Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

DEVS MODELING AND SIMULATION OF THE LOADING AND HAULING PROCESS IN OPEN PIT MINES

Joel Santana
Alonso Inostrosa-Psijas

Escuela de Ingeniería Informática
Universidad de Valparaíso
General Cruz 222
Valparaíso, CHILE

Francisco Moreno

Dept. de Matemática y Ciencia de la Computación
Universidad de Santiago
Av. Libertador Bernardo O'Higgins 3363
Santiago, CHILE

Mauricio Oyarzún

Facultad de Ingeniería y Arquitectura
Universidad Arturo Prat
Arturo Prat 2120
Iquique, CHILE

Gabriel Wainer

Dept. of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive
Ottawa ON K1S 5B6, CANADA

ABSTRACT

Chile is the world's leading copper producer, with more than 5.6 million tons produced in 2020. Most of the produced ore comes from open pit mines, whose extraction process consists of different subprocesses, with ore hauling incurring the highest operational cost. Tools to improve this subprocess are of paramount importance. Most tools use approaches that rely on optimization based on analytical methods. However, these fail to capture human behavior or to consider fine-grained details. To this end, we present a DEVS (Discrete-Event System Specification) simulation model. The formal definition of DEVS helps with the design and experimentation. DEVS modular interfaces allow users to extend the model easily to consider more entities, mine layouts, and dispatching policies. Simulations of the model delivered precise results compared to the literature, providing a valuable tool for decision-making in the mining industry.

1 INTRODUCTION

The mining industry is present in 13 out of the 15 regions of Chile, and it is the country's primary revenue source. In addition, Chile is the top exporter of copper and is among the primary producers of iodine and lithium (USGS 2021), among other 25 minerals and their derivatives (SONAMI 2021). Their most common method of operation is open-pit mining (Cleveland and Morris 2015).

Open-pit mining comprises several phases for delivering minerals, the productive phase being the most critical. This operation comprises four primary processes: i) Ore extraction, ii) Ore processing, iii) Melting, and iv) Refining. Moreover, each of these is composed of different subprocesses. For example, ore extraction comprises perforation, perforation triggering, blasting, and mineral handling. Of these, mineral loading and hauling is the most expensive subprocess, with costs ranging from 50% to 60% of the overall operational costs in large open pit mines (Curry et al. 2014; Alarie and Gamache 2010; Bozorgebrahimi et al. 2003; da Cunha Rodovalho et al. 2016).

Several kinds of machinery are involved in the truck loading and hauling subprocess, among which the most important are trucks (used for mineral transportation) and shovels (that load the extracted material into the trucks). Other important entities involved in this subprocess are the stockpiles for temporal mineral storage, and waste dumps where the waste or inert mineral is dumped. Both are the destinations of the material transported by trucks (Burt and Caccetta 2014). The dispatch system orchestrates this subprocess (Soumis et al. 1989; Alarie and Gamache 2010), assigning trucks to shovels and stockpiles - in the most efficient manner - in each cycle. Thus, this subprocess's main objective is to transport the required amounts of material (from the shovels to stockpiles or waste dumps) to achieve the production goals of management officers using as few resources as possible (that is, trucks and shovels).

Since loading and hauling are the most expensive subprocesses, the main research approaches are proposed as optimization problems, focusing on reaching the maximum of ore processing by using the minimum resources through integer programming, discrete-event simulation (DES), or a combination of both (Upadhyay and Askari-Nasab 2018). In addition, some of them consider stochastic modeling for some variables, such as failure and repair rates (Mena et al. 2013). Nevertheless, with integer programming approaches, each factor of the model maps to a variable and, therefore, to a new constraint. Thus, detailed models are more complex to solve. On the other hand, many DES approaches are informal or based on programming languages, therefore, guaranteeing the correctness of the simulation model is complex and expensive, requiring an extensive testing process. Nevertheless, a formal specification approach can help improve simulation's security and development costs (Zeigler et al. 2000; Wainer 2017).

In this paper, we show that the DEVS (Discrete-Event System Specification) formalism (Zeigler et al. 2000) can also be a valuable tool for performing discrete-event simulation of the material handling subprocess in open-pit mines. Here we have developed the first model to study this kind of system using DEVS. The model is fully parametrizable and is flexible enough to make it easy to add new types of machinery, mine layouts, and truck-shovel assignment strategies. Thus, our contribution is two-fold: it is a powerful simulation model that serves as a potent testing tool for dispatching personnel to assess the impact of new changes on the mining facility, and - even when it is not our primary objective - it can be added to (and combined with) the stack of tools used by researchers for optimization studies.

The remainder of this article is as follows: Section 2 reviews related works regarding optimizing the material handling subprocess in open pit mines and the DEVS formalism. Section 3 describes the proposed DEVS model. In section 4, we validate our model and define a simple brute-force method for estimating the amount of needed equipment given a hauling goal. Finally, in section 5, we present the conclusions of our work and discuss the possibilities of future work.

