ALNS operators. Overall, 6 of the 10 instances have a better solution when the depot is located at node n.

4.5 Changing the initial depot location

The natural next step would be to modify the initial placement of the depots. However, because of the inventory constraints, any change in the location of the depots requires a recalculation of the initial solution and the subsequent improvement. The development of an algorithm to change the depot location is beyond the scope of this work.

5 Conclusion

In the instances tested, the depot location is an important factor in improving dust suppression. When there is a single depot, improving its location significantly reduces the penalty and routing costs. Including more than one vehicle significantly reduces the operational costs.

Our results show the importance of the location of the main depot. In the majority of the instances tested with one or several depots, the total cost was reduced when the main depot, where the vehicles start at the beginning of the time horizon and return at the end, was located at a node different from 0.

Future work will include developing an algorithm to change the initial location of the depots or to adjust the number of depots.

References

[1] A. Amaya, A. Langevin, and M. Trépanier. The capacitated arc routing problem with refill points. *Operations Research Letters*, 35(1):45–53, 2007.

[2] A. Del Pia and C. Filippi. A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, 13(2):125–141, 2006.

[3] G. Ghiani and G. Laporte. Eulerian location problems. *Networks*, 34(4):291–302, 1999.

[4] G. Ghiani and G. Laporte. Location-arc routing problems. *OPSEARCH*, 38(2):151–159, 2001.

[5] P. Lacomme, C. Prins, and Ramdane-Chérif W. Evolutionary algorithms for multiperiod arc routing problems, 2002.

[6] L. Levy and L. Bodin. The arc oriented location routing problem. *INFOR*, 27(1):74–94, 1989.

[7] J-Q. Li, P. Mirchandani, and P. Knights. Water truck routing and location of refilling stations in open pit mines. In *Proceedings of 2008 Australian Mining Technology Conference*, pages 141–156, Sunshine Coast, Australia, 2008. The Australian Institute of Mining and Metallurgy.

[8] R-B. Lopes, C. Ferreira, B-S. Santos, and S. Barreto. A taxonomical analysis, current methods and objectives on location-routing problems. *International Transactions in Operational Research*, 20(6):795–822, 2013.

[9] R-B. Lopes, F. Plastria, C. Ferreira, and B-S. Santos. Location-arc routing problem: Heuristic approaches and test instances. *Computers and Operations Research*, 43(0):309–317, 2014.

[10] J.M. Molina-Martínez, V. Martínez-Alvarez, M.M. González-Real, and A. Baille. A simulation model for predicting hourly pan evaporation from meteorological data. *Journal of Hydrology*, 318(14):250–261, 2006.

[11] I-M. Monroy, C-A. Amaya, and A. Langevin. The periodic capacitated arc routing problem with irregular services. *Discrete Applied Mathematics*, 161(45):691–701, 2013.

[12] R. Neulicht and J. Shular. Emission factor documentation for ap-42. section 13.2.2. unpaved roads. Technical report, U.S. Environmental Protection Agency, 1998.

[13] C. Prodhon. A hybrid evolutionary algorithm for the periodic location-routing problem. *European Journal of Operational Research*, 210(2):204–212, 2011.

[14] C. Prodhon and C. Prins. *A Memetic Algorithm with Population Management (MA—PM) for the Periodic Location-Routing Problem*, volume 5296 of *Lecture Notes in Computer Science*, book section 4, pages 43–57. Springer Berlin Heidelberg, 2008.

[15] J-P. Riquelme-Rodríguez, M. Gamache, and A. Langevin. Periodic capacitated arc-routing problem with inventory constraints. *Journal of the Operational Research Society*, 2013.

[16] J-P. Riquelme-Rodríguez, M. Gamache, and A. Langevin. Adaptive large neighborhood search for the periodic capacitated arc routing problem with inventory constraints. Technical report, *Les Cahiers du GERAD* G–2014–01, HEC Montréal, 2014.

[17] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.

[18] R-J. Thompson and A-T. Visser. Mining roads. mine haul road design, construction and maintenance management. *The Journal of the South African Institute of Mining and Metalurgy*, 100(3):169–180, 2000.