An Agent-based System for Truck Dispatching in Open-pit Mines

Gabriel Icarte Ahumada[1,2] [a], Eduardo Riveros[2] and Otthein Herzog[3,4,5] [b]

[1]*International Graduate School for Dynamics in Logistics (IGS), University of Bremen, Bremen, Germany*

[2]*Faculty of Engineering and Architecture, University Arturo Prat – UNAP, Iquique, Chile*

[3]*Center for Computing and Communication Technologies - TZI, University of Bremen, Germany*

[4]*Jacobs University Bremen, Germany*

[5]*Tongji University Shangai, P.R. China*

Keywords: Truck Dispatching, Open-pit Mine, Multiagent Systems, Scheduling.

Abstract: An important logistic process in open-pit mines is material handling due to its high operational costs. In this process shovels extract and load materials that must be transported by trucks to different destinations at the mine. Several centralized systems have been developed to support this process. The methods applied for these systems are based on mathematical programming, heuristic processes or simulation modelling. The main disadvantages in these systems are performing calculations in a timely manner, addressing the dynamics of a mine, and not being able to provide a precise dispatching solution. In this paper, we describe a distributed approach based on Multiagent Systems (MAS). In this approach, the real-world equipment items such as shovels and trucks are represented by intelligent agents. To meet the target in the production plan at minimal cost, the agents must interact with each other. For this interaction, a Contract Net Protocol with a confirmation stage was implemented. To evaluate the MAS, an agent-based simulation with data from a Chilean open-pit mine was used. The results show that the MAS provide more precise solutions than the current centralized systems in a practical calculation timeframe. In addition, the MAS decreases the truck costs by 20% on average.

1 INTRODUCTION

An important logistic process in open-pit mines is material handling, since it can account for up to 50% of the operational cost (Alarie and Gamache, 2002). In this process, the most important equipment items are shovels and trucks. These equipment items must work together to extract and to transport all the material required in the operational plan at minimum cost. If the extracted material is waste, it must be transported to a waste dump, and if it is ore, it must be transported to a crusher or a stockpile. Figure 1 shows all the activities that a truck must perform to transport materials from a loading point to an unloading point. This is called the truck cycle. This cycle is performed and repeated by each truck until the shift ends.

To achieve efficient material handling, an important decision must be made each time that a truck finalizes an unloading activity: Where does this truck have to go now? The answer to this question is not easy due to the number of the involved variables in the process and the dynamics of the environment where the equipment items operate.

In the last decades, different centralized systems have been implemented to support truck dispatching in open-pit mines based on mathematical programming, heuristic processes or simulation. The strengths of these methods are their maturity and their well-known implementation. However, the weaknesses can be observed in addressing the dynamics of a mine (Bastos et al., 2011), not being able to provide a precise solution (Patterson et al., 2017), using estimated information (Chang et al., 2015; Costa et al., 2005; Krzyzanowska, 2007; Newman et al., 2010), and obtaining a dispatching solution in a timely manner when the model is too complex.

One common strategy applied in some centralized systems is based on the multistage approach (Alarie and Gamache, 2002). This approach uses a guideline that is computed in the upper stage. Afterwards, this guideline is used by the lower stage as a refer-

[a] https://orcid.org/0000-0002-1997-0053

[b] https://orcid.org/0000-0003-4781-2551

Icarte Ahumada, G., Riveros, E. and Herzog, O.
An Agent-based System for Truck Dispatching in Open-pit Mines.
DOI: 10.5220/0008961800730081
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 1*, pages 73-81
ISBN: 978-989-758-395-7; ISSN: 2184-433X

ence to make real-time dispatching decisions. For instance, most centralized systems determine the number of trips to transport the required material in the production plan from a loading point to an unloading point. They do this before the machines work. Then, when the trucks are working, and when one of them finalize an unloading activity, the centralized system selects and provides a new destination. To do this, the centralized system considers some criteria, such as the number of trips done between a loading and unloading point, the distance to a loading point or to the loading point with the least production, among others. This is also called an allocation model.

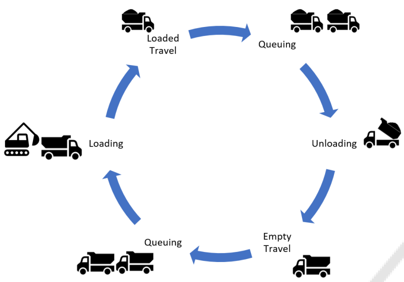

Figure 1: The truck cycle.

