

A MULTIAGENT SIMULATOR FOR SUPPORTING LOGISTIC DECISIONS OF UNLOADING PETROLEUM SHIPS IN HARBORS

Robison Cris Brito*

Cesar Augusto Tacla

Lúcia Valéria Ramos de Arruda

Graduate School of Electrical Engineering and Computer
Science (CPGEI)

Federal Technological University of Paraná (UTFPR)

Curitiba – PR, Brazil

robison@utfpr.edu.br

tacla@utfpr.edu.br

lvrarruda@utfpr.edu.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 07/2009; aceito em 03/2010

Received July 2009; accepted March 2010

Abstract

This work presents and evaluates the performance of a simulation model based on multiagent system technology in order to support logistic decisions in a harbor from oil supply chain. The main decisions are concerned to pier allocation, oil discharge, storage tanks management and refinery supply by a pipeline. The real elements as ships, piers, pipelines, and refineries are modeled as agents, and they negotiate by auctions to move oil in this system. The simulation results are compared with results obtained with an optimization mathematical model based on mixed integer linear programming (MILP). Both models are able to find optimal solutions or close to the optimal solution depending on the problem size. In problems with several elements, the multiagent model can find solutions in seconds, while the MILP model presents very high computational time to find the optimal solution. In some situations, the MILP model results in out of memory error. Test scenarios demonstrate the usefulness of the multiagent based simulator in supporting decision taken concerning the logistic in harbors.

Keywords: oil industry supply chain; inventory management; multiagent system; MILP.

Resumo

O objetivo deste artigo é apresentar e avaliar o desempenho de um modelo de simulação baseado em sistemas multiagentes para auxiliar a tomada de decisão na alocação de petróleo em complexos portuários. Os diversos elementos do problema são modelados como agentes e negociam por meio de leilões a alocação dos inventários de óleo. Os resultados obtidos são comparados com resultados gerados por modelos de otimização matemática, estes baseados em programação linear inteira mista. Esses modelos são capazes de encontrar soluções ótimas ou próximas da ótima dependendo do tamanho da instância testada. Em problemas com muitos navios e tanques, o modelo baseado em sistema multiagente encontrou soluções em segundos, enquanto os modelos baseados em otimização matemática apresentaram problemas de tempo computacional e falta de memória, não encontrando a solução ótima. Os diversos exemplos aqui apresentados evidenciam a necessidade do modelo de simulação baseado em multiagentes no auxílio a tomada de decisões logísticas de portos

Palavras-chave: cadeia de suprimento de petróleo; alocação de inventário; sistema multiagente; PLIM.

1. Introduction

Recent researches (Kwon, Paul Im & Lee, 2007; Wang, Liu, Wang *et al.*, 2008; Zarandi, Pourakbar & Turksen, 2008) have used the multiagent system (MAS) paradigm to develop computer models for supply chain management. These models are based on the natural correspondence between the actors of a supply chain (factories, customers, suppliers) and agents, facilitating the model development. Moreover, supply chains tend to be decentralized. Their actors work independently, in accordance with their own interests and policies, but they share information and resources to achieve the overall goal of the supply chain.

In other approaches, such as those based only on simulation or optimization models, it is usual to centralize information and processing of supply chain. Thus the real world is represented by a model where a central element has the responsibility for all decisions. Such central element needs specific information to all others involved areas (e.g. sales, purchase, and inventory). In contrast, in real supply chains, the information is decentralized and each element has specific knowledge and, consequently, has only partial and incomplete knowledge about the entire supply chain.

However, beyond some drawbacks of optimization mathematical models, they are commonly used to solve scheduling problems in supply chains. The main issue is concerned with the reliability of the obtained results. These models can ensure the achievements of an optimal solution due to the use of an objective function and constraint satisfaction.

In the specific case of the oil industry, there are in the literature different optimization mathematical models dealing with parts and or whole supply chains (Neiro & Pinto, 2004). Most of these models are based on Mixed Integer Linear Programming (MILP). In spite of their ability in compute optimal solutions, these models fail to solve real scenarios from the industry. In general, real world problems modeled by MILP present a great number of integer and continuous variables that require much computational efforts to attain the optimal solution.

In this context, this paper proposes a multiagent simulation system (MAS), inspired by MILP models proposed in literature, to optimize the harbor operations in the petroleum supply chain. The activities are related to oil discharge, local storage, and oil transport to refineries that are a part of the oil supply chain. The simulator aim is to assist the scheduling of such activities in order to maximize the profit. The crude oil is discharged from ships and stored in the harbor. In the following, the oil must be sent to refineries by pipelines respecting operational constraints, such as keeping the refineries with minimum inventory and respecting decanting times in tanks. In the proposed system, agents are autonomous and work in coordination to achieve the overall objective – to allocate the inventory of ships in a profitable way. The results produced by the MAS are compared with three MILP models proposed by Más (2001), Veiga (2006) and Boschetto *et al.* (2009). The MAS must also to find solutions to real size problems of this oil supply chain subsystem to which the others three models have failed.

In section 2, the problem of oil allocation in a harbor is described. Section 3 presents the methodology used to develop the proposed MAS. Section 4 presents the obtained results for scenarios with three, four and seven ships. The comparison with the others three models based on mathematical optimization is also presented. Finally section 5 presents the conclusion and suggestions for future works.

2. Problem Description

The harbor problem has been firstly proposed by Más (2001), and subsequently it is used as reference to compare results of the models proposed by Veiga (2006) and Boschetto *et al.* (2009). Más (2001) proposes a solution to optimize the resources allocation in oil harbors. As an oil supply chain subsystem, the harbor has a complex structure, consisting of piers, storage tanks and a pipeline network, connecting these elements to refineries. Refineries have its own tank farm and distillation units that can consume oil directly from pipelines at a known flow rate. The pipeline network has intermediate storage substations performing oil decantation in order to ensure oil quality and minimum stocks at the refineries. Refineries can never stop to work due to the lack of oil to process or to their own consumption. Refinery stops only occur in special cases to planned maintenance or accident.

Due to this system complexity, Más (2001) has decomposed the problem into two distinct parts. The first one involves the harbors and the second one corresponds to the substations. In Figure 1, the big ellipse labeled with Port represents the Harbor Model which includes the allocation of ships to piers, the allocation of storage tanks in the harbor, the oil decantation time control, and the oil delivery through pipelines. The ellipses Sub A and Sub B refer to the Substation Models and concern the transfer between the harbor's pipelines and the substations tanks, and the transfer of the substations' tanks to other pipelines linked to a terminal or refinery.

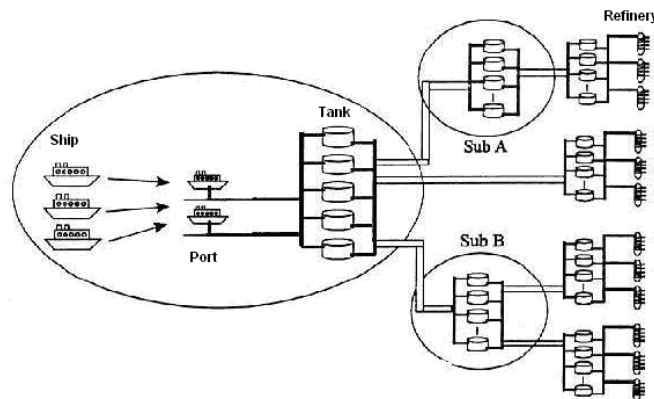


Figure 1 – Problem decomposition into Harbor Model and Substation Model (Más, 2001).

