A Dynamic Scheduling Multiagent System for Truck Dispatching in Open-Pit Mines

Gabriel Icarte Ahumada[1,2](✉), Jean Diaz Pinto[3], and Otthein Herzog[4,5,6]

[1] International Graduate School for Dynamics in Logistics (IGS), University of Bremen, Bremen, Germany
gicartea@uni-bremen.de

[2] Faculty of Engineering and Architecture, Arturo Prat University - UNAP, Iquique, Chile
gicarte@unap.cl

[3] Science Faculty, Arturo Prat University - UNAP, Iquique, Chile
jeadiaz@unap.cl

[4] TZI - Center for Computing Technologies, University of Bremen, Bremen, Germany
herzog@tzi.de

[5] Jacobs University, Bremen, Germany

[6] Tongji University Shanghai, Shanghai, People's Republic of China

Abstract. Material handling is an important process in the mining industry because of its high operational cost. In this process, shovels extract and load materials that must be transported by trucks to different destinations at the mine. When a truck ends an unloading operation, it requires a new loading destination. If a centralized system provides destinations by following dispatching criteria, then one of the main disadvantages of this kind of systems is not being able to provide a precise dispatching solution without knowledge about potentially changed external conditions and the dependency on a central node. In this paper, we describe a distributed approach based on Multiagent Systems (MAS) to alleviate these disadvantages. In this approach, the real-world equipment items such as shovels and trucks are represented by intelligent agents. The agents interact with each other to generate schedules for the machines that they represent. For this interaction, a Contract Net Protocol with a confirmation stage was implemented. In addition, when a machine failure occurs, the agents are able to update their schedules. In order to evaluate the MAS, an agent-based simulation with data from a Chilean open-pit mine was used. The results show that the MAS is able to generate the schedules in a practical computation timeframe. The schedules generated by the MAS decrease the truck cost by 17% on average. Moreover, when a machine failure occurs, the agents are able to repair their schedules in a short period of time.

Keyword: Truck dispatching · Open-pit mine · Multiagent systems · Scheduling · Rescheduling

A. P. Rocha et al. (Eds.): ICAART 2020, LNAI 12613, pp. 132–148, 2021.
https://doi.org/10.1007/978-3-030-71158-0_6

1 Introduction

Material handling is an important process in open-pit mines since its cost can represent up to 50% of the entire operational cost in these kinds of mines [1]. In this process shovels and trucks work together to extract and to transport all the material required in the operational plan at minimum cost. If the material extracted is ore, it must be transported and unloaded into a crusher or onto a stockpile. If the extracted material is waste, it must be transported to a waste dump. Figure 1 depicts the operations that a truck must perform to transport materials from a shovel to a crusher, stockpile, or waste dump. This cycle is performed and repeated by each truck until the shift ends.

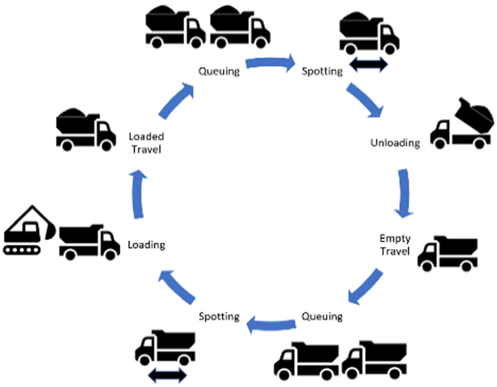

Fig. 1. The truck cycle. Adapted from [2, p. 2]

The material handling process is performed in a dynamic environment that affects the performance and availability of the involved equipment. For example, changes in weather conditions or to the state of the routes as well as equipment failures are some of the reasons for delays in material handling [3].

During the material handling process, each time a truck ends an unloading operation it requires a new loading destination, i.e., it requires a new assignment to a shovel where the truck must go for a new loading. This is called truck dispatching. Determining the new destination for the truck is not easy because of the stochasticity of the material handling process and the dynamics of the environment where the equipment items perform their operations.

Different centralized systems have been implemented to support truck dispatching in open-pit mines [4]. Most of these systems use a multistage approach [1], which computes a guideline in the first stage. A later stage uses this guideline to dispatch the trucks in real-time each time that a truck ends an unloading operation.

One example of the application of the multistage approach is a system with two stages. The first stage determines, before the shift starts, the number of travels to transport the required materials by the production plan from shovels to unloading points. Then, when the trucks are working and when a truck requires a new destination, the later stage determines the destination based on a dispatching criterion (e.g., the number of trips performed) and the current information on the process.

The strengths of these centralized systems are their maturity and their well-known implementation. However, the weaknesses that can be observed are using estimated information [5–8], and not being able to provide a precise solution [9]. In addition, using dispatching criteria in this way is myopic, and it is hard to predict the performance of the entire system since decisions are made locally in real-time [10]. Due to these weaknesses, the material handling process is not performed efficiently, since queues of trucks can form in front of shovels and crushers, and there can also be shovel idle times. Therefore, the question remains as how to improve the efficiency of the material handling process.