Despite the use of these systems, the trucks and shovels do not operate efficiently, since queues of trucks form in front of shovels and crushers, and there are also shovel idle times. Therefore, the question remains as how to improve the efficiency in the material handling process.

Alternatively, a better solution that would allow the equipment items to operate more efficiently would be to set up schedules for each equipment item. The schedules would contain all the activities that the equipment items must perform, pointing out the start times, end times, etc.

In this paper, we present a distributed solution based on Multiagent Systems (MAS) that provide more precise solutions than current systems. In order to demonstrate the validity of the proposed MAS, a comparison against a centralized dispatching algorithm and actual data from a Chilean open-pit mine was performed.

The remainder of this paper is structured as follows: Section 2 presents related work. Section 3 presents the distributed solution based on MAS. The dispatching algorithm used for the evaluation of the MAS is described in section 4. Section 5 presents the evaluation and discussion of the proposed MAS in a case study. Finally, the conclusions and outlook are presented in section 6.

2 RELATED WORK

There are several articles that deal with truck dispatching in open-pit mines. These articles show different approaches that try to achieve two goals: improve productivity and reduce operating costs (Alarie and Gamache, 2002). These approaches use a centralized strategy based on methods from operations research, simulation modelling or heuristic procedures. Operations research methods are most commonly used. For instance, (Ahangaran et al., 2012) proposed a real-time dispatching model considering trucks with different capacities. The model uses two methods: flow networks and integer programming. (Ercelebi and Bascetin, 2009) employed a closed queuing network theory for the allocation of trucks and linear programming for the purpose of truck dispatching to shovels.

Simulation modelling is also a method often employed. For instance, (Hashemi and Sattarvand, 2015) used simulation modelling in their work taking into account the match factor indicator, which reflects the relationship between shovel and truck productivity. (Jaoua et al., 2012) presented a simulation framework incorporating a traffic simulator with a classic discrete event simulation model of internal transport systems.

Another method used in recent years is related to heuristic procedures. These methods have been quite popular in the literature because they are easy to implement and do not require much computation when making dispatching decisions, which is important when decisions must be taken in real-time. For example, (Alexandre et al., 2015) presented a comparison of results obtained by different evolving algorithms for the Multi-objective Open-Pit Mining Operational Planning Problem. (Mendes et al., 2016) used a multi-objective genetic algorithm.

Most of these works provide a method that determines the next destination of the truck when the latter ends an unloading activity. To do this, they take into account a previously calculated guideline and the current status of the mine. These methods are not able to provide a precise description of the activities of shovels and trucks. Therefore, they cannot guarantee a good synchronization between the activities of the equipment items (Patterson et al., 2017).

In order to address the problem mentioned, two previous papers proposed using schedules for all the activities that the shovels and trucks must perform (Chang et al., 2015; Patterson et al., 2017). To do this, they used a dispatching algorithm and a metaheuristic method to generate the schedules. Their results showed that these algorithms generated schedules for different size instances with good results and performance in practical frametimes.

3 MULTIAGENT SYSTEM FOR TRUCK DISPATCHING IN OPEN-PIT MINES

To the best of our knowledge, there is no evidence that a multiagent system has been applied to truck dispatching in open-pit mines. However several articles show the use and applicability of MAS in the transport domain. (Chen and Cheng, 2010) performed a literature survey on MAS applied for the transport domain. The research results clearly demonstrate the potential of using agent technology to improve the performance of traffic and transportation systems.

As was mention before, an aspect that is not addressed very well for the centralized approaches is the dynamics of a mine. Equally significant, the dynamic of the environment plays an important role in MAS, since changes in the environment affect the agents. Several studies have considered this issue. For example, (Fischer et al., 1996) considered traffic congestion and its effect on the delivery process. (Gath, 2015; Mes et al., 2007) considered the dynamic environment by the arrival of orders for delivery / collection of products at any time. (Chen and Cheng, 2010) pointed out that the purpose of introducing mobile agents into traffic and transportation systems is to increase the flexibility and the ability of the system to deal with uncertainty in a dynamic environment.

Despite the fact that MAS has not been applied to truck dispatching in open-pit mines, we believe that MAS is a suitable alternative to be applied in the context of an open-pit mine. This is because of the evidence of the application of MAS in the transport domain and the addressing of the dynamics of the environment.