The main physical constraint on the Harbor Model is related to the oil storage, since there are no dedicated tanks to store each different type of oil carried by ships. Therefore, the oil types are grouped into classes, thus one tank or a set of dedicated tanks can store a specific class of oil. Different types of oil belonging to the same class can be stored in the same tank and this does not damage the oil characteristic quality.

The different types of oil arrive at harbor only by ships, which should completely discharge their inventory in one of the piers. Each ship has a time table with arrival and departure times from the harbor, which should be respected. The ship must pay a fee due to over-stay if it quits harbor after the specified departure time. The ship can discharge the oil to any tank at the harbor, if the tank class is compatible with the type of the discharged oil. The oil volume in a tank must respect the tank minimum and maximum limits, and a tank can only be discharged through a pipeline after a decanting period over 24 hours.

In this paper, the substations are not taken into account. It is considered that after the decanting period, the tanks send oil through pipelines directly to the refineries, thereby simplifying the process. The substations are treated in the Model of Substations (Más, 2001).

The cost of the interface must be considered when defining the order to move the classes of oil from the harbor's tanks to the refineries. The interface is a low cost product that is inserted between different classes of oil in order to prevent contamination during pipeline transportation. The cost of the interface depends on how different are the two classes of oil to be transported in sequence in the same pipeline.

The model proposed by Más (2001) considers three monetary values for the oil inventory: the first value is assigned per cubic meter for each type of oil in the ship. The operation of unloading a type of oil in the harbor, adds value to it. Thus the old value is replaced by a new one, but now oil types are referenced by their class. When transported to the refinery, the class of oil is valued again, reaching its greatest value in the considered supply chain. The difference between the value of the oil class in the refinery and its value in the harbor is called of nobleness. Therefore, there are noble classes and less noble classes of oil. There are tanks dedicated to only a certain classes of oil, thus one can say that tanks has also a degree of nobleness. In other words, if a noble class of oil is put in an incompatible tank, the oil is degraded and diminishes its value.

According to Veiga (2006) and Boschetto *et al.* (2009), although the solution of Más (2001) improve the transfer of oil from ships to refineries in relation to the costs involved, the computational time to obtain the solution is high (in the order of hours) compared to the time decisions that must be taken daily in the harbor (in the order of minutes). Moreover, for problems with larger number of ships, with larger number of classes of oil, and tanks with limited storage capacities, all of these are common characteristics in real-world scenarios, the model proposed by Más (2001) becomes quickly non-computable.

3. The Proposed Multiagent Simulator

In the development of the multiagent simulator, we model the information about the harbor problem as a multiagent system (MAS). According to Jennings & Wooldridge (1999), MAS are systems composed of agents that cooperate to solve problems that go beyond the agents individual capabilities. The main features of MAS are: each agent has limited competencies and/or partial information about the problem to be solved, the absence of centralized control, no globally available data and the existence of communication protocols that allow agents to coordinate their actions and exchange information independently of the application.

To fairly compare the results generated by the simulator with the results obtained by models that make use of mixed integer linear programming, such as Más (2001), Veiga (2006), and Boschetto *et al.* (2009), the proposed simulator has adopted the same restrictions and the same objective functions used by these authors.

3.1 System's Agents

One of the advantages in the MAS development is that the identification of agents is done in most cases in a one-to-one way, i.e., for each real element there is one surrogate entity in the MAS. Thus, for the proposed system, we identified the following types of agents:

- **Ship:** in the supply chain, it is responsible for providing oil. The Ship agent aims to unload its inventory of crude oil in the piers optimally, i.e. with lowest cost and as fastest as possible.
- **Pier:** when a Pier agent receives information about the arrival of a new ship, it competes with others to get the oil cargo since oil discharge occurs at only one pier. This competition is based on the cost of mooring and on the added value to the oil as a function of the available tanks connected to the pier.
- **Tank:** the goal of a Tank agent is to receive the largest amount of oil from ships. On receiving information of the piers about the arrival of a new ship, the Tank agent sends its proposal to the piers to receive the cargo of oil in competition or in collaboration with other Tank agents. The proposal can be to receive a part of or the whole ship's cargo. To elaborate the proposal, Tank agents should observe the decanting period after receiving oil and negotiate with Pipeline agents for sending their inventory to refineries.
- **Pipeline:** a Pipeline is responsible for sending oil from the harbor's tanks to the refineries. Upon receiving a request from a refinery, the Pipeline agent checks if the refinery supports the new inventory, and if so the download is scheduled;
- **Refinery:** a Refinery agent consumes crude oil at a known rate and constantly sends information to the Pipeline agent informing the refinery about its situation and schedule for receiving additional quantities of crude oil. This agent is the final link in the considered oil supply chain and its goal is to ensure that the refinery's inventory always respects the minimum and maximum established limits.

3.2 FIPA Contract-Net applied to the Harbor Model

To work cooperatively, the agents need to negotiate several times. After ship arrival, the piers must decide which of them will receive the ship's inventory taking into account the size of the ship and already planned discharges. For inventory transfer from a ship moored at a pier to a tank, the Tank agents must negotiate among them to decide which will receive the cargo and, afterwards, they also decide about tank unloading operation. Pipeline agents must negotiate to decide which one will transfer oil to the refinery.

In this paper, we not design a specific and complex heuristics for data exchange and decision making in order to implement the negotiation strategy. We chose to implement the Auction protocol Contract-Net (Smith, 1980) specified by (FIPA, 2008). The use of the standard protocol allows the simulator to be easily adapted to other cases, such as the Substation Model.

The Contract-Net protocol is well suited to the Harbor Model because there is competition for resources (piers compete for ships and tanks compete for the several types of crude oil). Another reason is that the Contract-Net meets the four minimum requirements of a good negotiation protocol defined by Kraus (2001): simplicity, efficiency, short negotiation time and stability.

Among the various models of auctions (Smith, 1980), we used the sealed-bid auction of prime price, which is a variation of the English Auction. In the original English auction, participants can submit multiple bids, and finally who sends the bid with the highest value wins the auction. The difference in the sealed-bid auction of Prime Price is that all participants have the right to submit a single bid. After the arrival of all bids, or after a time limit, the agent who started the auction selects the best bid. In the harbor simulator, auctions occur in three sequential moments as shown in Figure 2.

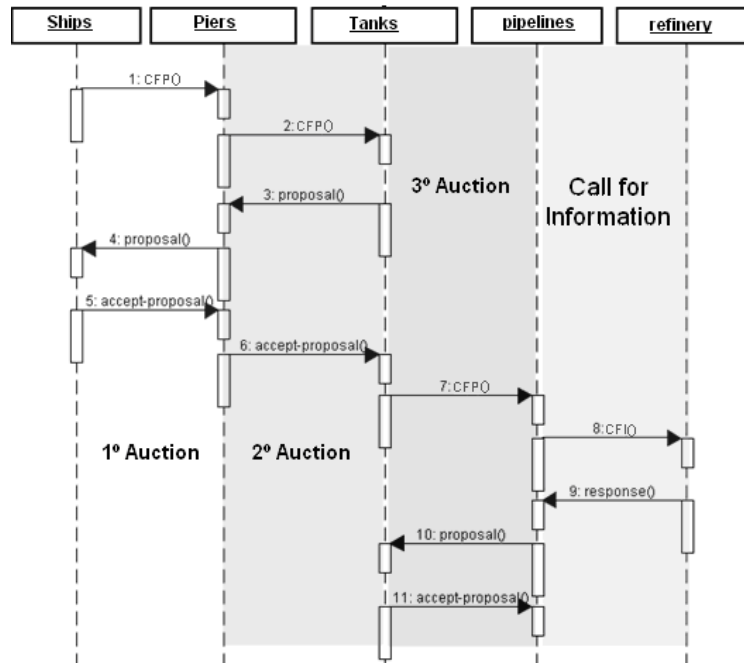


Figure 2 – Sequence diagram showing the auctions in the Harbor Model.