Icarte et al. [2] developed an alternative solution that allows the equipment items to operate more efficiently. The solution is based on a Multiagent system in which intelligent agents that represent real-world equipment interact with each other to generate schedules.

In this paper, we extend the work presented in [2] by performing and evaluating extended experiments for the generation of schedules, and by the evaluation of the MAS when a major unforeseen event occurs at the mine. In order to demonstrate the validity of the MAS, the experiments were performed based on actual data from a Chilean open-pit mine.

The remainder of this paper is structured as follows: Sect. 2 presents related work, Sect. 3 presents the formulation of the problem, and Sect. 4 presents the distributed solution based on MAS. Section 5 presents the evaluation and discussion of the proposed MAS. Finally, conclusions and outlook are presented in Sect. 6.

2 Related Work

In the past years, many articles deal with the truck dispatching problem in open-pit mines. These articles show different methods that try to achieve two goals: improve productivity and reduce operating costs [1]. These methods are based on Operations Research, simulation modelling or heuristic procedures and follow a centralized approach. For instance, Bakhtavar and Mahmoudi [11] developed a two-phase scenario-based robust optimization (SBRO) model by considering the maximization of production, control of ore grade sent to the crusher, minimization of waiting times for trucks and shovels, and trucks with different capacities. Koryagin and Voronov [12] presented a heuristic algorithm to dispatch trucks after unloading. The algorithm uses priority parameters for choosing the shovel to which the truck will be allocated, and the corresponding route for material transportation is considered. In addition, the algorithm applies a minimizing truck waiting time criterion. Chaowasakoo et al. [13] illustrated the differences between the dispatching strategies by conducting a stochastic simulation study based on the data gathered from an actual big-size mine.

Most of the articles on the truck dispatching problem in open-pit mines propose a method that determines the next destination of a truck after the completion of an unloading activity. To do this, the methods use a previously calculated guideline, the current status of the mine, and some dispatching criteria. Although these methods are easy to implement, they 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 [9]. In scheduling theory, this kind of method is called "Completely reactive scheduling" [10, 14].

In "Completely reactive scheduling" no schedule is generated. The decision on what is the next task to be performed is made locally in real-time [10] considering the available information at the moment and some dispatching criteria. This way of dynamic scheduling is quick, usually intuitive, and easy to implement. However, a schedule of the tasks to perform has the potential to significantly improve the performance of a system [10].

Few articles have proposed to generate schedules for the truck dispatching problem in open-pit mines. Chang et al. [3] and Patterson et al. [7] proposed algorithms that generate an initial schedule which is improved by using a metaheuristic method. Their results show that these algorithms generate schedules for different size instances with good results and performance in practical frame times. Icarte et al. [2] developed a distributed approach based on multiagent systems to generate schedules for the equipment item involved in the material handling process. Their results show that a MAS approach provides schedules in practical frame times for the mining industry. However, none of these articles mention how the schedules could be rescheduled when a major unforeseen event occurs in a mine.

There are several articles that use multiagent systems for scheduling and rescheduling problems. Most of them are applied in manufacturing [15–18] and for vehicles [19–24]. All these authors mention the good performance of MAS to generate schedules and, some of them, a good reaction when an event occurs [16, 19–21, 23, 25]. These articles demonstrate the applicability of MAS to scheduling and rescheduling problems.

3 Problem Representation

Truck dispatching in open-pit mines as a scheduling problem is represented by the following notations, constraints, and objective functions:

3.1 Notations

- $S = \{s_1, \ldots, s_n\}$. It is the set of shovels.
- $T = \{t_1, \ldots, t_m\}$. It is the set of trucks.
- $J = \{j_1, \ldots, j_p\}$. It is the set of the material extracted by a shovel.
- C_t^s. It is the loading time that the shovel s takes for loading the truck t.
- C_j^t. It is the unloading time that the truck t takes for unloading at destination j.
- C_loaded_t. It is the cost matrix $(C_{s,j})$ of truck t that represent the cost of traveling from shovel s to destination j.
- C_empty_t. It is the cost matrix $(C_{j,s})$ of truck t that represent the cost of traveling from destination j to shovel s.

- $TargetPlan = \{(s, j, x) | s \in S,\ j \in J,\ x \in \mathbb{N}\}$. It is the material that must be extracted by shovel s and transported to destination j.
- $TruckOperation = \{$'emptyTrip','*loading*','*loadedTrip*','*unloading*'$\}$. The operation that a truck can perform.
- $TruckSchedule = \{(t, operation, startTime,\ endTime, from, to) | t \in T, operation \in TruckOperation, startTime \in \mathbb{N}, \mathrm{endTime} \in \mathbb{N}, \mathrm{from} \in (\mathrm{S} \cup \mathrm{J}), \mathrm{to} \in (\mathrm{S} \cup \mathrm{J})\}$. It is the schedule of the trucks.
- $ShovelSchedule = \{(s, startTime, endTime, quantityLoaded) | s \in S, startTime \in \mathbb{N}$, $\mathrm{endTime} \in \mathbb{N}\}$. It is the schedule of the shovels.