2 RELATED WORK

2.1 Open-Pit Mines

In an open pit mine facility, the extraction process is carried out through four fundamental stages: drilling, blasting, loading, and transportation (also known as material handling). These last two operations are a critical logistics process, as they account for 50% to 60% of the total operating cost (Upadhyay and Askari-Nasab 2018; Alarie and Gamache 2010)**.**

Each cycle of the material handling subprocess (see Figure 1) consists of a truck assigned to a shovel. Upon arrival, the shovel loads it with ore or disposal soil. Then, the truck hauls the material to a stockpile (for ore storage) or a waste dump (for disposal). Queues of trucks might form before shovels and stockpiles or waste dumps. If no trucks are available, shovels become idle and wait. The dispatcher is responsible for assigning truck-to-shovel, truck-to-stockpile, or truck-to-waste dumps. Therefore, an efficient dispatcher policy is paramount for achieving production goals and reducing operational costs. Besides, pieces of equipment involved in this process are the most expensive and - in open pit mines – can be up to 70 trucks and 20 shovels (Chaowasakoo et al. 2017). The previous means that loading and transportation are the operations that influence costs the most (mostly due to its operational costs).

Furthermore, a truck can transport approximately 300 tons of material with copper ore - roughly valued at 30,000 US dollars - (National Productivity Commission 2017). Considering that a truck's hauling distance is about 10-25 minutes with shifts of 12 hours (Alarie and Gamache 2010), optimization to the process (no matter how small) represents significant savings. The main objective is to accomplish material handling goals while keeping shovel and truck utilization close to 100% without overestimating the needed machinery. If the truck fleet is too large compared to the number of shovels, queues will appear, reducing their utilization levels and increasing operational costs.

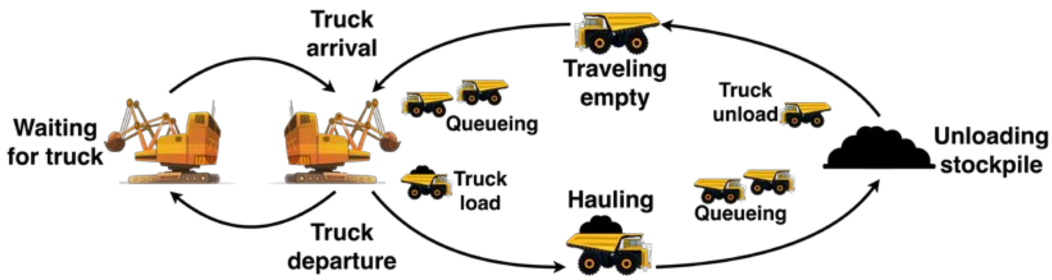

Figure 1: Loading and hauling cycle.

2.2 Open-Pit Mine Optimization, Modeling and Simulation

Most modeling efforts to solve this problem use optimization techniques for the open-pit material handling subprocess, usually performed through analytical methods (operations research) or discrete-event simulation (DES).

Some operations research studies focus on truck assignment and allocation, such as (Ta et al. 2013), whose authors present a model that minimizes the number of trucks for a given number of shovels. The model contemplates throughput and ore concentration constraints and also considers the probabilities of idle shovels for a more accurate truck allocation. The work of (Zhang and Xia 2015) addresses the dispatching problem by formulating an integer programming model to optimizing load-unload truck cycles while meeting production goals. More recent studies present mixed-integer non-linear programming for optimal fleet size subject to fleet efficiency (Mohtasham, et al. 2021). Whereas other efforts focus on reducing costs, such as the work of (Bajany et al. 2017), which presents a truck and shovel fuel consumption optimization method while meeting material handling demands, claiming to have a noticeable costs reduction. The research presented in (Upadhyay and Askari-Nasab 2016) proposed a mixed-integer linear goal programming model for achieving overall trucks and shovels utilization above 90%.

On the other hand, when compared to linear programming DES is a relatively new approach to material handling optimization. In (Fioroni, et al. 2008) the authors used a combination of simulation with a mixed integer linear programming-based model by planning optimal production to reduce costs. The authors of (Meng, et al. 2013) modeled the problem using Petri Nets. Another method was proposed by (Tan and Takakuwa 2016), they used an Arena simulation model combined with a simple algorithm (implemented in Visual Basic) to optimize the dispatching system to achieve stable production of a specific grade. Other simulation approaches are based in multiagent simulation, particularly focusing on an optimal assignment of trucks to shovels at each hauling cycle (Icarte et al. 2020; Icarte et al. 2021).

Finally, mixed approaches combine optimization methods with DES. Authors in (Mena et al. 2013) implemented a simulator that allocates trucks in routes according to their operating performance. In (Upadhyay and Askari-Nasab 2018) authors developed a DES framework that interacts with an operational optimization tool to develop an uncertainty-based short-term schedule for achieving long-term operational objectives.

DES advantages can be summarized as i) To allow decision-making personnel to test new assignment strategies without the risk of interfering with the actual process, ii) To test different layouts that could be too expensive or even impossible to test in real-life settings, iii) Simulate highly detailed and fine-grain models without incurring higher computational costs, in contrast to analytical models, where entities are homogenous, and each new detail becomes a new model constraint (Marin, et al. 2019; Gil-Costa et al.

2020), and iv) Optimizing through simulation allows to consider more realistic details, such as heterogeneous fleet machinery and to consider the behavior of the operators.

A formal discrete-event simulation model (being implemented as DES) delivers all the DES advantages mentioned above, plus it improves model verification. Also, formal specification mechanisms can help improve simulation's security and development costs (Zeigler et al. 2000; Wainer 2017). We chose the DEVS formalism because it provides several advantages for the modeling and simulation of similar systems: it is a hierarchical and modular technique that allows models to be easily extended through the description of models in multiple levels, provides the means for translating formal specifications into executable models (it is not dependent on the programming language), its logical and timing correctness relies on sound mathematical theory, DEVS can represent every system, and it is widely known in the simulation community (Zeigler et al. 2000; Fonseca 2009; Wainer 2017).