Initially, when a ship arrives at the harbor, it starts the first auction asking for proposals to the piers wishing to moor it. In Figure 2, this first auction starts with the message "1:CFP()" (i.e. call-for-proposals). Then, each pier receiving the ship's call-for-proposal, starts a second auction, requesting proposals from tanks for storing the ship's oil cargo ("2:CFP()") which is also a call-for-proposals.

Upon receiving the proposals of the tanks ("3:proposal()"), the piers select only one proposal if the ship's cargo can be moved on to a single tank, or more than one proposal when it is necessary to move the inventory to more than one tank. Thus piers compose their proposals, responding the ship with proposal messages ("4:proposal()"). If the pier cannot receive the ship because the ship is too large, it answers the call-for-proposal with a refuse message.

After receiving proposals from all piers or the time limit is reached, the ship selects the most profitable proposal and sends a message of acceptance to the pier that won the auction ("5:accept-proposal ()"). The pier, in turn, sends a message of acceptance to all the tanks selected to receive the inventory of the ship ("6:accept-proposal()").

A tank, upon receiving an acceptance notification from a pier, checks the status of its inventory already considering oil transfer. If the tank reaches its maximum capacity, it schedules the decantation. To do this, it starts a new auction by sending a third call-for-proposals for all the pipelines connected to it ("7:CFP ()"). The pipeline, upon receiving the call-for-proposals, sends a call-for-information message ("8:CFI()") to the refinery requesting the agenda of crude oil distillation. Based on the received agenda ("9:response()"), the pipeline agent composes a proposal to the tank. On receiving all the proposals from the pipelines ("10:proposal()"), the tank selects the best proposal and sends an acceptance message for the winner ("11:accept-proposal()"), ending the cycle.

Piers, tanks, pipelines, and refineries agents have individual agendas. Such agendas are consulted upon receiving of call-for-proposals avoiding simultaneous allocation of resources for two different requestors. When the winner Pier agent receives the accept-proposal from a ship in the first auction, the Pier agent adds on its agenda the initial and ending time of mooring. When the winner tank receives the acceptance message of the second auction, it adds the beginning and ending time of the transfer oil operation in its agenda. As a result of the third auction, the pipeline schedules in its agenda the time necessary to transfer the oil from the tank to the refinery.

To reduce the complexity of the system, no auctions are executed in parallel, i.e. the auction of a new ship starts only after the end of the auction of the previous one. This is because the allocating time to new ship used by MAS is small. Therefore, doing parallel auctions would add unnecessary complexity to the system.

3.3 Policies for Resource Allocation

In modeling the proposed MAS, the process of developing a model using MILP was examined. The aims were to combine the advantages of a MAS (decentralization) with the advantages present in the MILP model, that is, the ability to assure that the restrictions are respected and the optimal solution of objective function is reached when a solution is computed. In developing a MILP model, one must identify the objective function and model constraints.

3.3.1 Objective Function

The objective function implemented in simulator maximizes profit as the main criterion, i.e. the difference between revenue and costs in the harbor. The objective function given by Equation 1 is identical to the objective function of Más (2001), Veiga (2006) and Boschetto *et al.* (2009).

$$\begin{aligned}
 \max \quad & \sum_{r=1}^R \sum_{cl \in CLR_r} REV R_{cl,r}^{class} \cdot \left[\sum_{o \in OR_r} \sum_{TCL_{cl} \cap TO_o} \sum_{e=1}^{E-1} QUT_{t,o,e} \right] \\
 & + \sum_{cl} REV P_{cl}^{class} \cdot \sum_{t \in TCL_{cl}} (V_{t,E}^T - V_t^0) \\
 & - \sum_c COST_c^{crude} \cdot \left(\sum_{n \in NC_c} C_{n,c} \right) \\
 & - \sum_p COST_p^{pier} \cdot \left[\sum_{n \in NP_p} (\tau_{n,p}^{end} - \tau_{n,p}^{start}) \right] \\
 & - \sum_c COST_n^{se} \cdot T_n^{se} \\
 & - \sum_o \sum_{cl \in CLO_o} \sum_{\substack{cl' \in CLO_o \\ cl' \neq cl}} COST_{cl,cl'}^{face} \cdot \sum_{e=1}^{E-1} INT_{cl,cl',o,e}
 \end{aligned} \tag{1}$$

where

R	set of refineries;
CL	set of crude oil classes;
T	set of storage tanks;
N	set of ships;
P	set of piers;
O	set of pipelines;
E	set of events;
CLR_r	set of crude oil classes supplied to the refinery r ;
$REV R_{cl,r}^{class}$	Revenue of class cl in refinery r [\$/1000m ³];
OR_r	set of pipelines that supply oil to refinery r ;
TCL_{cl}	set of tanks able to storage oil of cl class;
TO_o	set of tanks able to be connected to pipeline o ;
$QUT_{t,o,e}$	inventory of crude oil moved of tank t to pipeline o due to the event e [1000m ³];
$REV P_{cl}^{class}$	Revenue of class cl at the harbor [\$/1000m ³];
$V_{t,E}^T$	Volume of oil in tank t at the end of simulation [1000m ³];
V_t^0	Initial volume of oil in tank t [1000m ³];
$COST_c^{crude}$	Cost of crude oil type c in the ships [\$/1000m ³];
NC_c	Set of ships able to transport crude oil type c ;
$C_{n,c}$	Cargo of crude oil type c in the ship n [1000m ³];
$COST_p^{pier}$	Cost of utilization of pier p [\$/h];
NP_p	Set of ships able to moor at pier p ;
$\tau_{n,p}^{start}$	arriving time of ship n in pier p [h];
$\tau_{n,p}^{end}$	departure time of ship n from pier p [h];
$COST_n^{se}$	Cost of over-stay for ship n [\$/h];
T_n^{se}	Duration of over-stay for ship n [h];
CLO_o	Set of crude oil classes able to be transported by pipeline o ;
$COST_{cl,cl'}^{face}$	Cost for interfacing classes cl and cl' [\$/];
$INT_{cl,cl',o,e}$	Integer value set to 1 if classes cl and cl' were interfaced in the pipeline o between events e and $e+1$. Otherwise, it is set to zero.

As the proposed simulator is a MAS, the objective function implementation is distributed in the logic of the various system agents. Each agent performs only the specific functions related to its activities in the supply chain. Thus the terms composing the objective function were distributed among the various agents as follows:

- Revenue of oil classes in the refineries: this term is computed by pier agents when they receive call-for-proposal messages from a ship. Pier agents choose tanks that do not impair oil to be stored trying to maximize oil value in the refinery. This revenue is maximized when using tanks of greater nobleness. The nobleness is the value assigned to each class of oil when moved from the harbor to the refinery, since the value of the class of oil in the refinery is greater than the value of the oil class in the harbor.
- Difference between final and initial revenues of oil in the harbor: as the goal is to move all oil from the harbor to the refineries by the end of the simulation, this term should not be taken into account by the agents. At the simulation end, the final oil revenue in the harbor is always the volume at the harbor's tanks (tanks cannot be completely emptied due to operational constraints) minus the volume of the harbor's tanks at the beginning of the simulation.