The following constraints set the sequence and times of the operations scheduled and avoiding overlapping them.

$$startTime + C\_empty_t(j, s) = endTime \forall x_{(t, emptyTrip, starTime, endTime, s, j)} \in TruckSchedule \tag{1}$$

$$startTime + C\_loaded_t(s, j) = endTime \forall x_{(t, loadedTrip, starTime, endTime, j, s)} \in TruckSchedule \tag{2}$$

$$startTime + C_t^s = endTime \forall x_{(t, loading, starTime, endTime, s, s)} \in TruckSchedule \tag{3}$$

$$startTime + C_j^t = endTime \forall x_{(t, \mathrm{un}loading, starTime, endTime, j, j)} \in TruckSchedule \tag{4}$$

$$x_{(t, emptyTrip, startTime, endTime, s, j)} \leq y_{(t, loading, startTime', endTime', s, s)} \, iff \; endTime \leq startTime' \forall x, y \in TruckSchedule \tag{5}$$

$$x_{(t, loading, startTime, endTime, s, s)} \leq y_{(t, loadedTrip, startTime', endTime', s, j)} \, iff \; endTime \leq startTime' \forall x, y \in TruckSchedule \tag{6}$$

$$x_{(t, loadedTrip, startTime, endTime, s, j)} \leq y_{(t, unloading, startTime', endTime', j, j)} \, iff \; endTime \leq startTime' \forall x, y \in TruckSchedule \tag{7}$$

$$x_{(t, unloading, startTime, endTime, j, j)} \leq y_{(t, emptyTrip, startTime', endTime', j, s)} \, iff \; endTime \leq startTime' \forall x, y \in TruckSchedule \tag{8}$$

$$startTime + C_t^s = endTime \forall x_{(s, starTime, endTime)} \in ShovelSchedule \tag{9}$$

$$x_{(s, startTime, endTime)} \leq y_{(s, startTime', endTime')} \, iff \; endTime \leq startTime' \forall x, y \in ShovelSchedule \tag{10}$$

3.2 Objective Functions

The truck dispatching problem pursues two objectives: to achieve the production targets and the minimization of the truck travel costs. These objectives can be formulated as follows:

$$max \sum_{x \in ShovelSchedule} (quantityLoaded) \tag{11}$$

$$min \sum_{x \in TruckSchedule} (endTime - starTime) \tag{12}$$

4 Dynamic Scheduling Multiagent System for Truck Dispatching in Open-Pit Mines

The objective of the developed MAS is to achieve the goals of the production plan at minimum cost. In order to do this, the agents must interact with each other to generate schedules for each equipment item involved in the material handling process. Moreover, the agents must update the generated schedules when a major unforeseen event occurs at the mine. Table 1 shows the implemented agents, their objectives, and properties.

Our developed multiagent system for truck dispatching in open-pit mines has the following advantages over current centralized systems:

- Efficient Dispatching Solution: The schedules generated by the MAS are more efficient than a “Completely reactive scheduling”, which is applied by the current systems.
- Use of Specific Data from Equipment Items: This allows the MAS to generate more precise schedules.

Table 1. Agent description. Adapted from [2, p. 4].

Agent	Real-world representation	Objective	Properties
truckAgent	Trucks	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 loaders	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	Crusher, stockpiles, waste dumps	Create a schedule of the activities of the equipment that it represents	Number of trucks unloading simultaneously

- Robustness: As there is not a central node, if any node is out of order, the system continues working without it.
- Flexibility: An agent can change its behavior 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.
- Reactiveness: The agents are able to update the schedules when a major unforeseen event occurs at the mine.

The following subsections describe the interactions among the agents and their decision-making processes.

4.1 Interactions for Scheduling

In order to generate the schedules, the agents negotiate among them by using the adapted Contract Net Protocol presented in [2]. In this protocol, a *shovelAgent* plays the role of the initiator and *truckAgents* play the role of participants. The *shovelAgents* in the system start this negotiation process in parallel, therefore the protocol must manage concurrent negotiations.

To manage concurrent negotiations, the original CNP is adapted by including a confirmation stage. The state transitions in the negotiation process between a *shovelAgent* and *truckAgents* are depicted in Fig. 2. The process starts when a *shovelAgent* sends a *call-for-proposal* message (CFP) to *truckAgents*. After sending the CFP, the process passes to the first state "Evaluating CFP". In this state, *the truckAgents* evaluate the received CFP and decide on sending a *proposal* or a *refuse* message as the answer. Meanwhile the *shovelAgent* is waiting for the messages from the *truckAgents*.

When a *shovelAgent* receives a message from a *truckAgent*, the negotiation process passes to the state "Receiving proposals". In this state, the *shovelAgent* receives and stores the proposals sent by the *truckAgents* and keeps a count of the received messages from the *truckAgents*. The negotiation keeps in this state until a deadline expires or when the *shovelAgent* has received all the messages (*propose* or *refuse*) from the *truckAgents*. In the "Receiving proposals" state two situations can happen: having received proposals or not. If it has not received any proposals, the negotiation process finishes without a contract. If it has received proposals, the negotiation process passes to the state "Selecting best proposal".