The objective of the developed MAS is to accomplish the targets of the production plan at the minimum cost. To do this, the agents interact with each other to create schedules for each equipment item that they represent. Here, an agent represents only one equipment item from the real-world. Table 1 shows the implemented agents, their objectives and properties.

Our multiagent system for truck dispatching in open-pit mines offers several advantages over current centralized systems:

- Efficient dispatching solution. Using schedules organizes the machine activities more efficiently than an allocation model. In addition, the use of specific data from equipment items generates the schedules more precisely.
- Robustness. Due to the fact that there is not a central node, if any node faults, the system continues working.
- Flexibility. An agent can change its behaviour to adapt to the new conditions in its environment and, in this way, to achieve its own objectives or the objectives of the entire system.

3.1 Interaction in the Scheduling MAS

In order to create the schedules, the agents must interact with each other. To do this, a Contract Net Protocol (CNP) (Smith, 1980) is implemented. In the context of the MAS for truck dispatching, the CNP works as follows: a *shovelAgent* initiates a negotiation with the *truckAgents* sending them a call for proposals (CFP). The CFP points out the time when the shovel is available to load a truck and the idle time from the last loading. When a *truckAgent* receives a CFP, it must decide whether or not to send a proposal. To do this, it asks the unloadingAgent about the prospective waiting time and, with the information of the CFP, it determines whether the offer fits in its schedule. If yes, it prepares a proposal, sends it to the *shovelAgent* and waits for the answer. If not, it sends a rejection to the *shovelAgent*.

When the *shovelAgent* receives all the proposals, or when the deadline is expired, it looks for the best proposal. Then, the *shovelAgent* sends an acceptance message to the *truckAgent* that proposed the best proposal and sends a rejection message to the other *truckAgents* that sent a proposal. After this, the *truckAgent* that receives the acceptance message and the *shovelAgent* that initiated the negotiation add to their schedules all the activities that must be performed and the negotiation is then finished. The *shovelAgent* repeats this protocol until the target of the production plan is met or until it reaches the end of the shift. In the case that a *shovelAgent* ends a negotiation without a winner, it starts a new negotiation adding one minute to the loading time offered in the last negotiation.

Given that the agents work in parallel, several CNP negotiations are performed concurrently. Therefore, a *truckAgent* may receive several CFPs. If the *truckAgent* is taking part in a previous negotiation and it is still waiting for the answer from a *shovelAgent*, the other received CFPs will be rejected. In this context, it could happened that the *truckAgent* rejects a CFP that is a better option than the CFP answered previously. This problem is also called "the eager bidder problem" (Schillo et al., 2002). To tackle this problem, a confirmation stage was included in the original CNP: when the *shovelAgent* finalizes the evaluation of the proposals, it sends a confirmation message to the *truckAgent* with the best proposal. The *truckAgent* that receives the confirmation message accepts

Table 1: Agent description.

Agent	Real-world representation	Objective	Properties
truckAgent	Trucks	To create a schedule of the activities of the truck at minimum cost	Capacity, loaded velocity, empty velocity, spotting time and unloading time, layout of the mine.
shovelAgent	Shovels Front loader	To create a schedule of the activities of the equipment that it represents considering its target in the production plan	Capacity, dig velocity, load velocity and the destination of extracted material
unloadingPointAgent	Crushers Stockpiles Waste dumps	Its objective is to create a schedule of the activities of the equipment that it represents	Number of trucks unloading simultaneously

the confirmation if it has not received a better CFP from another *shovelAgent*. Otherwise, it rejects the confirmation message. If the *shovelAgent* receives a rejection of the confirmation message, it sends a new confirmation message to the *truckAgent* that sent the second-best proposal. The *shovelAgent* will continue sending confirmation messages until it receives an acceptance of the confirmation or until there are no more proposals. In this way, the *truckAgent* might decommit a previous proposal sent. Figure 2 depicts the interaction between the agents using the CNP with a confirmation stage. Table 2 shows a schedule example for a truck created by the MAS using this protocol.