- Cost of crude oil on ships: this term is not taken into account by the agents, once the inventory of each ship is inputted to the MAS.
- Cost of piers utilization: Pier and Ship agents have responsibility for calculating this term. Pier agents should schedule docking, enabling faster unloading of oil. A Ship agent select among the piers that support it, the one with lower cost per hour of mooring.
- Cost of ship over-stay: the calculation of this term is due to pier agents, since their agenda should respect the period to ship arrive in the harbor and shall not exceed the ship departure date. If the ship departure is delayed, pier agents try to minimize the over-stay.
- Cost of interface between two classes of crude oil in the pipeline: Tank agents select among all the pipelines connected to them, those allow moving oil with the lowest interface cost respecting the nobleness of the oil class. The cost of the interface is not taken into account if the stock of a refinery is very close to the minimum. In this case, the refinery has priority in receiving crude oil regardless of the cost of the interface.

3.3.2 Constraints

In MILP models, constraints are represented by equations and inequations. In the simulator, the constraints are embedded in the internal logic of agents. The main constraints of the Harbor Model are the following ones according to Más (2001). Each one of them is represented in a particular agent:

- Each ship can only dock at one pier. This constraint is dealt with by ship agents when they receive proposals from piers. A ship agent selects at most one pier.
- The amount of crude oil to be discharged from the ship must be equal to the total cargo of oil in the ship. This constraint is treated by pier agents that upon receiving call-for-proposal messages from ship agents search for tanks to hold that entire inventory. Otherwise pier agents respond to the call-for-proposal with a refuse message, which indicates that the pier is not able to dock the ship.
- A tank cannot be loaded and unloaded at the same time. This constraint is treated by Tank agents before scheduling a new loading or unloading operation. These agents must check their agenda not allowing that both operations occur at the same time interval.
- A pipeline can be connected to only one tank at a time. The Pipeline agent checks its agenda to not allow two connections at the same time.
- A ship can be connected to only one tank at a time. This constraint is treated by Pier agents upon receiving a call-for-proposal message of a Ship agent. Pier agents select one or more tanks to receive the ship inventory. If more than one tank is needed, the Pier agent checks the Tank agents' proposals preventing the ship to use two tanks at the same time.
- When transporting two distinct classes of oil in a pipeline, an interface product must be put among products. This constraint is treated by the Pipeline agent which has an agenda of the last class transported. Thus, the Pipeline agent can compute the value of the interface that must be used.
- A tank can only transfer oil through a pipeline after a decanting period of at least 24 hours. This constraint is dealt with by the Tank agent when its current volume is greater than 50% of its maximum capacity. When this condition is true, the Tank agent schedules a 24 hours period for decanting, and sends call-for-proposal messages requesting the pipelines to transfer such oil volume after decanting time.

The presented constraints reflect operational limits. However when using continuous time representation in MILP models, as in the models proposed by Veiga (2006) and Boschetto *et al.* (2009), it is necessary to introduce time constraints in order to respect the events precedence. In MAS, these time constraints are not necessary, as explained in the following section.

3.3.3 Time representation in the MAS

Time in MILP models can be represented with discrete or continuous variables. A MILP model requires the definition of temporal constraints, for instance, the ending time of moving oil from a ship to a tank must be less than the ship departure time from the pier.

For the MAS such constraints are unnecessary, since the process of allocating resources happens sequentially, starting with the arrival of the ship at the harbor. This event fires a sequence of auctions (Figure 2) that aim to moor the ship in one of the piers, to distribute its cargo between the more profitable tanks, to schedule decanting operations for tanks and to determine which pipelines should transfer decanted oil from tanks to refineries. All of these events have a start time, an end time and quantifiers that inform, for instance, the volume of oil transferred from one place to another. As the use of resources is scheduled in a sequential way, operations in the agents' agendas do not suffer from temporal inconsistencies.

3.4 Architecture

The simulator's architecture is strongly influenced by the software environment used for the computational implementation of agents. We used the tool MOISE+ (*Model of Organization for multi-agent SystEm*) (Hübner, 2003) which is "an organizational model for Multi-Agent Systems based on notions like roles, groups, and missions". It enables a MAS to have an explicit description of its organization. The idea is to have a static specification of an organization that can be adopted by agents forming then an instance of the organizational specification. This instance is named organizational entity. Thus, agents can enter/leave the organizational entity, create groups, and adopt roles always respecting the organization specification.

MOISE+'s infrastructure is built with independent layers, where the elements of a layer provide services to the upper layer and obtain services from the lower layer. Thus, the simulator inherited a layered organizational architecture as shown in Figure 3.

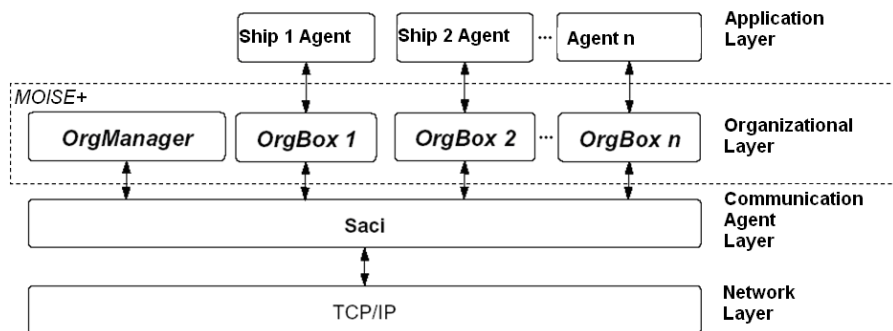


Figure 3 – Layers of MOISE+ (Hübner, 2003).

The characteristics of each layer according to Hübner (2003) are:

- Network Layer: this layer allows the communication among several computers where agents are running.
- Agent Communication Layer: it allows agents to exchange messages using an Agent Communication Language (ACL) called KQML (Knowledge Query and Manipulation Language). SACI (Simple Agent Communication Infrastructure) is a platform developed by Sichman & Hübner (2000) for supporting message exchanging among agents using KQML. SACI also controls distributed execution of agents through the scheduling of the agents' tasks.
- Organizational Layer: this layer provides services for consulting states of goals assigned to agents' organizations (for example, if a group of agents has already achieved a goal). It also verifies if the rules established for the functioning of an organization are satisfied (e.g. whether a group has the right number of agents required for the organization). This layer is formed by two types of components: OrgManager and OrgBox, both are distributed components. There is one OrgBox for each agent and one OrgManager for the whole MAS. The OrgManager is supposed to maintain the state consistency of the MAS, for example, preventing an agent to assume two incompatible roles. Every change in the organization (e.g. entry of an agent, group creation, adoption of a role) should pass through this agent. The OrgBox is an interface that agents use to access information about the organization. Whenever an agent wants to perform an action on the organization (e.g. commit to some mission) or send a message to another agent, it requests the service to its OrgBox which interact via SACI with the other agent's OrgBox.
- Application layer: the application agents (e.g. ships, piers) are in this layer. These agents have specific behaviors and interact with each other using their OrgBoxes.

The OrgManager notifies the OrgBoxes to keep them synchronized about the state changes of the MAS. For instance, in Figure 4, when the Ship 1 agent is created by the system, it receives message 1 from the OrgManager. This message tells the Ship 1 to accomplish a goal, i.e. to move its cargo to the harbors' tanks. Thus, Ship 1 starts an auction in order to select a pier for mooring and tanks for unloading its cargo (messages 2 to 7). At the end of the auction, the Ship 1 achieved its goal and sends message 8 communicating it to the OrgManager. OrgManager then multicasts message 9 informing the fact to all engaged agents. Agents' local visions are not totally synchronized with the current state of the organization, but they are updated whenever possible.

OrgBoxes provides a number of services to the agents:

- Communication;
- Generation of events about organization of agents' entities are inserted in;
- Information of agents' obligations;
- Information about possible goals.

The organizational layer facilitates coordination of agents, eliminating the communication between them needed to perform their actions in the correct sequence. It is responsibility of the MOISE+'s Organizational layer to coordinate the execution of actions.