In the state "Selecting best proposal", the *shovelAgent* selects the best proposal and sends a message with the performative *requestConfirmation* to the truckAgent that sent the best proposal, and the negotiation process passes to the state "Waiting for answer".

In the state "Waiting for answer", three situations can happen: receiving a message with the performative *acceptConfirmation*, receiving a message with the performative *refuseConfirmation*, or a deadline expires. In the first case, the negotiation process passes to the state "Sending results". In the "Sending results" state the *shovelAgent* sends a *reject-proposal* message to those *truckAgents* that were not awarded, and the negotiation process ends with a contract. In the second and third cases, the negotiation process passes to the "Removing proposal" state. In this state, the *shovelAgent* removes the proposal from its storage. If there are more proposals, the negotiation process

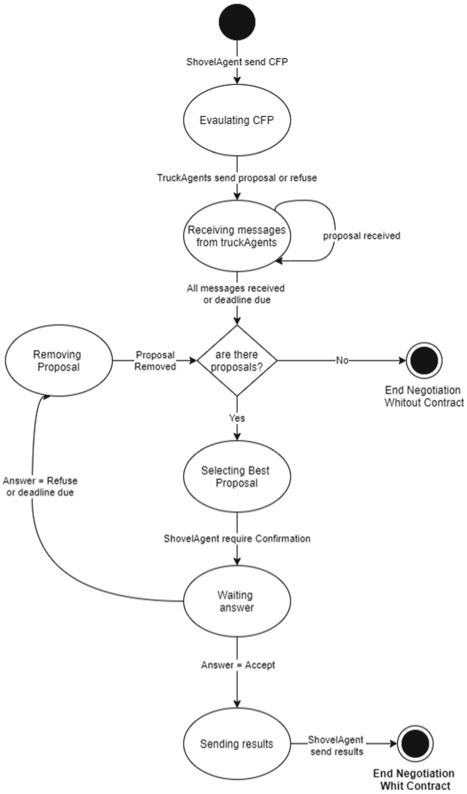
ShovelAgent send CFP
Evaulating CFP
TruckAgents send proposal or refuse
Receiving messages from truckAgents
proposal received
All messages received or deadline due
Removing Proposal
Proposal Removed
are there proposals?
No
End Negotiation Whitout Contract
Yes
Answer = Refuse or deadline due
Selecting Best Proposal
ShovelAgent require Confirmation
Waiting answer
Answer = Accept
Sending results
ShovelAgent send results
End Negotiation Whit Contract

Fig. 2. State machine diagram of the negotiation process by applying the adapted Contract Net Protocol.

Table 2. Example of schedule created for a truck.

Assignment	Destination	Start time of the Trip	Arrival time	Start time of the loading/unloading	End time of the assignment
0	Shovel.04	05:57:01	06:10:23	06:10:36	06:15:12
1	WasteDump.03	06:15:12	06:32:33	06:38:23	06:40:23
2	Shove 1.04	06:45:25	06:58:47	07:00:10	07:05:35
3	WasteDump.02	07:05:35	07:22:24	07:26:38	07:27:12
4	Shovel.02	07:37:44	07:41:25	07:43:18	07:48:32

passes to the state "Selecting best proposal" and repeats the cycle. If there are no more proposals, the negotiation process ends without a contract. Figure 3 depicts the interaction between the agents using the adapted Contract Net Protocol and Table 2 shows a schedule example for a truck generated by the agents by applying this protocol.

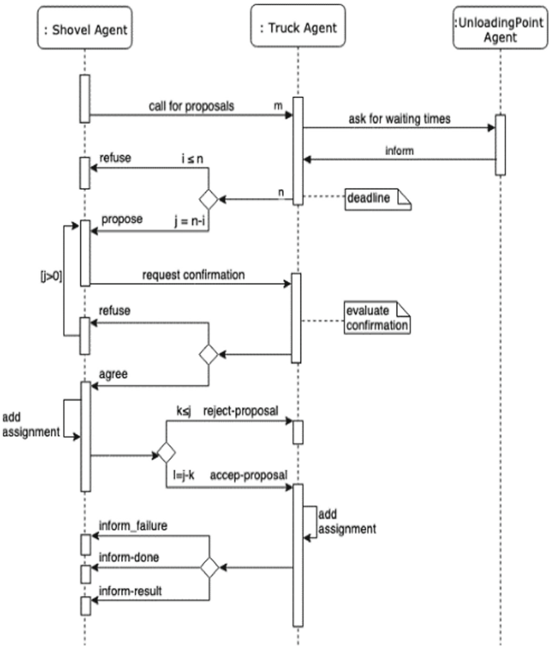

Fig. 3. The interaction between the agents using the CNP with the confirmation stage. Adapted from [2, p. 4].