2.3 DEVS Formalism

The Discrete Event System Specification (DEVS) is a formalism for modeling and analyzing discrete event systems (Zeigler et al. 2000). DEVS provides the means for describing discrete-event systems by using two different kinds of elements to model a real system: atomic and coupled models. Atomic models are the most elemental and basic entities to represent systems and define the behavior of the system elements, while coupled models define the systems' structure. Furthermore, atomic models can react to internal and external events, which allows defining a way of specifying systems whose states change upon the reception of an input event or the expiration of a time delay. Coupled models are composed of two or more atomic or coupled models, and - thanks to the closure under coupling property - can be regarded as another DEVS model. Coupled models can be integrated to form a model hierarchy, allowing model reuse.

Coupled models may have their own input and output events. Upon the arrival of an external event, a coupled model has to redirect the input to one or more of its components. In addition, when a component produces an output, it must be mapped as another component's input or as an output of the coupled model itself. Thus, coupled models represent the structure of a system, whereas atomic models represent its behavior. Since DEVS is widely known in the simulation community, and due to lack of space, a formal definition of DEVS atomic and couple models is not supplied here, however, it can be found in (Chow and Zeigler 1994; Zeigler et al.2000; Wainer 2017).

3 DEVS MATERIAL HANDLING SUBPROCESS MODEL

3.1 DEVS Model Assumptions and Logic

This section presents the material handling subprocess (described in Figure 1) modelled using the DEVS formalism. A graphical representation of the DEVS model definition corresponding to the material handling subprocess is depicted in Figure 2.

It is worth mentioning that the proposed model is not spatially explicit, so aspects related to locations and distances are represented differently. For example, the position of each shovel and stockpile in the mine layout and the routes among them are represented as the time to travel between them (instead of location and distances). It implies that the way trucks travel is only represented as a delay in the time it takes them to travel from one point to another; that is, it is not possible to (at least directly) represent the location of a truck in a given route other than when it begins or finish its travel. As such, each factor of the route that affects the displacement of trucks is represented as shorter/longer delays. Thus, route congestion cannot be represented directly but as a longer travel time. Another aspect that is represented differently is when queues form at a given shovel or stockpile. Such queues cannot be represented in the surrounding space; instead, queues are defined directly on the shovels or stockpiles. Despite not being a spatially explicit model, it allows fast simulations while not losing precision, as shown in section 4.2. Another consequence of not being a spatially explicit model is that each atomic model representing a truck must be connected (through input/output ports) to the shovels and stockpiles it interacts with, as seen in Figure 2.

The model's rationale is as follows: To represent the arrival of a given $truck\ i$ to a $shovel\ j$; the former sends its $truck\_id$ (a positive integer i as its output value) from its $NOTIFY_i$ output port to the input port $TRUCK$ of $shovel\ j$. Likewise, to simulate the loading process of a truck, a non-negative integer output value from $shovel\ j$ (through its $out\_load\ i$ output port) is sent back to $truck\ i$, representing the number of tons of ore loaded into it. On the other hand, to simulate the unloading of the hauled ore on a given stockpile, a truck sends an output value (through its $out\_load\ k$ output port) to the input port $load$ of $stockpile\ k$. Should a shovel be busy upon a truck's arrival, the truck id will be added to an FCFS queue and - eventually – will be serviced. The same logic applies to a truck arriving at an already active stockpile; the only difference is that a stockpile has a serving capacity and can serve several trucks simultaneously.

The time spent by a truck while transiting each route corresponds to a time value used by the corresponding truck as a delay. Finally, it can be observed the presence of an atomic model called Collector, whose input port in_event is connected to all atomic models' output port $DATA$. Its purpose is only to collect values from every shovel, truck, and stockpile atomic model for computing performance metrics about the model.

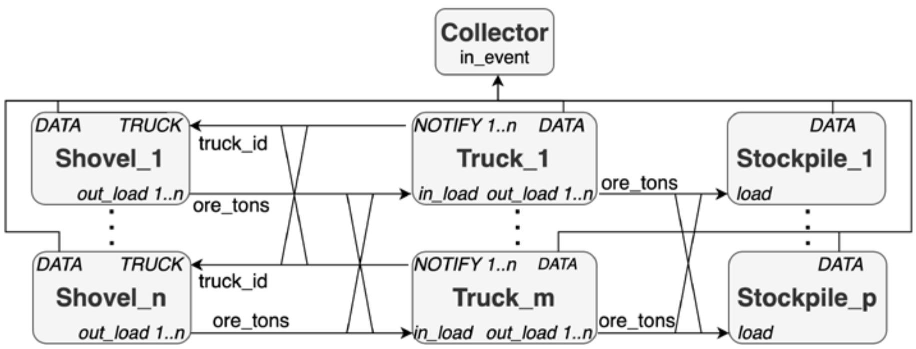

Figure 2: Generalized material handling DEVS model.

The dispatcher logic corresponds to a class method since it only handles the truck-shovel or truck-stockpile assignments logic policy. Its logic is based only on the status (idle or busy) of each shovel and stockpile; its assignments do not alter simulation time directly. As such, any new assignment strategy the user may want to experiment with must be implemented simply by overloading the corresponding class method.

3.2 DEVS Model Definition

As observed, the model is flexible enough to allow users to consider arbitrary amounts of shovels, trucks, and stockpiles. Plus, since each entity is parametrizable, it allows heterogenous machinery fleets.