Unlike most implementations of organizational models where the model is hard-coded in the agents, MOISE+ allows a declarative representation of organizations. Besides, MOISE+ allows the development of open systems (Hübner, 2003). In such systems, agents can be included and excluded from the organization and the only requirement for an agent to participate in an organization is to know KQML (to communicate with the OrgManager) or use the OrgBox (which encapsulates communication by KQML). An agent does not need to know the internal logic of others.

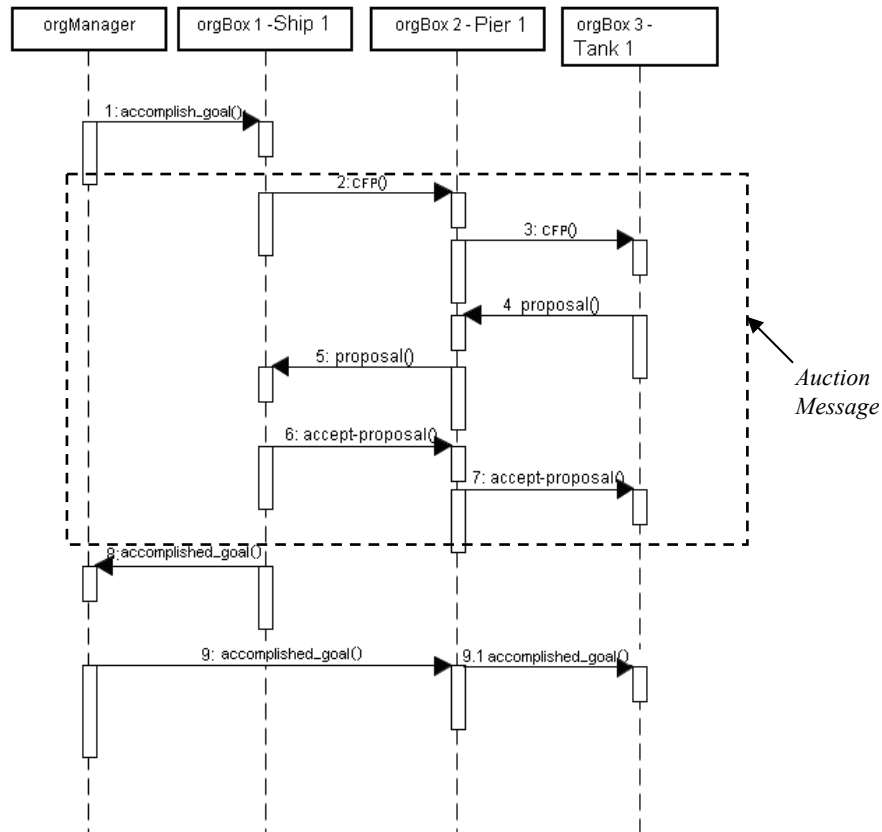


Figure 4 – Sequence diagram showing exchanged messages between OrgManager and OrgBoxes.

4. Results

To test the multiagent simulator, five scenarios were used. These scenarios correspond to simplified real situations took up from the harbor of São Sebastião in São Paulo. The data were extracted from Más (2001). In all scenarios there is only one pipeline and one refinery (OSVAT pipeline linking the São Sebastião-SP harbor to the REVAP refinery in São José dos Campos-SP). Differences between scenarios refer only to the number of ships, tanks and piers. Details of the scenarios are presented in Table 1.

A computer with Centrino 1.73 GHz processor and 1 GB of RAM was used during the tests. For the modeling and implementation of MILP models, the ILOG OPL Studio 3.6.1 was used with the CPLEX 8.1 solver (ILOG, 2002).

The results produced by the multiagent simulator are compared with results of approaches using MILP techniques, such as Más (2001), Veiga (2006) and Boschetto *et al.* (2009). For the sake of a fair comparison, the same objective function and the same constraints used by the aforementioned authors were used in the multiagent simulator.

Initially, one simulation was executed for each scenario. The scheduling horizon for each scenario (time required to send the entire volume of oil from the harbor to the refinery) and

the processing time spent by the simulator to compose the agenda for resource allocation in the harbor were saved for each scenario. These data were used as input parameters for the MILP model of Más (2001). This allowed comparing results considering the same scheduling horizon and the same runtime. Thus, the transferred oil volume at the end of the simulation is the same, since in both cases the models receive the same ships and send the entire inventory from the harbor to the refinery.

To validate their models Más (2001), Veiga (2006) and Boschetto *et al.* (2009) used three sets of tests with 3, 4 and 7 ships. Scenario 1 consists of a large (Front Brea), a medium (Quarries), and a small ship (Rebouças). Ships are loaded with different types of oil and have an arrival/departure date at/from the harbor, a cost of over-stay, a know flow rate for oil discharge and a time for entry and exit of some pier.

For the scenario with 4 ships, it was added to the previous scenario a new large ship (Muriae). For the scenario with 7 ships, three new large ships (Vergina II, North Star and President) were added, which increases the complexity of the problem.

For scenarios with 3 and 4 ships, due to the small number of vessels, only two piers (P-1 and P-2) were used. Each one has a different cost for docking. In the scenario with 7 vessels two more piers are added (P-3 and P-4). Pier P-4 only allows the mooring of small ships and, consequently, has a lower cost of mooring.

In scenarios with 3 and 4 ships, 5 tanks were used, one for each class of oil. As in the real world, the tanks have different capacities and initial volumes, as well as a maximum flow rate for loading and unloading and a time for decanting. In the scenario with 7 ships were used 10 tanks, due to the great volume of crude oil to be transferred from ships. There are two tanks for each class of oil.

In all scenarios the refinery REVAP planning was used. This refinery has a known rate of consumption of crude oil and a known initial volume of oil. As in the real world, the harbor has to keep the refinery oil inventory into minimum and maximum limits.

Monetary values for the types and classes of crude oils are given for the three steps of the process (in the ship, in the harbor and in the refinery) as well as the costs of interfacing different classes of oil in pipelines.

For comparisons, for each scenario was obtained whenever possible the best result called the global optimal solution. This global optimal solution is obtained by executing the model of Más (2001) without limitation of computational time. From this point forward, the model developed by Más (2001) is referenced as original model. In order to find feasible results in shorter processing times, Más (2001) also proposed to use a margin of optimality to compare some scenarios.

The model of Veiga (2006) modifies the model of Más (2001) by introducing a hierarchical decomposition of the problem based on three key elements of scheduling: resource allocation, activity sequencing and timing. The goal of this decomposition is to reduce the computational time for larger instances of the problem.

The model proposed by Boschetto *et al.* (2009) is a continuous time MILP model based on the original model (Más, 2001) and implemented with a preprocessing step, based on the theory of constraints (TOC), aiming at reducing the computational time. The preprocessor provides parameters for the MILP model, based on the solution of a simplified model without some of original constraints. Like the model of Veiga (2006), Boschetto's model provides only feasible solutions, not ensuring finding the global optimal solution.

Table 1 – Input data for experimental scenarios.