4.2 Interactions for Rescheduling

In the scheduling literature, two main rescheduling strategies are mentioned: schedule repair, and complete rescheduling [10]. scheduling repair refers to some local adjustment of the current schedule. complete rescheduling regenerates a new schedule from scratch.

In the context of the MAS for the truck dispatching in open-pit mines, the agents can use both strategies, depending on the type event that triggers the rescheduling. In both strategies, the agents apply the adapted CNP described previously. For the schedule repair strategy, some agents may apply changes to their schedules. For instance, in the case of a truck failure, the *truckAgent* that represents the broken truck informs *shovelAgents* and *unloadingAgents* that the truck will not be able to perform the assignments. The *shovelAgent* must cancel the affected assignment and its production will decrease. Here, the *shovelAgent* starts new a negotiation process cycle to try to reach its production goals again. In the case of complete rescheduling, all agents must generate a new schedule from the point in time where the failure occurred. All agents cancel their assignments and the agents interact with each other by applying the adapted CNP to generate new schedules.

4.3 Decision Making

In the MAS for truck dispatch in open-pit mines, *ShovelAgents* and *TruckAgents* make decisions at different moments in the negotiation process. The following subsections describe the decision-making process for these types of agents.

ShovelAgent. *ShovelAgents* only must make one decision during a negotiation process: what is the best proposal received. to make this decision, the *shovelagents* use the utility function proposed in [2], which promotes those proposals that decrease the shovel's waiting time and minimize the cost to perform the truck operations. formally:

$$offer = number \in \{0\} \cup \mathbb{N}.\text{ It represents the time offered by a}shovelAgent \tag{13}$$

to start a new loading.

$$P = set\,of\;received\,proposals p_1 \ldots .p_n \tag{14}$$

The decision is a multicriteria problem as

$$Decision = arg_{min}(p_i \in P)\{U(p_1), \ldots, U(p_n)) \tag{15}$$

$$U = U(arrivalTime)\prime + \text{U}(\text{cost})\prime \tag{16}$$

$$\delta = (arrivalTime - offer) \in \mathbb{Z} \tag{17}$$

$$U(arrivalTime) = \begin{cases} \delta,\; if\,\delta < 0 \;\; (the\,truck\,arrives\,earlier) \\ 0,\; if\,\delta = 0 \;\; (the\,truck\,arrives\,just\,on\,time) \\ 2 * \delta,\; if\,\delta > 0 \;\; (the\,truck\,arrives\,later) \end{cases} \tag{18}$$

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

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{19}$$

TruckAgent. A *truckagent* must make two decisions. the first one is deciding whether or not to send a proposal to a *shovelagent* that sent a *call-for-proposal* message. the second one is to decide on whether or not to confirm a previously sent proposal. Figure 4 depicts both decision-making processes.

To decide whether or not to send a proposal, the *truckAgent* uses the *call-for-proposal* message's information and its current schedule. The *shovelAgents* send *a call-for-proposal* offering the next available time to load a truck. After receiving a CFP, the *truckAgent* checks its schedule and determines if there is a free time slot for the offered time. If not, it sends a *reject* message. If yes, it calculates the total time to perform all the operations and determines if it fits into its schedule. If it does not fit into the schedule, it sends a *reject* message. If it fits into the schedule, it sends a proposal. This decision-making process is shown in Fig. 4a.

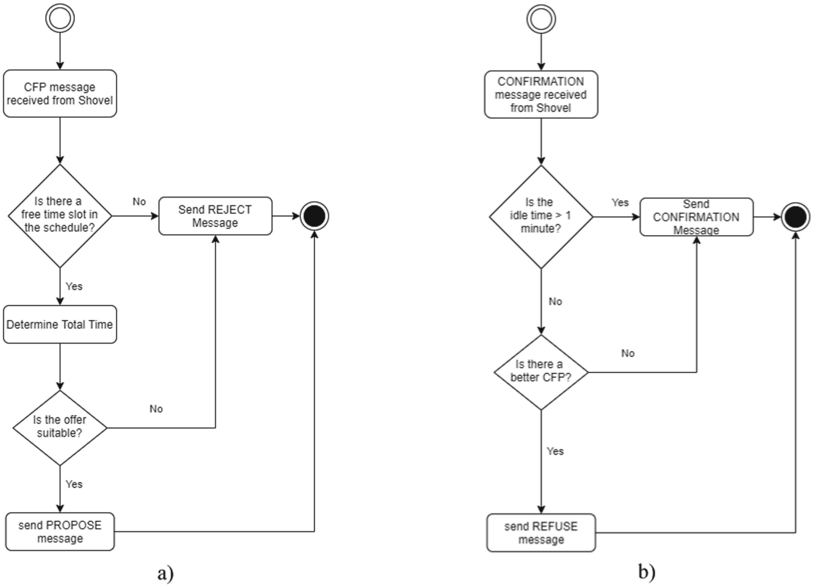
CFP message received from Shovel
Is there a free time slot in the schedule?
No
Send REJECT Message
Yes
Determine Total Time
Is the offer suitable?
No
Yes
send PROPOSE message
a)
CONFIRMATION message received from Shovel
Is the idle time > 1 minute?
Yes
Send CONFIRMATION Message
No
Is there a better CFP?
No
Yes
send REFUSE message
b)

Fig. 4. Decision-making processes for a *TruckAgent*.