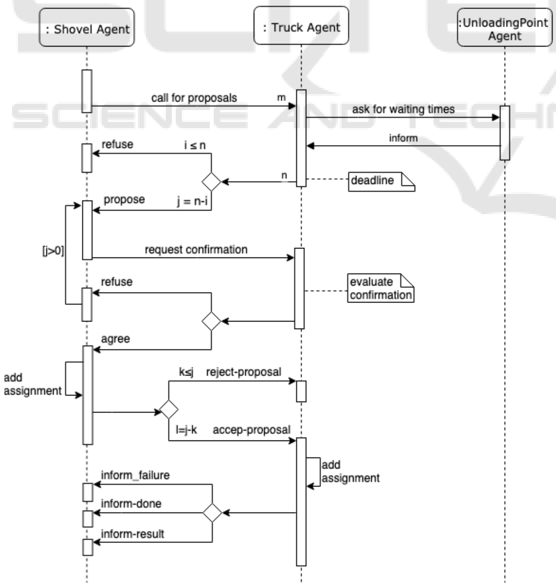

Figure 2: The interaction between the agents using the CNP with the confirmation stage.

3.2 Decision Making

The decision-making process of an agent is an important characteristic. A bad design could cause the agent to make a wrong decision or take too much time to make it. This could also affect the performance of the whole system. In the MAS for truck dispatching in open-pit mines, the *shovelAgents* and *truckAgents* make the main decisions in the system.

A *shovelAgent*, after receiving all the proposals (or when the deadline is expired), must evaluate all the proposals using a utility function. This function promotes the proposals that propose to start the loading on time and with the least time to perform all the activities.More formally:

$Offer = number \in \{0\} \cup \mathbb{N}$. It represents the time offered by a shovel to start a loading
P = set of received proposals $p_1 \ldots . p_n$
$p_i = (arrivalTime, cost) | arrivalTime \in \{0\} \cup \mathbb{N}, cost \in \{0\} \cup \mathbb{N}$

The decision is a multicriteria problem as

$Decision = arg_{min}(p_i \in P)\{U(p_1), .., U(p_n)\}$
$U = U(arrivalTime)' + U(cost)'$
$U(cost) = cost$
$\delta = (arrivalTime – offer) \in \mathbb{Z}$
$U(arrivalTime) =$

$$\begin{cases} \delta, \text{ if } \delta < 0 & \text{(The truck arrives earlier)} \\ 0, \text{ if } \delta = 0 & \text{(The truck arrives just on time)} \\ 2 * \delta, \text{ if } \delta > 0 & \text{(The truck arrives later)} \end{cases}$$

Where $U(arrivalTime)'$ is the normalized value of $U(arrivalTime)$ and $U(cost)'$ is the normalized value of $U(cost)$. The normalization formula applied is

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

A *truckAgent* must decide whether the offer received can be performed by the truck. To do this, the *truckAgent* looks for a free time slot considering the loading time offered by the *shovelAgent*. If it finds one, the *truckAgent* estimates the total time that the

Table 2: Example of schedule created for a truck.

Assignment	Destination	Start Time of the Trip	Arrival Time	Start Time of the Spotting	Start Time of the Loading or Unloading	End Time of the Assignment
0	Shovel.01	00:47:01	01:20:23	01:20:23	01:21:36	01:23:12
1	WasteDump.02	01:23:12	01:32:33	01:32:33	01:32:33	01:33:23
2	Shovel.04	02:10:39	02:18:47	02:18:47	02:20:00	02:21:12
3	WasteDump.03	02:21:12	02:26:38	02:26:38	02:26:38	02:27:28
4	Shovel.04	02:27:28	02:31:37	02:31:37	02:32:50	02:34:02
5	WasteDump.03	02:34:02	02:39:28	02:39:55	02:39:55	02:40:45

truck spends to perform all the activities. Then, it determines whether the offer is suitable for the free time slot. If there is not a free time slot, the *truckAgent* sends a rejection message to the *shovelAgent*.

Another decision by a *truckAgent* is made during the confirmation stage. Here, the *truckAgent* must decide whether it accepts the confirmation message or rejects it. In the original CNP, we mentioned that a *shovelAgent* would add one minute if the negotiation ends without a winner. In the confirmation stage, it was mention that a *truckAgent* would reject a confirmation message if it has received a better CFP (a better CFP is that one that the truck can perform at a lower cost). If this situation is repeated continually (negotiation of a *shovelAgent* without a winner), it will generate idle time in the schedule of the shovel.

To tackle this situation, the *truckAgents* consider the shovel idle time since the last loading. If this time is less than one minute, the *truckAgent* can reject the confirmation message. Nevertheless, if the shovel idle time is higher or equal to one minute, the *truckAgent* must accept it. Thus, the *truckAgent* chooses to achieve the target of the production plan instead of decreasing its own cost.