Using the DEVS formalism, the material handling subprocess can be modelled as a coupled model like:

$$HAULINGPROCESS = < X, Y, D, \{M_D\}, \{I\}, \{Z\}, select >$$

Where:

$X = \emptyset;\ Y = \emptyset;$

$D = \{Shovel_i, Truck_j, Stockpile_k, Collector\}\ \forall\ i \in [1, n]$ where n is the number of shovels, $\forall\ j \in [1, m]$ where m is the number of trucks, $\forall\ k \in [1, p]$ where p is the amount of stockpiles.

$M_{shovel\ 1..n} = Shovel$; $M_{truck\ 1..m} = Truck$; $M_{stockpile\ 1..p} = Stockpile$;

$I = \emptyset;$

$Z = \{((Shovel_i, DATA), (Collector, in\_event));\ \left((Truck_j, DATA), (Collector, in\_event)\right);$
$\left((Stockpile_p, DATA), (Collector, in\_event)\right); \left((Truck_j, NOTIFY_i), (Shovel_i, out\_load_i)\right);$
$\left(\left(Truck_j, out\_load_p\right), \left(Stockpile_p, LOAD\right)\right); \left(Shovel_i, out\_load_j\right), \left(Truck_j, in\_load\right)\};$

$select = \{Truck_j, Shovel_i, Stockpile_p, Collector\}$;

The above coupled model is composed only of atomic models (Shovel, Truck, Stockpile and Collector). Therefore, such atomic models can be formally defined as:

$$SHOVEL = < X_s, Y_s, S_s, \delta_{ext_s}, \delta_{int_s}, \lambda, ta >$$

Where:

$X_s = \{(TRUCK, N_t)\}$ where $TRUCK$ is the input port and N_t represents the ID value of the truck requesting service;

$S_s = \{state \in \{loading, waiting\}, preparationTime, timeLeft \in \mathbb{R}_0+, queue \in \{truck_{id} \in N_t\}^*\}$; where $N_t \in [1, numberOfTrucks]$.

$Y_s = \{(out\_load_i, L)\}$ with $i \in [1, numberOfTrucks]$ corresponds to the output port connecting to the truck whose ID equals i, and $L \in \mathbb{R}+$ represents the amount of ore (in tons) loaded into the truck;

```
δ_ext_s (s, e, x){
    add(truck_id, s.queue)          //A truck's ID was received and added to the shovel's queue.
    if(sizeOf(s.queue) == 1){       //The queue was empty, no other trucks were being loaded.
      state = loading;              //The shovel must start loading.
      ta(state) = loadingTime;      //The shovel starts loading the truck.
    }else{                          //There is a truck being loaded, the arriving truck must wait.
      ta(state) = timeLeft;         //Use time left before loading next (queued) truck.
    }
}

λ (s){                                            //When the shovel stops loading the truck, it sends the
  sendOutput(L, out_load_i, first(queue));        //Outputs the amount of ore loaded into the truck (L) to the
}                                                 //corresponding truck atomic model through the associated
                                                  //out_load_i output port, with i having the value of the truck ID.

δ_int_s(s){
  if(sizeOf(s.queue) > 1){          //If the queue was empty, it means that there are no other trucks being
    state = loading;                // loaded or waiting to be loaded. Therefore, shovel becomes active.
    queue.pop();                    // Removes the truck from the queue.
    ta(state) = loadingTime;        //The shovel starts loading the truck.
  }else{
    state = waiting;                // loaded or waiting to be loaded. Therefore, shovel becomes active.
    passivate();                    //No more trucks are in line to be loaded, wait until another arrives.
  }
}
```

On the other hand, a truck atomic model is formally defined as:

$$TRUCK = < X_t, Y_t, S_t, \delta_{ext_t}, \delta_{int_t}, \lambda, ta >$$

$X_t = \{(in\_load, L_t)\}$ where in_load is the input port and L_t represents the amount of ore (in tons) received from the shovel (loaded onto the truck);

$S_t = \{state \in \{beingLoaded, unloading, hauling, travellingEmpty, readyShovel\},$

$preparationTime, timeLeft \in \mathbb{R}_0+, L \in \mathbb{R}+\}$;

$Y_t = \{(out\_load_i, L_t)\}$ with $i \in [1, numberOfStockpiles]$ corresponds to the output port connecting to the stockpile whose ID equals i, and $L \in \mathbb{R}+$ represents the amount of ore (in tons) unloaded by the truck into the stockpile;

$\delta_{ext_t}\,(s, e, x)\{$
  $if(state == beingLoaded)\{$ //The truck is being loaded by a shovel.
    $L = x.value$; //Stores the received value from the shovel (amount of ore).
    $i = dispatch.getStockpileID()$; //Asks the dispatch the stockpile to dump the ore.
    $state = hauling$; //New state is hauling the ore.
    $ta(state) = haulingTime$; //The time needed to transport the ore to stockpile *i*.
  $\}$
$\}$

$\lambda\,(s)\{$
  $if(state == hauling\ OR\ state == travellingEmpty)\{\ \}$ //do not output;
  $if(state == unloading)\{$
    $sendOutput(ID, L, out\_load_i)$; //Outputs the truck ID and hauled ore to the stockpile *i*.
  $\}$
  $if(state == readyShovel)\{$
    $sendOutput(ID, NOTIFY_i)$; //Outputs the truck Id to the shovel to request to be loaded.
  $\}$
$\}$