To decide whether or not to confirm a proposal sent previously, the *truckAgent* must consider two aspects: the shovel idle time (informed by the *requestConfirmation* message sent by the *shovelAgent*), and the negotiations which the *truckAgent* is taking part in. If the shovel idle time is higher or equal than one minute, the truckAgent sends an *acceptConfirmation* message. If the shovel idle time is less than one minute, the *truckAgent* considers the negotiations which it is taking part in. If there is a negotiation with more potential benefit for the truck (i.e., with a lower cost to perform the operations), the *truckAgent* sends a *refuseConfirmation* message. Otherwise, it sends an *acceptationConfirmation* message. This decision process is depicted in Fig. 4b.

5 Experiments and Discussion

In order to validate the MAS, two types of experiments were performed. On the one hand, some experiments evaluated the performance and quality of the schedules generated by the MAS. On the other hand, additional experiments evaluated the ability of the MAS to reschedule when a machine failure occurs at the mine. The following subsections describe the experiments and their results.

5.1 Experimental Setup

Our experiments were based on actual data from an open-pit copper mine in Chile. The material handling process at the mine is performed using a heterogeneous fleet of trucks and shovels in twelve-hour shifts. The truck dispatching is supported by DISPATCH

(TM), which is a well-known dispatch system in the mining industry. This system follows a centralized approach and Completely Reactive Scheduling. The actual data, such as velocities and capacities, is used to set the properties of the agents. Table 3 shows the property sets of the agents.

The experiments were performed by simulations run in PlaSMA [29], which is an agent-based event-driven simulation platform created for the simulation and evaluation of multiagent systems. It focuses on simulating logistics processes and is based on the FIPA-compliant Java Agent DEvelopment Framework (JADE) [30]. All simulations have been run on a laptop computer with an Intel Xeon 3 GHz CPU, 32 GB of RAM, and Windows 10.

5.2 Experiment on Schedule Generation

The aim of this experiment was to determine the required time by the MAS to generate the schedules. In order to do this, we simulated several scenarios of different sizes. 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 (column H). Table 4 shows the scenarios and the time it took the MAS to generate the schedules.

Table 3. Property values for the simulations.

Equipment	Property	Unit	Min value	Max value
Trucks	Velocity loaded	[km/hr]	20	25
	Velocity empty	[km/hr]	40	55
	Capacity	[tons]	300	370
	Spotting time	[sec]	20	80
	Current load	[tons]	0	370
Shovel	Capacity	[tons]	35	80
	Load time	[sec]	8	30
	Dig time	[sec]	8	20
	Destination	Location at mine (crusher, stockpile, or waste dump)		
Crusher	Equipment discharging	[number of trucks]	1	1
Stockpile	Equipment discharging	[number of trucks]	1	20
Waste Dump	Equipment discharging	[number of trucks]	1	20

Table 4. Required time to generate the schedules by the MAS.

Scenario	H	Shovels	Trucks	Minutes
1	1	1	10	0.04
2	3	3	25	0.22
3	6	5	40	0.90
4	9	7	60	3.35
5	12	10	90	13.28

The results show that the required time for the MAS to generate the schedules increases with a bigger problem. However, the required time for the last scenario, which is a common scenario of a big-size open-pit mine, is a practical timeframe to generate schedules in the context of the mining industry.

5.3 Experiment on the Quality of the Generated Schedules

In this experiment, the aim was to analyze the quality of the generated schedules by the MAS, by comparing the production and cost of simulated shifts against real shifts. We simulated five shifts based on the actual data. Table 5 shows the amount of transported material and the truck cost to transport those materials.

Table 5. Comparison of transported material and truck cost between MAS and actual data.

Shift (hours)	Shovels	Trucks	Actual material transported (Tons)	Actual travel time (Hours)	Simulated material transported (Tons)	Simulated travel time (Hours)	Delta material transported	Delta travel time
12	11	99	367.623	863,00	368.215	702,24	+0,16%	−18,63%
12	11	96	360.505	820,25	361.254	688,42	+0,21%	−16,07%
12	12	99	392.773	824,47	394.259	691,93	+0,38%	−16,08%
12	12	101	405.459	804,88	406.254	655,26	+0,20%	−18,59%
12	12	95	379.234	767,82	381.245	648,69	+0,53%	−15,52%

The results show that if the trucks had been dispatched according to the schedules, the material handling process had been more efficient. This is because 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 travel costs by 17% on average in comparison to the actual data.

The main reason for these savings is that the agents decrease their travel times by applying a shortest path algorithm, 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 calculations better adapted to the activity times of each equipment item, and in this way, the agents can create more appropriate and efficient schedules.

Table 6. Performance of rescheduling strategies.