4 A DISPATCHING ALGORITHM

In order to evaluate the schedules generated by the MAS, we developed a dispatching algorithm called Disp-ALG. Our algorithm uses the following inputs:

- Travel Times Matrix. These are the travel times that the trucks need to travel from one point to another point at the mine.
- Production plan. This points out the amount of material that must be extracted by the shovels and transported to a destination.
- Equipment data. This is information about available shovels and trucks, such as capacity and velocity.

Table 3: Dispatching algorithm.

Name	Description
S	Set of shovels (index s)
T	Set of trucks (index t)
J	Set of destination of the material extracted by shovel s (index j)
C_s	Loading time of the shovel s
C_j	Unloading time at the destination j
C_j^s	Travel time from destination j to shovel s
C_s^j	Travel time from shovel s to destination j

The algorithm output is a schedule of the activities that the equipment items must perform. The notation of sets, indices, parameters and variables used throughout the algorithm is shown in Table 3. More formally:

$truckActivity = \{\text{"}emptyTrip\text{"} \vee \text{"}loading\text{"} \vee \text{"}loadedTrip\text{"} \vee \text{"}unloading\text{"}\}$
$plan = \{(s, j, x) | s \in S, j \in J, x \in \mathbb{N}$
$shovelSchs = \{(s, startTime, endTime) | s \in S, startTime \in \mathbb{N}, endTime \in \mathbb{N}, t \in T\}$
$truckSchs = \{(t, truckActivity, startTime, endTime, from, to) | t \in T, activity \in truckActivity, startTime \in \mathbb{N}, endTime \in \mathbb{N}, from \in (S \cup J), to \in (S \cup J)\}$

The Main algorithm (algorithm 1) consists of 3 steps: the first one looks for a shovel task that is defined as (s, j, tll). The second step looks for the best truck t to perform the shovel task. If the second step is successful, the algorithm proceeds with the third step, which consists of adding the activities and times to the schedules of the truck t and shovel s; otherwise the shovel task is discarded. These steps are repeated while the operational targets of each shovel have not been met or until the last activity of the shovels exceeds the end of the shift. To determine whether these conditions are met, the algorithm invokes the boolean function ISCONDITIONSMEET.

The function FINDJOB (algorithm 2) is a function that returns a shovel task. To do this, the algorithm selects a shovel s (SELECTSHOVEL) in the production

Algorithm 1: Main.

Input: S, T, Plan, H
Output: shovelSchds, truckSchds

$shovelSchds \leftarrow \emptyset; truckSchds \leftarrow \emptyset$;
while not isConditionsMeet **do**
  $s, j, tll \leftarrow$ FindJob$(S, shovelSchds, H)$
  if s **not** null **then**
    $T, act, act' \leftarrow$ FindTruck$(T, s, j, tll, truckSchds)$
    if t **not** $null$ **then**
      Schedule(t,s,j,tll,shovelSchds, truckSchds, act, act')
    end if
  end if
end while

plan whose target has not been met. Then, it gets the destination j of the material extracted by shovel s (getDestination), its schedule (getSchedule) and the end time of the last loading activity tll (getEndTimeLastLoading) of the shovel s. Finally, it returns s, j, and tll. The next time that the function is invoked, the function selects another shovel following a round-robin method.

Algorithm 2: FindJob.

Input: S, shovelSchds, H
Output: s, j, tll

$tll \leftarrow \emptyset$
$s \leftarrow$ SelectShovel(S, H)
if s **not** $null$ **then**
  return $null$
else
  $j \leftarrow$ getDestination(s)
  $tempShovelSchd \leftarrow$ getSchedule(s)
  if $tempShovelSchd$ **is** $null$ **then**
    return $s, j, 0$
  else
    $tll \leftarrow$ getEndTimeLastLoading(tempShovelSchd)
    if $tll > H$ **then**
      return null
    else
      return s, j, tll
    end if
  end if
end if

After finding a shovel task, the function FindTruck (algorithm 3) is invoked with the parameters s, j, tll. The function returns the best truck t to perform the shovel task and its previous and next activities to tll (act and act'). In order to do this, the function gets the schedules of the trucks (getSchedule) and looks for all the trucks that have a free time slot to perform the shovel task. Then, for each of these trucks, the algorithm calculates the time to perform all the activities necessary to perform the shovel task. The truck that performs all the activities in the least amount of time is returned by the function. The procedure Schedule (algorithm 4) updates the shovel and truck schedules adding all the activities with their start and end times.

Algorithm 3: FindTruck.