Ship	Arrival (hours)	Departure (hour)	Inventory (x1000m ³)	C. Demurrage (\$/h)	Flow rate min/max (1000m ³ /h)	Time Arr./Dep. Pier (hours)	
Front Brea (N1)	0	48	106 (oc-05)	0,8333	0,0/8,0	2/2	
Pedreiras (N2)	12	60	38 (oc-08) 20 (oc-27)	0,4167	0,0/8,0	2/2	
Rebouças (N3)	24	72	30 (oc-38)	0,4167	0,0/8,0	2/2	
Muriae (N4)	48	48	120(oc-01)	0,8333	0,0/8,0	2/2	
Vergina II (N5)	72	120	92(oc-38)	0,8333	0,0/8,0	2/2	
North Star (N6)	96	144	135(oc-26)	1,6667	0,0/8,0	2/2	
Presidente (N7)	97	144	80 (oc-27)	0,8333	0,0/8,0	2/2	
Number of Piers			4				
Piers		Capacity to Moor		Cost per Hour (\$/h)			
P-1		N1, N2, N3, N4, N5, N6, N7		5,0314			
P-2		N1, N2, N3, N4, N5, N7		2,5157			
P-3		N1, N2, N3, N4, N5, N7		2,5157			
P-4		N2, N3		1,2579			
Number of Tanks			10				
Tank	Capacity Min/Max (x1000m ³)	Initial Volume (x1000m ³)	Class	Types of crude oil per class	Max flow rate (1000m ³ /h)	Time to decant (hours)	
TQ3234 (TQ1)	9.766/77.077	60.630	cl-1	oc-01;05;08;38	4.470	24	
TQ3235 (TQ2)	9.282/77.224	9.858	cl -1	oc-01;05;08;38	4.470	24	
TQ3237 (TQ3)	10.457/73.077	18.232	cl -3	oc-01;05;08;26;29	4.390	24	
TQ3238 (TQ4)	3.235/ 33.395	31.001	cl -3	oc-01;05;08;26;29	4.390	24	
TQ3239 (TQ5)	9,654/77.355	9.997	cl -4	oc-01;05;26;27	4.429	24	
TQ3240 (TQ6)	11.433/73.352	60.629	cl -4	oc-01;05;26;27	4.429	24	
TQ3241 (TQ7)	11.370/73.745	69.524	cl -5	oc-01;05;26;27	4.447	24	
TQ3242 (TQ8)	9.137/72.718	11.385	cl -5	oc-01;05;26;27	4.447	24	
TQ3243 (TQ9)	11.022/77.001	39.015	cl -6	oc-01;05;27;38	4.511	24	
TQ3244(TQ10)	10.819/72.589	40.513	cl -6	oc-01;05;27;38	4.511	24	
Number of refinery/pipeline			1 (REVAP-PLAN) / 1 (OSVAT)				
Refinery	Initial Content (x1000m ³)	Content min/max (x1000m ³)			Consumption per hour (1000m ³ /h)		
REVAP-PLAN	920	186.06/1515.487			3.625		
Value of oil in the refinery (\$/1000m³)		cl-1:135.2201; cl-3:132.0755; cl-4:128.6037; cl-5:132.8302; cl-6:137.1069					
Value of oil in the harbor (\$/1000m³)		cl-1:129.8113; cl-3:126.7925; cl-4:128.6037; cl-5:127.2956; cl-6:131.6227					
Value of oil in ship (\$/1000m³)		oc-05:116.9811; oc-08:109.7484; oc-27:100.6289; oc-38:114.7798					
Cost of Interface (\$)			cl-1	cl-3	cl-4	cl-5	cl-6
		cl-1	X	1.57235	0.62895	1.57235	0.94340
		cl-3	1.57235	X	0.94335	3.14465	2.51570
		cl-4	0.62895	0.94335	X	2.20130	1.57235
		cl-5	1.57235	3.14465	2.20130	X	0.62895
cl-6	0.94340	2.51570	1.57235	0.62895	X		

Table 2 presents a comparison of results for scenario 1 with three ships. The MAS model stands out from the others because it was able to find the global optimal solution in 8 seconds of processing, while the original model taken 5h31m23s running in the same computer (Centrino 1.73 GHz, 1GB RAM).

To a fixed computational time of 8 seconds (the lower computational time obtained with the MAS), the original model achieved a solution with a profit of \$5.287,52, i.e. \$194.32 (3.67%) less than the global optimal solution, and a scheduling horizon of 93h, i.e. 3 hours greater than the global optimal solution. The model with pre-processing found a lower value for the objective function, i.e. \$11.30 (0.21%) less than the global optimal solution. The same happened with the model with a margin of optimality which was \$96.35 (1.82%) lower. In both situations the total scheduling horizon was 96 hours (3 hours greater than the global optimal).

Table 2 – Results for the scenario 1 with three ships.

	Original Model: global optimal (Más, 2001)	Original Model with time limitation (Más, 2001)	Model with margin of optimality (Más, 2001)	Model with Preprocessing (Boschetto <i>et al.</i> , 2009)	MAS
Execution Time	5h31m23s	8s	58s	12m24s	8s
Objective Function	5.287,52	5.093,20	5.191,17	5.276,22	5.287,52
Scheduling Horizon	93 hours	96 hours	96 hours	96 hours	93 hours
Computer	Centrino 1.73 GHz, 1GB RAM	Centrino 1.73 GHz, 1GB RAM	Pentium III 450MHz, 128MB RAM	AMD Athlon 64 2GHz, 1GB RAM	Centrino 1.73 GHz, 1GB RAM

The scenario 2 with four ships was used by Boschetto *et al.* (2009) in their model with preprocessing. For this scenario, the original model was executed without computational time limitation until it could find the global optimal and also with time limit of 11s. Table 3 presents the results of executions for the scenario with four ships.

Table 3 – Results for the scenario 2 with four ships.

	Original Model: global optimal (Más, 2001)	Original Model with time limitation (Más, 2001)	Model with Preprocessing (Boschetto <i>et al.</i> , 2009)	MAS
Execution Time	9h23m10s	11s	54m01s	11s
Objective Function	5.915,32	5.390,68	5.713,71	5.868,53
Scheduling Horizon	96 hours	96 hours	96 hours	96 hours
Computer	Centrino 1.73 GHz, 1GB RAM	Centrino 1.73 GHz, 1GB RAM	AMD Athlon 64 2GHz, 1GB RAM	Centrino 1.73 GHz, 1GB RAM

The MAS model has achieved a result close to the global optimal solution, with \$46.79 (0.79%) less than the global optimal, taking only 11 seconds of processing. This time is much less than the processing time of the original model to find the global optimal solution (9h23m10s). The original model with time limitation got an objective function value below the SMA at \$477.85 (8.14%) for the same processing time (11s). The model with pre-processing achieved a lower objective function value at \$154.82 (2.63%) compared to the global optimal solution, taking more processing time than the SMA. It should be noted that the scheduling horizon was the same for all models (96 hours), regardless of the value achieved for the objective function.

The scenario 3 with seven ships is the more complex one. It was modeled by Más (2001) with optimality margins for finding feasible solutions. Veiga (2006) has also modeled this scenario using a hierarchical approach in order to improve the solution quality and processing time. For this scenario, the original model is not able to find the global optimal solution due to an out of memory error. Thus, the global optimal result was not taken into account in the comparisons presented in Table 4.

Table 4 – Results for the scenario 3 with seven ships.

	Model with margin of optimality (Más, 2001)	Original Model with time limitation (Más, 2001)	Hierarchical Model (Veiga, 2006)	MAS
Execution Time	2h38m26s	20s	49m14s	20s
Objective Function	14.141,32	11.150,79	14.670,79	15.511,59
Scheduling Horizon	168 hours	168 hours	168 hours	168 hours
Computer	Pentium III 450MHz, 128MB RAM	Centrino 1.73 GHz, 1GB RAM	Pentium IV 2.4 GHz, 1GB RAM	Centrino 1.73 GHz, 1GB RAM

For the scenario 3 with seven ships, Más (2001) and Veiga (2006) used a fixed horizon of 168 hours, thus the objective function is calculated from a monetary perspective when the scheduling horizon achieved 168 hours. Operations occurring after that date were discarded. Within a 168 hours scheduling horizon, the entire inventory cannot be transferred from the harbor to the refinery, thus the objective function represents a partial solution to the scenario. For setting a reference value, the MAS calculated a feasible solution for the entire transfer of oil from the harbor to the refinery within a horizon of 200 hours and an amount of \$16,290.82 for the objective function. So to make the comparison with the other two models, the MAS was executed with a fixed scheduling horizon of 168 hours obtaining \$15,511.29 for the objective function, which was compared with the other two models.