	Production (tons)	Cost (hours)	Required time (sec)
Initial schedule	512,049	844,048	796,800
Simple schedule update	467,949	781,354	0.002
Schedule repair	494,633	818,339	15,215
Complete rescheduling	496,927	821,121	785,500

5.4 Experiment on Rescheduling

The objective of this experiment was to evaluate the reaction of the MAS when a machine failure occurs in the mine. In this context, we compared a simple schedule update (cancelation of the affected assignments), a schedule repair, and a complete reschedule when truck failures occur. We simulated truck failures because it is a common situation in open-pit mines.

In the simulated scenario, the fleet is compounded by 90 trucks and 10 shovels and operates in a twelve-hours shift. Two trucks have a failure at the beginning of the shift and are not available anymore. Table 6 shows the performance of the new schedules (in terms of production level and the cost) and the required time to get them by applying a simple schedule updating, schedule repair, and complete rescheduling.

These results show that the simple schedule update achieves the lowest production level in a very short time. This was predictable because the agents update only their own schedules canceling the affected assignments. It was calculated in order to contrast it with the increase in the production level by applying schedule repair and Complete rescheduling.

Schedule repair increases the production level by 5,7% and requires more time to regenerate the schedules than the simple schedule update. This happens because the affected *shovelAgents*, start the negotiation processes with the cancelation of some loading assignments to achieve their production goals.

Complete rescheduling achieves the highest production level, but the required time to generate all schedules is much higher in comparison to schedule repair. This happens because all *shovelAgents* cancel all their assignments and initiate the negotiation processes to regenerate the schedules from scratch. As it was described before, in schedule repair only a subset of *shovelAgents* (the affected ones) start the negotiation processes. In addition, in complete rescheduling, since the *truckAgents* do not have schedules, they take more time to evaluate and decide on whether to send a proposal. In schedule repair, as the *truckAgents* have schedules, they are more restricted to send proposals, and therefore it takes them less time to evaluate and decide on whether to send a proposal. Basically, a negotiation process takes more time in complete rescheduling than in schedule repair.

At first sight, it looks like schedule repair is the most suitable strategy to apply by the MAS when truck failures occur since in a short period of time the schedules are repaired. However, total regeneration is also a suitable alternative to be applied by the MAS in spite the longer time it requires. This is because the adapted CNP is implemented by anytime algorithms, i.e., complete rescheduling returns a valid schedule even if it is interrupted before it ends.

6 Conclusions

A major process in open-pit mines is material handling. In this process, dispatching a truck when it completes an unloading operation becomes a complex decision because of the stochasticity of the process and the dynamics of the environment where the equipment items operate. In general, a centralized system supports this process. However, most of them follow a reactive scheduling strategy to dispatch the trucks. This strategy 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 an extended Contract Net Protocol with a confirmation stage to generate initial schedules for their represented equipment items. Schedules are a more precise way to organize and synchronize the equipment item activities than those proposed by a reactive scheduling strategy. In addition, the agents are able to repair their schedules when trucks have failures.

In order to evaluate the MAS, we made experiments based on simulations to determine the efficiency of the schedules generated by the MAS, and the reaction of the agents to repair the schedules when truck failures occur. Regarding the efficiency of the schedules, the simulations show that the trucks achieved the production level of the actual data by following the schedules with decreasing truck cost by 17% on average. In addition, the MAS generated the schedules in a practical frame time for the mining industry.

Regarding the reactiveness of the MAS when trucks have failures, the experiments demonstrate that both, scheduling repair and complete repair strategies, are suitable even if the second one takes more time for scheduling than the first one.

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

In our further research, we will compare our approach also against a metaheuristic algorithm for both scheduling and rescheduling. In addition, we will consider other events such as shovel and truck delays.

References

1. Alarie, S., Gamache, M.: Overview of solution strategies used in truck dispatching systems for open pit mines. Int. J. Surf. Min. Reclam. Environ. **16**(1), 59–76 (2002)
2. Icarte, G., Rivero, E., Herzog, O.: An agent-based system for truck dispatching in open-pit mines. In: ICAART (2020)