Input: $T, s, j, tll, truckSchds$
Output: t, act, act'

$t \leftarrow \emptyset$
for $each\ t \in T$ **do**
  $tempTruckSchd \leftarrow$ getSchedule(t)
  if $tempTruckSchdisnull$ **then**
    return $t, null, null$
  else
    for $each\ act \in tempTruckSchd$ **do**
      if $act_{activity} =' unload' \wedge act_{endtime} < tll \wedge act'_{startTime} > tll$ **then**
        $totalTime \leftarrow c_s^j + c_s + c_s^j + c_j$
        if $totalTime < (act'_{starTime} - act_{endtime})$ **then**
          if $(tll + c_s + c'_s) < act'_{start}$ **then**
            return t, act, act'
          end if
        end if
      end if
    end for
  end if
end for
return $null$

Algorithm 4: Schedule.

Input: $t, s, j, tll, shovelSchds, truckSchds, act, act'$
Output:

$empTrip \leftarrow t, "emptyTrip", tll - c^s_{act_{to}}, tll, act_{to}, s$
$loadAtShovel \leftarrow t, "load", tll, c_s, s, s$
$tripToUnload \leftarrow t, "loadedTrip", tll + C_s, tll + c_s + C_s^j, s, j$
$Unload \leftarrow t, "unload", tll + c_s + C_s^j, tll + c_s + C_s^j + c_j, j, j$
$truckSchds \leftarrow truckSchds \cup empTrip \cup loadAtShovel \cup tripToUnload \cup Unload$
$shovelLoad \leftarrow s, tll, tll + c_s, t$
$shovelSchds \leftarrow shovelSchds \cup shovelLoad$

5 EVALUATION AND DISCUSSION

In order to validate the MAS, two experiments were performed. In the first one the aim was to make a comparison between the MAS and the dispatching algorithm presented in section 3 in terms of time calculation to generate the schedules. The aim of the second experiment was to compare the truck costs obtained by the MAS and the dispatching algorithm against the actual data. The experiments include several simulations run in PlaSMA, which is described in the next subsection.

5.1 Agent-based Simulation

Trialling a new method in real-world scenarios is the perfect way to determine whether the method works well or not. However, this is very hard to perform because it implies risks to the involved entities. In our case, to trialling the MAS in a real open-pit mine would affect the performance of the entire activity of the mine. In order to trial and assess the MAS, we applied a multiagent-based simulation.

PlaSMA (Warden et al., 2007) is an agent-based event-driven simulation platform created for simulation and evaluation multiagent systems. It has a special focus on simulating logistics processes. It is based on the FIPA-compliant Java Agent DEvelopment Framework (JADE) (Bellifemine et al., 2007). The transport infrastructure within the simulator is represented as a directed graph where nodes represent crossroads or logistic points such as warehouses and destinations. Edges represent different types of roads. The platform supports graphs with up 300.000 edges and 150.000 nodes, as well as the import of real-world infrastructure from OpenStreetMap.

5.2 Experimental Setup

In our experiments we use actual data from a copper open-pit mine in Chile. The modelled transport infrastructure contains 608 nodes and 1,272 edges. A heterogeneous fleet of trucks and shovels operate in shifts of 12 hours. The agent properties such as velocities and capacities are set based on the actual data. Figure 3 shows a simulation deployed in PlaSMA.

All simulations have been run on a laptop computer with an Intel Xeon 3 gigahertz CPU, 32 gigabytes of RAM and Windows 10.

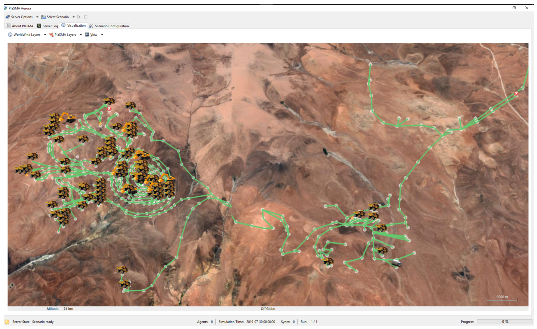

Figure 3: Open-pit mine simulated in PlaSMA.

5.3 MAS vs Disp-ALG

Several scenarios of different sizes were used in our first experiment. The scenarios had the same characteristics, i.e. the same layout of the mine and equipment items with the same properties (velocity, capacity, etc). The differences between the scenarios were the size of the fleet and the length of the shift (H). Table 4 shows the scenarios and the times it took the MAS and the Disp-ALG to generate the schedules.