The results highlight the superiority of the MAS simulator which achieved an objective function 3.74% greater than the objective function obtained by Veiga (2006), and 9.69% greater than the objective function obtained by Más (2001). This better solution was computed in 20 seconds. Comparing the objective function obtained by the MAS with the original model with time limit, the superiority of the MAS was \$4,360.80 (28.11%).

The graphic in Figure 5 shows the superiority of the MAS over the other approaches considering values of the objective function. With regard to the processing time, the MAS found the results in seconds, while the other approaches found the results in minutes and sometimes in hours.

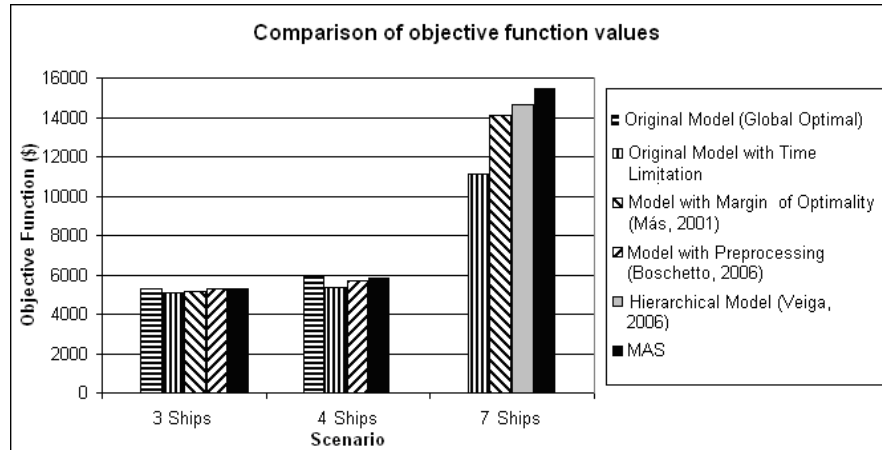


Figure 5 – Comparison among objective functions' values of the different approaches.

4.1 Computational Time vs. Solution Quality

The MAS showed better results when compared to the original model proposed by Más (2001) in all tests taking the processing time as measure. In tests where the processing time was not considered (in the comparisons with the global optimal solutions), the performance of the MAS was very good. In the scenario 1 with three ships, the MAS found the global optimal solution. In the scenario 2 with four ships, the MAS found a solution very close to the global optimal solution. In the scenario 3 with seven ships, the MAS achieved a feasible solution as the original model could not find the global optimal solution because it requires more processing power than available for the experiments.

From the experiments, one can observe that as new elements are added to the harbor environment, which augments the complexity of the problem, the worse results were found by the original model for a priori set processing time.

Taking into account the processing time, the superiority of the MAS is even greater, recalling that both approaches, the original and the SMA, use the same objective function. The difference is the method used to find the solution. While the mixed integer linear programming generates graphs and through the branch-and-bound algorithm searches the global optimal solution, the MAS allocates a resource at a time, not looking for the global optimal. Instead, it searches the optimal solution for a specific situation, i.e. the local optimal.

For instance, in the scenario 1 with three ships, when the first ship arrives at the harbor, the MAS tries to allocate the best available elements at the moment: the lower cost pier, the more profitable tanks, with minimal over-stay. Once the first ship's cargo is allocated, the agendas of the resources are updated and become heavy constraints that can no longer be modified. These heavy constraints in an MILP model would eliminate lots of possibilities in the original model. However, with this pruning action, the MAS can also discard the global optimal solution and, therefore, sometimes a global solution cannot be found by simulation.

When the second ship arrives at the harbor, its cargo is allocated respecting the agenda of the previous ship. It means the second ship uses only the resources which are still free. Thus, the gain in processing time is very large and the loss in quality is small.

When working with many resources, as in the scenarios with seven ships, the original model works with many variables, trying to find the best solution among all possible alternatives. This procedure may require too much memory and computational time, which often just makes it impossible for processing all alternatives due to combinatorial explosion.

The chart of Figure 6 shows the processing times (in seconds) for the MAS model to achieve solutions for five scenarios with different numbers of ships and tanks. These results are the same reported in Table 1. The time curve grows linearly and it depends on the number of ships.

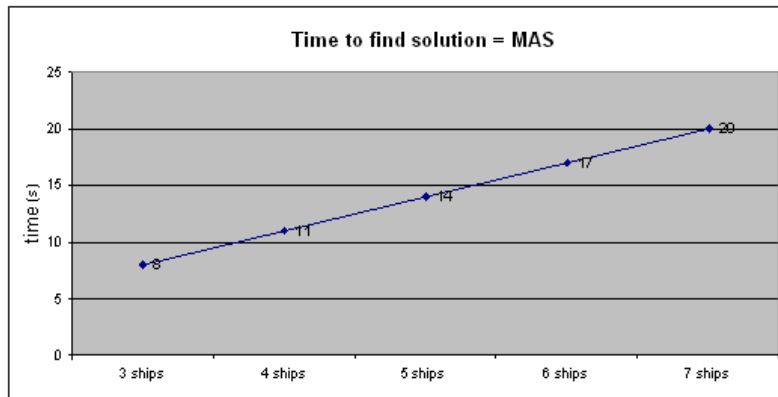


Figure 6 – Time for the SMA to find solutions in the various scenarios.

Figure 7 depicts the time spent by the original model (Más, 2001) to find the global optimal solution for various scenarios. Since the original model only found optimal solutions for scenarios with three and four vessels, scenarios with one and two ships were executed in order to depict the processing time curve for this model. All scenarios have two piers, five tanks, one pipeline, and one refinery. Thus scenario is not very complex, allowing the MILP model to find the global optimal solutions. In Figure 7, the time grows exponentially, and the number of possibilities and variables in the mathematical model can also grow in the same manner. This increase in the number of variables and possibilities causes the error out of memory as it occurred for the scenario with seven ships.

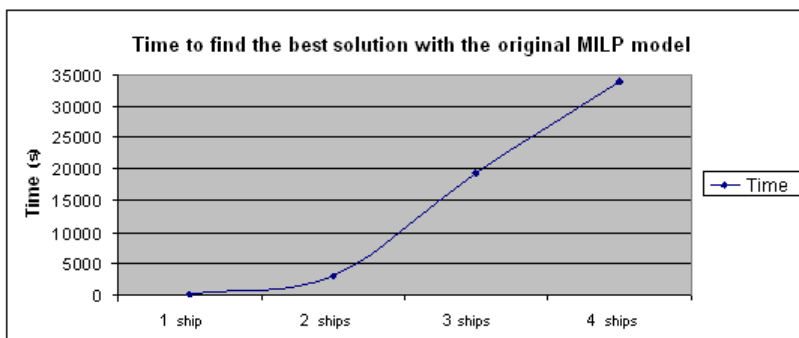


Figure 7 – Time to the original MILP model for finding global optimal solutions for the various scenarios.