$\delta_{int_t}(state)\{$
  $if(state == hauling)\{$
    $state = unloading$; //After hauling the state changes to unloading the ore.
    $ta(state) = unloadingTime$; //The time needed to unload
  $\}$
  $if(state == unloading)\{$
    $state = travellingEmpty$; //After unloading the state changes to travelling empty.
    $ta(state) = travellingTime$; //The time needed to arrive at a shovel.
  $\}$
  $if(state == travellingEmpty)\{$
    $state = readyShovel$; //After travelling, the truck asks the shovel to be loaded.
    $ta(state) = readyTime$; //The time needed to indicate that the truck is ready to be loaded.
  $\}$
  $if(state == readyShovel)\{$
    $state = beingLoaded$; //After being ready to be loaded, the truck changes to being loaded.
    $passivate()$; //Waits for the shovel to finish loading the truck (might be queued).
  $\}$
$\}$

The formal definition of the stockpile and collector models is not included for lack of space. However, the formal definition of the stockpile is similar to the definition of the shovel. They only differ because the stockpile outputs each unloaded amount of ore to the collector. On the other hand, the collector's definition is trivial as it only receives values used for computing performance statistics on the simulation model.

3.3 Model Implementation

The model presented in Section 3.1 was implemented using PyPDEVS (Van Tendeloo and Vangheluwe 2014), a Python-based DEVS/PDEVS simulation framework. The implementation corresponds to the definition of four atomic models (truck, shovel, stockpile, and collector). Also, an additional Python class representing the *dispatcher* is included. However, since it only computes the next target of a truck (shovel or stockpile) and does not alter the simulation time, it was not represented as an atomic DEVS model.

The number and parameters of trucks, shovels, stockpiles, and shovel-to-stockpile routes are not hard coded, nor are the input-output port connections among entities. Instead, they are dynamically created from the parameters in a JSON configuration file.

The interaction among atomic models works as follows. The loading of a truck by a shovel starts when a truck atomic model is in state $readyShovel$. First, the truck obtains a message from the class dispatcher with the ID of the target shovel. Then, the truck sends its ID to the shovel through the corresponding output port using its output function in state $readyShovel$. Next, the shovel atomic model activates its external function and queues the received ID, changing its state to loading. If there are no other trucks' IDs queued, it activates its internal transition function, and then it advances time (representing the loading of a truck).

Finally, the shovel sends the number of tons loaded into the truck using its output function through the corresponding output port. The interaction between a truck and a stockpile works similarly.

The definition of a simulation scenario if made on a simple JSON file like the following:

```
{
"trucks" : 6,"shovels" : 2, "stock_piles" : 3, "simTime":3600,
"shovelToStock" : [ [0,1], [1,2] ]
}
```

The simulation consists of 6 trucks, 2 shovels, 3 stockpiles and will last 3600 seconds. Each subarray from the field *shovelToStock* contains the ID of the connected stockpiles, that is, it defines the internal coupling among the ports of trucks-shovels and trucks-stockpiles. Thus, shovel ID 0 has routes to stockpiles with ID 0 and 1, and shovel ID 1 is connected with stockpiles 1 and 2.

4 MINERAL HANDLING: CASE STUDIES

4.1 Experiment Parameters

Most open-pit mining companies do not openly expose the details of their daily chores, as it is a paramount strategic asset. However, a small set allows only a few researchers to analyze their data after signing NDAs (non-disclosure agreements), forbidding them from revealing the name of the corporation or from publishing raw data about their daily activities but just aggregated data. During our literature review, we found the work of (González-Gazmuri 2016) that published a few crucial parameters and measures obtained from an important Chilean copper company (shown in Table 1) along with actual utilization levels of shovels and trucks. The validation of our model is based on this data.

Table 1: Simulation parameters for validation applied to the proposed DEVS simulation model, values obtained from (González-Gazmuri 2016).

Machinery	Function	State	Parameter	Measure Unit
Shovels	Time advance	Loading	$\ln(X) \sim \mathcal{N}(4.41, 2.54^2)$	Minutes

	Output	Loading	$\ln(X) \sim \mathcal{N}(331, 16.06^2)$	Tons.
Trucks	Time advance	Unloading	$\ln(X) \sim \mathcal{N}(1.84, 1.38^2)$	Minutes
	Time advance	Hauling	$\ln(X) \sim \mathcal{N}(34.1, 21.9^2)$	Minutes
	Time advance	Travelling empty	$\ln(X) \sim \mathcal{N}(26.8, 19.0^2)$	Minutes

In the following subsections, first, we validate our model to ensure it predicts valid and meaningful results. Then we simulate scenarios to show this tool's power to determine the fleet size needed for achieving a given material handling objective during a 12-hour shift.

4.2 Validation

For the present validation, we focus on shovel and trucks utilization levels, simulating scenarios using a single shovel, three stockpiles, and 5 to 14 trucks, with the parameters presented in Table 1. As no details about the dispatch were provided, it was assumed a round-robin assignment of truck-shovel and truck-stockpile.

Table 2 shows a comparison among actual truck utilization levels (González-Gazmuri 2016) and results reported by our DEVS simulation model. The difference shows a mean square error of 2.96, indicating a remarkable accuracy of the proposed model. In fact, the difference is explained due to the lack of details of the real observations as they do not report unplanned fails nor pauses made by the machinery operators.

Table 2: Comparison of real and simulated truck utilization levels.