Table 4: Comparison of the required time to generate the schedules between MAS and Disp-ALG.

Scenario	H	Shovels	Trucks	MAS (min)	Disp-ALG (min)
1	1	1	10	0.04	0.0025
2	3	3	25	0.43	0.0028
3	6	5	40	2.45	0.0075
4	9	7	60	4.52	0.0091

It observes that the MAS takes much more time to generate the schedules in comparison to the Disp-ALG. This is due to the overload generated by the simulation platform where the MAS is deployed and the communication effort among agents - aspects that are not present in the Disp-ALG. Notwithstanding the above, the time that the MAS takes to generate the schedules is a practical timeframe for the mining industry.

5.4 MAS vs Actual Data

In the second experiment, we simulated five shifts. These simulations were set using the actual data. The MAS and the Disp-ALG generated schedules for the equipment items to transport the same amount of transported material as displayed in the actual data. Table 5 shows the amount of transported material and the truck costs to transport those materials through

Table 5: Comparison of transported material and truck costs between Disp-ALG, MAS and actual data.

			Disp-ALG		MAS		Actual Data			
Scenario	Shovels	Trucks	Material Transported (tons)	Travel Times (hours)	Material Transported (tons)	Travel Times (hours)	Material Transported (tons)	Travel Times (hours)	Delta Material Transported (%)	Delta Travel Time (%)
1	11	99	218,891	357.39	351,659	597.44	350,117	821.89	+0.44%	-27.31%
2	11	98	203,570	414.81	405,903	713.34	404,921	849.96	+0.24%	-16.08%
3	12	93	285,677	446.94	389,404	656.58	386,973	783.49	+0.62%	-16.19%
4	12	97	284,732	439.53	384,158	698.355	381,785	831.34	+0.62%	-16.01%
5	10	102	259,492	408.85	363,531	587,07	362,020	820.66	+0.42%	-28.46%

the three alternatives: actual data, MAS and the Disp-ALG.

It observes that the MAS generates schedules in which the trucks transport more material than the Disp-ALG. This is because the matching between a shovel task and a truck is more precise in the MAS. The Disp-ALG follows a sequential creation of the schedule: it looks for the best truck for a shovel task and then continues with the following shovel task. However, it can happen that this truck is more appropriate for another shovel task. The MAS avoids this situation thanks to the concurrent negotiation mechanism in which the truck can decide between several shovel tasks and select the most appropriate. For instance, it selects the shovel task with the least cost for the truck or it selects the shovel task of the shovel with higher idle time.

Regarding truck costs, we focused on the truck costs in the schedules generated by the MAS, since the schedules generated by the Disp-ALG do not manage to transport the amount of material transported in the actual data. The schedules generated by the MAS allow the trucks to transport the same amount of material as the actual data (even a little bit more), decreasing the truck costs by 20% on average in comparison with the actual data. The main reason for these savings is that the MAS travel times of a truck are less than the travel times from the actual data, since the agents in the MAS use the shortest path for their travel, whereas the truck operators in the real world decide by themselves which path to follow. Another reason is the use of specific data to allow for more adapted calculations of the activity times of each equipment item, and in this way, the agents can create more appropriate and efficient schedules.

6 CONCLUSIONS

Truck dispatching in an open-pit mine is an important and complex process. Several centralized systems support this process. However, most of them follow an allocation model to dispatch the trucks that does not guarantee a correct synchronization between the operations of the equipment items. In order to address this situation, we developed a multiagent system (MAS) with agents that represent equipment items from the real world. The agents interact with each other using a contract net protocol with a confirmation stage to generate schedules for their represented equipment item. Schedules are a more precise way to organize and synchronize the equipment item activities than those proposed by the allocation models.

In order to evaluate the MAS, we implemented several scenarios in PlaSMA, which is a simulation platform for multiagent systems. We compared the MAS performance against a dispatching algorithm (Disp-ALG) developed by us and against actual data from an open-pit mine in Chile. The results show that the MAS achieves more efficient schedules than a centralized approach such as the Disp-ALG due to its concurrent negotiation model. The MAS schedules achieve the production levels of the actual data, while the Disp-ALG does not. However, the MAS takes much more time to generate the schedules in comparison with the Disp-ALG, although the time that the MAS takes to generate the schedules is still within a practical timeframe. Regarding the truck costs, the MAS schedules achieve the same production levels (even a little bit higher), decreasing the truck costs by 20% on average.