3. Adams, K.K., Bansah, K.K.: Review of operational delays in shovel-truck system of surface mining operations. In: 4th UMaT Biennial International Mining and Mineral Conference, pp. 60–65 (2016)
4. Icarte, G., Herzog, O.: A multi-agent system for truck dispatching in an open-pit mine. In: Second International Conference Mines of the Future (2019)
5. Chang, Y., Ren, H., Wang, S.: Modelling and optimizing an open-pit truck scheduling problem. Discret. Dyn. Nat. Soc. **2015** (2015)
6. Da Costa, F.P., Souza, M.J.F., Pinto, L.R.: Um modelo de programação matemática para alocação estática de caminhões visando ao atendimento de metas de produção e qualidade. Rem Rev. Esc. Minas **58**(1), 77–81 (2005)
7. Krzyzanowska, J.: The impact of mixed fleet hauling on mining operations at Venetia mine. J. South. Afr. Inst. Min. Metall. **107**(4), 215–224 (2007)
8. Newman, A.M., Rubio, E., Caro, R., Weintraub, A., Eurek, K.: A review of operations research in mine planning. Interfaces (Providence) **40**(3), 222–245 (2010)
9. Patterson, S.R., Kozan, E., Hyland, P.: Energy efficient scheduling of open-pit coal mine trucks. Eur. J. Oper. Res. **262**(2), 759–770 (2017)
10. Ouelhadj, D., Petrovic, S.: A survey of dynamic scheduling in manufacturing systems. J. Sched. **12**(4), 417–431 (2009)
11. Bakhtavar, E., Mahmoudi, H.: Development of a scenario-based robust model for the optimal truck-shovel allocation in open-pit mining. Comput. Oper. Res. **115**, 104539 (2020)
12. Koryagin, M., Voronov, A.: Improving the organization of the shovel-truck systems in open-pit coal mines. Transp. Probl. **12**(2), 113–122 (2017)
13. Chaowasakoo, P., Seppälä, H., Koivo, H., Zhou, Q.: Digitalization of mine operations: scenarios to benefit in real-time truck dispatching. Int. J. Min. Sci. Technol. **27**(2), 229–236 (2017)
14. Vieira, G.E., Herrmann, J.W., Lin, E.: Rescheduling manufacturing systems: a framework of strategies, policies, and methods. J. Sched. **6**(1), 39–62 (2003)
15. Lopes Silva, M.A., de Souza, S.R., Freitas Souza, M.J., Bazzan, A.L.C.: A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. Expert Syst. Appl. **131**, 148–171 (2019)
16. Gehlhoff, F., Fay, A.: On agent-based decentralized and integrated scheduling for small-scale manufacturing. At-Automatisierungstechnik **68**(1), 15–31 (2020)
17. Wang, J., Zhang, Y., Liu, Y., Wu, N., Member, S.: Multiagent and bargaining-game-based real-time scheduling for internet of things-enabled flexible job shop. IEEE Internet Things J. **6**, 2518–2531 (2018)
18. Martin, S., Ouelhadj, D., Beullens, P., Ozcan, E., Juan, A.A., Burke, E.K.: A multi-agent based cooperative approach to scheduling and routing. Eur. J. Oper. Res. **254**(1), 169–178 (2016)
19. Chargui, K., El fallahi, A., Reghioui, M., Zouadi, T.: A reactive multi-agent approach for online (re)scheduling of resources in port container terminals. IFAC-PapersOnLine **52**(13), 124–129 (2019)
20. Whitbrook, A., Meng, Q., Chung, P.W.H.: Reliable, distributed scheduling and rescheduling for time-critical, multiagent systems. IEEE Trans. Autom. Sci. Eng. **15**(2), 732–747 (2018)
21. Seitaridis, A., Rigas, E.S., Bassiliades, N., Ramchurn, S.D.: An agent-based negotiation scheme for the distribution of electric vehicles across a set of charging stations. Simul. Model. Pract. Theory **100**, 102040 (2020)
22. Madhyastha, M., Reddy, S.C., Rao, S.: Online scheduling of a fleet of autonomous vehicles using agent-based procurement auctions. In: Proceeding - 2017 IEEE International Conference on Service Operation Logistics and Informatics, SOLI 2017, vol. 2017, pp. 114–120 (2017)

23. Skobelev, P., Budaev, D., Brankovsky, A., Voschuk, G.: Multi-agent tasks scheduling for coordinated actions of unmanned aerial vehicles acting in group. Int. J. Des. Nat. Ecodyn. **13** (1), 39–45 (2018)
24. Granichin, O., Skobelev, P., Lada, A., Mayorov, I., Tsarev, A.: Cargo transportation models analysis using multi-agent adaptive real-time truck scheduling system. In: ICAART 2013 - Proceeding 5th International Conference Agents Artificial Intelligence, vol. 2, pp. 244–249 (2013)
25. Lin, F., Dewan, M.A.A., Nguyen, M.: Optimizing rescheduling intervals through using multi-armed bandit algorithms. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 746–753 (2018)
26. Smith, R.G.: The Contract Net Protocol: high-level communication and control in a distributed problem solver. IEEE Trans. Comput. **29**(12), 1104–1113 (1980)
27. Schillo, M., Kray, C., Fischer, K.: The eager bidder problem: a fundamental problem of DAI and selected solutions. In: Proceeding 1st International Joint Conference on Autonomous Agents Multiagent System, pp. 599–606, January 2002
28. Aknine, S., Pinson, S., Shakun, M.F.: An extended multi-agent negotiation protocol. Auton. Agent. Multi. Agent. Syst. **8**(1), 5–45 (2004)
29. Warden, T., Porzel, R., Gehrke, J.D., Herzog, O., Langer, H., Malaka, R.: Towards ontology-based multiagent simulations : plasma approach (2010)
30. Bellifemine, F., Caire, G., Greenwood, D.: Developing multi-agent systems with JADE (2007)