Truck Qty	5	6	7	8	9	10	11	12	13	14
Real Utilization (%)	98.1	95.5	93.9	90.3	85.5	80.4	74.8	69.5	64.7	60.4
Simulated Utilization (%)	95.5	93.9	92.37	89.7	86.2	83.2	78.4	74.1	68.8	63.9

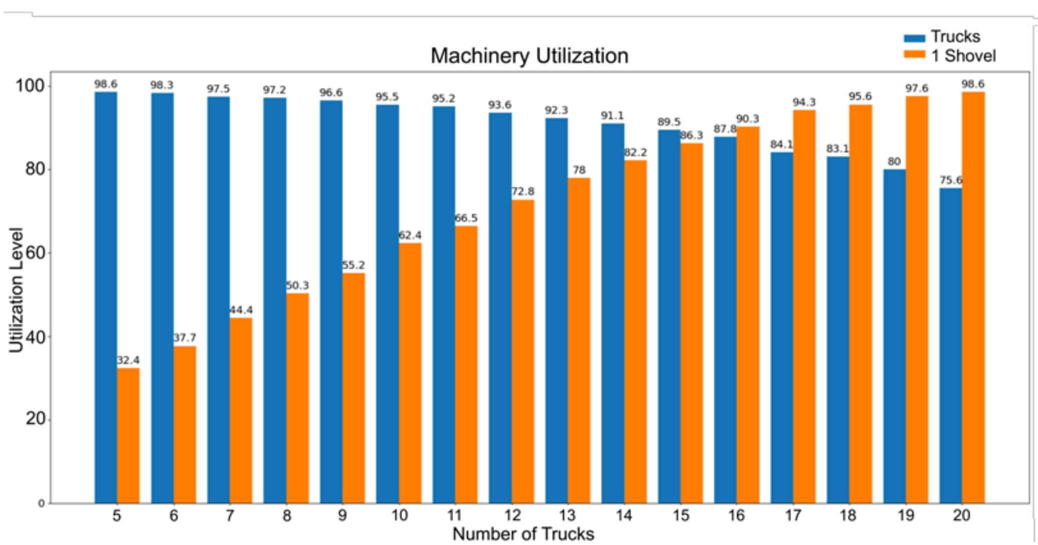

Figure 3: Machinery utilization using one shovel (baseline scenario).

4.3 Model Application: Determining the Fleet Size

Using our DEVS simulation model, we can execute a variety of experiments with ease. This way, modelers can explore multiple scenarios to estimate the effects of different actions. As an illustrative application, we propose a brute-force way of determining the adequate fleet size for different ore transportation goals for a given 12-hour shift.

Figure 3 and Figure 4 show machinery utilization levels for simulated scenarios with one and two shovels (respectively), considering three stockpiles and 5 to 20 trucks. It is clear that by increasing the number of shovels from 1 to 2, its utilization decreases since they distribute their attention to trucks. By simply using two shovels, their utilization decreases from 32% to 16% (with five trucks hauling material), and from 98% to 63% with 20 trucks operating. On the other hand, when using just one shovel (Figure 3) and increasing from 15 to 16 trucks, their utilization dramatically decreases as they spend more time queuing.

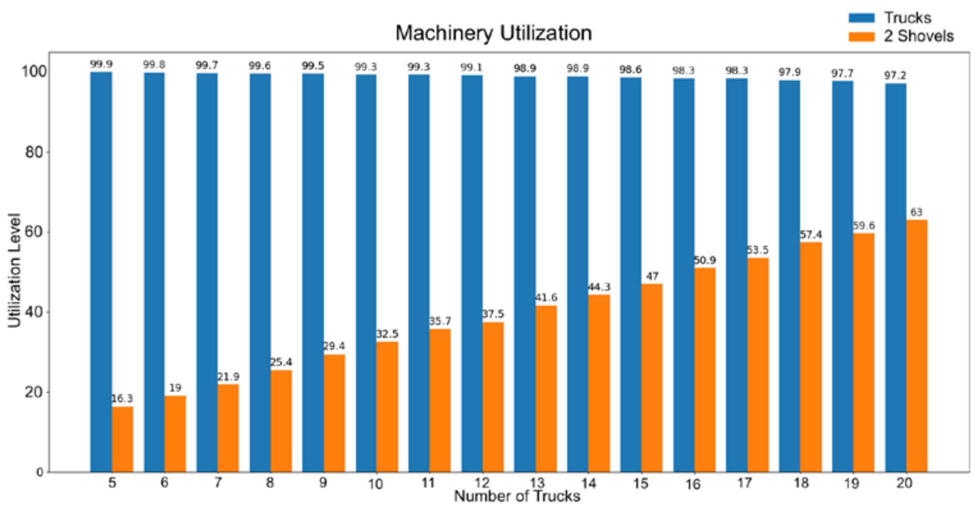

Figure 4: Machinery utilization using two shovels.

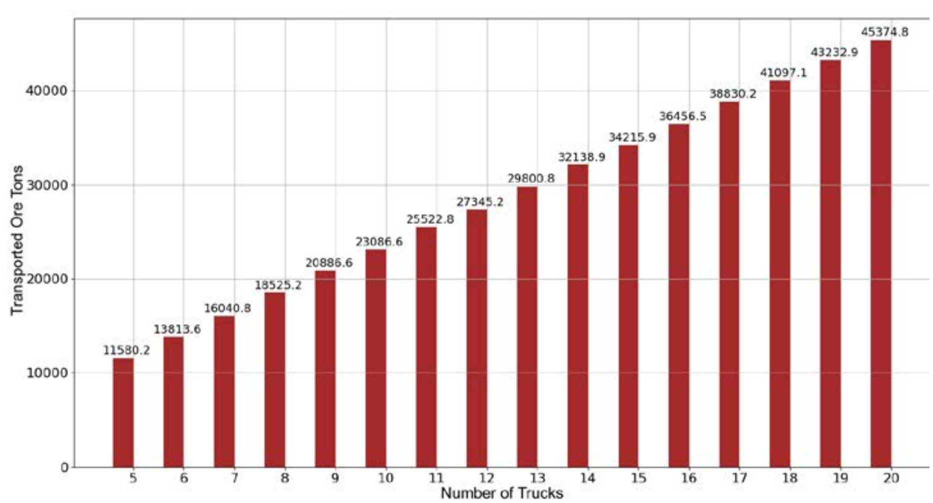

Figure 5: Simulated transported ore using two shovels and different number of trucks in a 12-hour shift.