Our work demonstrates that an agent-based system for truck dispatching in open-pit mines is a suitable alternative. Several characteristics of the agent technology such as flexibility, robustness, and autonomy allow the agents to generate a dispatch solution that is more precise than the current approaches.

In our further research, we will develop a simulated open-pit mine. In the simulation, the simulated trucks will perform the operations pointed out in their schedules. In addition, the dynamic aspects of the material handling process will be considered, e.g., dealing with a major change in the mine such as equipment failures or changes in the mine layout. In these cases, the affected agents will have to react appropriately, interacting with each other to update their schedules.

REFERENCES

Ahangaran, D. K., Yasrebi, A. B., Wetherelt, A., and Foster, P. (2012). Real –time dispatching modelling for trucks with different capacities in open pit mines. *Archives of Mining Sciences*, 57(1):39–52.

Alarie, S. and Gamache, M. (2002). 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.

Alexandre, R. F., Campelo, F., Fonseca, C. M., and de Vasconcelos, J. A. (2015). A Comparative Study of Algorithms for Solving the Multiobjective Open-Pit Mining Operational Planning Problems. In *International Conference on Evolutionary Multi-Criterion Optimization*, volume 9019, pages 433–447.

Bastos, G. S., Souza, L. E., Ramos, F. T., and Ribeiro, C. H. C. (2011). A single-dependent agent approach for stochastic time-dependent truck dispatching in open-pit mining. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1057–1062.

Bellifemine, F., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*.

Chang, Y., Ren, H., and Wang, S. (2015). Modelling and optimizing an open-pit truck scheduling problem. *Discrete Dynamics in Nature and Society*, 2015.

Chen, B. and Cheng, H. H. (2010). A review of the applications of agent technology in traffic and transportation systems.

Costa, F. P. D., Souza, M. J. F., and Pinto, L. R. (2005). Um modelo de programação matemática para alocação estática de caminhões visando ao atendimento de metas de produção e qualidade. *Rem: Revista Escola de Minas*, 58(1):77–81.

Ercelebi, S. G. and Bascetin, A. (2009). Optimization of shovel-truck system for surface mining. *Journal of the Southern African Institute of Mining and Metallurgy*, 109(7):433–439.

Fischer, K., Müller, J. P., and Pischel, M. (1996). Cooperative transportation scheduling: an application domain for dai. *Applied Artificial Intelligence*, 10(1):1–34.

Gath, M. (2015). *Optimizing Transport Logistics Processes with Multiagent-based Planning and Control*. PhD thesis.

Hashemi, A. S. and Sattarvand, J. (2015). Proceedings of the 12th International Symposium Continuous Surface Mining - Aachen 2014. pages 213–214.

Jaoua, A., Riopel, D., and Gamache, M. (2012). A simulation framework for real-time fleet management in internal transport systems. *Simulation Modelling Practice and Theory*, 21(1):78–90.

Krzyzanowska, J. (2007). The impact of mixed fleet hauling on mining operations at Venetia mine. *Journal of the Southern African Institute of Mining and Metallurgy*, 107(4):215–224.

Mendes, J. B., D'Angelo, M. F. S. V., Maia, N. A., and Veloso, R. R. (2016). A Hybrid Multiobjective Evolutionary Algorithm for Truck Dispatching in Open-Pit-Mining. *IEEE Latin America Transactions*, 14(3):1329–1334.

Mes, M., van der Heijden, M., and van Harten, A. (2007). Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research*, 181(1):59–75.

Newman, A. M., Rubio, E., Caro, R., Weintraub, A., and Eurek, K. (2010). A review of operations research in mine planning. *Interfaces*, 40(3):222–245.

Patterson, S. R., Kozan, E., and Hyland, P. (2017). Energy efficient scheduling of open-pit coal mine trucks. *European Journal of Operational Research*, 262(2):759–770.

Schillo, M., Kray, C., and Fischer, K. (2002). The eager bidder problem: a fundamental problem of DAI and selected solutions. *Proceedings of the 1st international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, (January):599–606.

Smith, R. G. (1980). The Contract Net Protocol:High-level communication and control in a distributed problem solver. *IEEE Tranactions on Computers*, C-29(12):1104–1113.

Warden, T., Porzel, R., Gehrke, J. D., Herzog, O., Langer, H., and Malaka, R. (2007). Towards Ontology-Based Multiagent Simulations : Plasma Approach.