Following this and considering the chart of transported ore in Figure 5, dispatchers can forecast the amount of machinery (without overestimating it) needed to accomplish their goals of transported material.

For example, if the objective is to haul 33,000 tons, the number of trucks working during that shift must be 15. In addition, shovels should be at least 2 for supporting fault tolerance, as they are a critical resource, and their failure would stop the whole material handling subprocess.

5 CONCLUSIONS

We have presented a DEVS model of the loading and transport subprocess of an open pit mine implemented in PyPDEVS. Using a formal modeling and simulation approach allows the modelers to focus on modeling aspects without worrying about the simulation's low-level complexities, which are handled by the tool.

Furthermore, the model enables dispatch personnel to define new and fully parameterizable scenarios quickly and easily. Also, it is flexible enough to be extended with new truck assignment policies and to add new kinds of machinery. The resulting DEVS model has been used to simulate different scenarios for determining the fleet size given a material handling goal without overestimating the amount of equipment or, on the other hand, to forecast the amount of transported ore with a given fleet size.

Despite the lack of actual (publicly available) data on the material handling subprocess of open pit mines, we validated our DEVS model using parameters in the literature, obtaining good precision.

In future work, we intend to model time-changing open pit layouts (with extraction zones or deposits moving over time). Also, to consider realistic details that may affect overall hauling goals, such as planned/unplanned machinery maintenance for allowing to study the effects of unavailable equipment due to failures, operator-induced delays (due to biological needs), and different truck assignment strategies. Finally, we aim to test the simulator performance to evaluate a possible integration with actual real-time values from open-pit mines to produce a digital twin.

ACKNOWLEDGMENTS

This work has been partially supported by Fondecyt de Iniciación 11230961 from ANID, Chile, and by INICI-UV UVA 20993 program from Universidad de Valparaíso, Chile.

REFERENCES

Alarie, S. and M. Gamache 2010. "Overview of Solution Strategies Used in Truck Dispatching Systems for Open Pit Mines." *International Journal of Surface Mining, Reclamation and Environment* 16 (1):59-76.

Bajany, DM, X. Xia, and L. Zhang 2017. "A MILP Model for Truck-shovel Scheduling to Minimize Fuel Consumption". *Energy Procedia* 105:2739-2745.

Bozorgebrahimi, E., R.A. Hall, and G.H. Blackwell 2003. "Sizing Equipment for Open Pit Mining - A Review of Critical Parameters". *Mining Technology* 112 (3):171-179.

Burt, Christina N., and L. Caccetta 2014. "Equipment Selection for Surface Mining: A Review". Interfaces 44 (2):143-162.

Chaowasakoo, P., H. Seppälä, H. Koivo, and Q. Zhou 2017. "Improving Fleet Management in Mines: The Benefit of Heterogeneous Match Factor". *European Journal of Operational Research* 261(3):1052-165.

Chow, A. C. H., and B. Zeigler 1994. "Parallel DEVS: A Parallel, Hierarchical, Modular Modeling Formalism". In *Proceedings of Winter Simulation Conference*, 1994. Lake Buena Vista, FL, USA, 716-722.

Cleveland, C. J., and C. Morris 2015. *Dictionary of Energy*, 2nd ed. Elsevier.

Curry, J., M., J.L. Ismay, and G. J. Jameson 2014. "Mine Operating Costs and the Potential Impacts of Energy and Grinding". *Minerals Engineering* 56:80-80.

da Cunha Rodovalho, E., H. Mota Lima, and G. de Tomi 2016. "New Approach for Reduction of Diesel Consumption by Comparing Different Mining Haulage Configurations". *Journal of Environmental Management* 172:177-185.

Fioroni, M. M., L. A. G. Franzese, T. J. Bianchi, L. Ezawa, L. Ricardo Pinto, and G. de Miranda 2008. "Concurrent Simulation and Optimization Models for Mining Planning." In *Proceedings of the Winter Simulation Conference 2008*, Miami, FL, USA.

Fonseca, P. 2009. "Towards an Automatic Transformation from a DEVS to a SDL Specification." In *Proceedings of the 2009 Summer Simulation Multiconference*, 348-353.

Gil-Costa, V., A. Inostrosa-Psijas, and M. Marin 2020. "Elastic and Real-time Capacity Planning for Web Search Engines". In *28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, Västerås, Sweden.

González-Gazmuri, R. I. 2016. *Optimización de las Horas Operativas de los Caex en los Procesos de Descarga en Chancado y Abastecimieno de Combustible, Mediante la Utilización de Modelamientos Compuestos y Redes Neuronales*. Master Thesis, Universidad de Chile, Santiago, Chile. https://repositorio.uchile.cl/handle/2250/141629, accessed 24th December 2023.

Icarte, G., J. Diaz, and O. Herzog 2021. "A Dynamic Scheduling Multiagent System for Truck Dispatching in Open-Pit Mines". In *International Conference on Agents and Artificial Intelligence 2021 (ICAART)*, Valetta, Malta, 132-148.

Icarte, G., P. Berrios, R. Castillo, and O. Herzog 2020. "A Multiagent System for Truck Dispatching in Open-Pit Mines". In *International Conference on Dynamics in Logistics 2020*, Bremen, Germany, 363-373.

Marin, M., V. Gil-Costa, A. Inostrosa-Psijas, and C. Bonacic 2019. "Hybrid Capacity Planning Methodology for Web Search Engines". *Simulation Modelling Practice and Theory* 93:148-163.

Mena, R., E. Zio, F. Kristjanpoller, and A. Arata 2013. "Availability-Based Simulation and Optimization Modeling Framework for Open-Pit Mine Truck Allocation Under Dynamic Constraints". *International Journal of Mining Science and Technology* 23(1):113-119.

Meng, C., S. Srinivas Nageshwaraniyer, A. Maghsoudi, Y.J. Son, and S. Dessureault 2013. "Data-Driven Modeling and Simulation Framework for Material Handling Systems in Coal Mines". *Computers & Industrial Engineering* 64 (3):766-779.

Mohtasham, M., H. Mirzaei-Nasirabad, H. Askari-Nasab, and B. Alizadeh 2021. "Truck Fleet Size Selection in Open-Pit Mines Based on the Match Factor Using a MINLP Model". *Mining Technology* 130 (3):159-179.

National Productivity Commission. 2017. “Productivity in the Chilean Copper Mining Industry”. Comisión Nacional de Productividad.

Open-Pit DEVS Model GitHub repository. https://github.com/JoelSR/Carguio-y-Transporte-PyPDEVS, accessed 7th July 2023

SONAMI 2021. Sociedad Nacional de Minería, Chile. https://www.sonami.cl/v2/informacion-de-la-mineria/, Accessed 21st December 2022.

Soumis, F., J. Ethier, and J. Elbrond 1989. "Truck Dispatching in an Open Pit Mine". *International Journal of Surface Mining, Reclamation and Environment* 3(2):115-119.

Ta, C. H., A. Ingolfsson, and J. Doucette 2013. "A Linear Model for Surface Mining Haul Truck Allocation Incorporating Shovel Idle Probabilities". *European Journal of Operational Research* 231(3):770-778.

Tan, Y., and S. Takakuwa 2016. "A Practical Simulation Approach for an Effective Truck Dispatching System of Open Pit Mines Using VBA". In *2016 Winter Simulation Conference (WSC)*, Washington, DC, USA, 2394-2405.

USGS 2012. U.S. Geological Survey, Mineral Commodity Summaries 2021. https://pubs.usgs.gov/periodicals/mcs2021/mcs2021-copper.pdf, accessed 7th July 2022.

Upadhyay, S.P., and H. Askari-Nasab 2016. "Truck-Shovel Allocation Optimisation: A Goal Programming Approach". *Mining Technology* 125(2):82-92.

Upadhyay, S.P., and H. Askari-Nasab 2018. "Simulation and Optimization Approach for Uncertainty-Based Short-Term Planning in Open Pit Mines". *International Journal of Mining Science and Technology* 28(2):153-166.

Van Tendeloo, Y., and H. Vangheluwe 2014. "The Modular Architecture of the Python (P) DEVS Simulation Kernel". In *Proceedings of the 2014 Symposium on Theory of Modeling and Simulation - DEVS Integrative*, Tampa, FL, USA, 387-392.

Wainer, G. A. 2017. *Discrete-event Modeling and Simulation: A Practitioner’s Approach*. Florida: CRC press.

Zeigler, B., T. G. Kim, and H. Praehofer 2000. *Theory of Modeling and Simulation*. Academic Press, Inc.

Zhang, L., and X. Xia 2015. "An Integer Programming Approach for Truck-Shovel Dispatching Problem in Open-Pit Mines". *Energy Procedia* 75:1779-1784.

AUTHOR BIOGRAPHIES

JOEL SANTANA obtained his civil informatics engineering and computer science degree from Universidad de Valparaíso. His research interest is discrete-event simulation applied to the mining industry. His email address is joel.santana@alumnos.uv.cl.

ALONSO INOSTROSA-PSIJAS received the Ph.D. degree from Universidad de Santiago, Chile. He is an Associate Professor at the School of Informatics Engineering at Universidad de Valparaiso, Chile. His research interests are discrete-event and parallel/distributed simulation. He can be contacted at alonso.inostrosa@uv.cl.

FRANCISCO MORENO received the Ph.D. degree from Universidad de Santiago, Chile. He is an Associate Mathematics and Computer Science Professor at Universidad de Santiago, Chile. His research interests are information theory and multivariate statistics and their applications. He can be contacted at francisco.moreno@usach.cl.

MAURICIO OYARZÚN received the Ph.D. degree from Universidad de Santiago, Chile. He is an Assistant Professor at the Faculty of Engineering and Architecture at Universidad Arturo Prat, Chile. His research interests include information retrieval, index compression, and data structures for discrete-event simulation. He can be contacted at moyarzunsil@unap.cl.

GABRIEL WAINER received the Ph.D. degree from Université d’Aix-Marseille III. He is a Full Professor in Carleton University. His current research interests are related with modeling methodologies and tools, parallel/distributed simulation, and real-time systems. His e-mail is gwainer@sce.carleton.ca.