Taking into account the results obtained for fixed processing times, one can observe that the MAS achieves better results than the models based on MILP, and this difference increases with the complexity of the scenario. In Figure 8, one can observe that increasing the number of ships in the scenarios, also increases the difference between the MAS' objective function and the original model's one. The difference represented in the graph is calculated by the following equation where *dif* is the difference, *ObjMAS* is the value for the objective function for the MAS, and *objMILP*, the same for the original MILP model:

$$dif = \left(\frac{ObjMAS - ObjMILP}{ObjMILP} \right) * 100$$

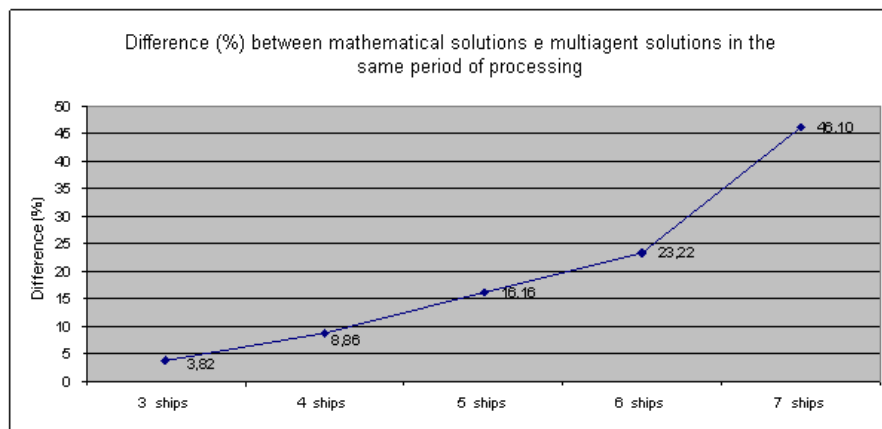


Figure 8 – Superiority (%) of the solutions found by the MAS compared to the solutions produced by the original MILP model for a fixed processing time.

It was also compared the best solution found by the original MILP model, whether the global optimal solution or the best feasible solution found, with the solution produced by the MAS. The difference is expressed in percentage. Positive values mean that the highest value was reached by the MILP model, and negative values represent the superiority of the MAS (Figure 9).

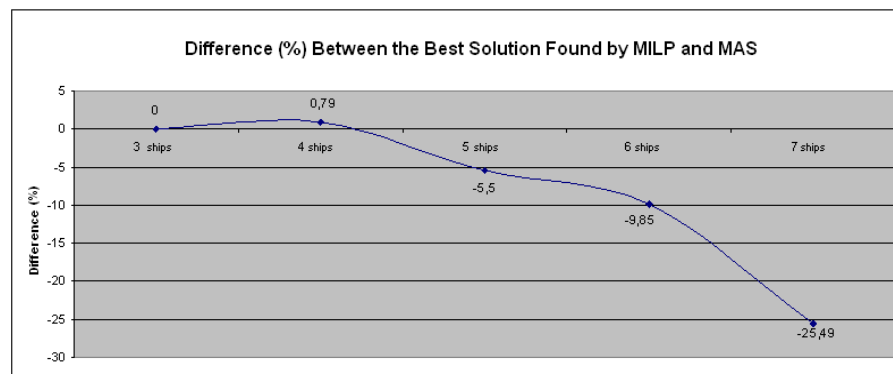


Figure 9 – Difference (%) between the best solutions found by the original MILP model and by the MAS.

For the scenario with three ships, due to their low complexity, the MAS found the same result as the original MILP model (the global optimal solution). With four ships, the original model was superior to the MAS obtaining an objective function value 0.79% greater than the result obtained by the MAS. In other scenarios (with 5, 6 and 7 ships), the MAS was superior because the original model exits by out of memory error before terminating its execution.

Besides the lower processing time to find feasible solutions and to find solutions even in complex scenarios, another advantage of the MAS is the possibility of working with online processing, allocating resources at runtime, unlike the MILP models, which are characterized by batch processing, which hinders interacting with users in a “what if” manner.

5. Conclusion and future work

This paper presented a simulation model based on multi-agent systems to assist decision making in the oil allocation in harbor sites. The MAS model was inspired from a model proposed in the literature (Más, 2001) which is based on mixed integer linear programming.

The architecture of the MAS model was organized in layers according to the tasks necessary to the interaction of agents. Unlike most multi-agent architectures in the literature, the proposed MAS model was built as an open system in which agents can easily be included or excluded. These system properties were made possible because the organizational layer was developed with MOISE+ and with the communication and distribution environment called SACI. Due these characteristics, the proposed simulator enables the decision maker to work with online processing, allocating resources (ships, piers, tanks) at runtime and getting solutions without having to reconfigure the model each time a change happens in the scenario.

To implement the strategy of negotiation and cooperation between agents, it was chosen to not develop a specific protocol to the problem. All negotiation among agents was done through the FIPA implementation of the Contract-Net auction protocol. The use of a standard protocol allows the simulator to be easily adapted to other sub-systems of the supply chain of oil and to other problems of decision making similar to those treated here.

To validate the multi-agent simulator, several typical scenarios of a harbor operated by PETROBRAS were used. These scenarios correspond to simplified real situations for the harbor of São Sebastião in São Paulo. This harbor is connected to the REVAP refinery located at São José dos Campos-SP by the OSVAT pipeline. The scenarios have different number of ships, tanks, and types of oil. Furthermore, results obtained with the MAS were compared to results obtained with the original MILP model (Más, 2001) and two other models based on MILP (Veiga, 2006) (Boschetto *et al.*, 2009). These comparisons were made in terms of computational time and optimality of the solution obtained for the different scenarios.

The MAS showed better results compared to the other models in all tests having the processing time as measure. In tests where the quality of the solution was used as measure, the MAS found the global optimal solution in the scenario with three vessels. It found a solution very close to the global optimal solution in the scenario with four ships, and it found better results in the scenario with seven ships in terms of optimality for a fixed scheduling horizon.

It was also possible to observe from the experiments as long as new elements are added to the harbor (i.e. the complexity of the problem grows up), the results found by the MILP models with fixed processing time are worse. This happens, because when the MAS allocate the first ship to some pier and tanks in the best possible way, the agendas of the allocated

resources are updated and become heavy constraints that cannot be modified. When the second ship arrives, resources are allocated respecting the heavy constraints determined by the previous ship. Thus the second ship has only the resources that are still free. This causes a gain in processing time with little degradation in the quality of the solution.

When working with many resources, as was the case with seven ships, the MILP-based models work with many variables, trying to find the best solution among all possible alternatives. This procedure may require too much memory and computational time, which often causes a combinatorial explosion. This problem does not happen with the MAS simulator, since it treats the problem sequentially as the ships arrive, the tanks are emptied, and the pipeline is occupied according to oil inventory variations at the refinery.

As future work we intend to extend the multiagent simulator to treat the refinery model originally proposed by Más (2001) and then modeling the entire supply chain of the petroleum industry based on the model proposed by Neiro & Pinto (2004). It is also possible to use the multi-agent simulation as a pre-processing step to calculate model parameters for a MILP model in an approach similar to that used by (Boschetto *et al.*, 2009).

References

- (1) Boschetto, S.; Lüders, R.; Neves-Jr, F. & Arruda, L.V.R. (2009). Um Modelo de Otimização da Operação de Terminais Petrolíferos Usando a Teoria das Restrições como Pré-Processamento. *Pesquisa Operacional*, **29**(1), 1-21, Janeiro a Abril de 2009.
- (2) FIPA – Foundation For Intelligent Physical Agentes. (2009). *FIPA Contract Net Interaction Protocol Specification*. Retrieved April 14, 2010, from <<http://www.fipa.org/specs/fipa00029/SC00029H.pdf>>.
- (3) Global Supply Chain Forum. (2008). Retrieved April 14, 2010, from <<http://fisher.osu.edu/centers/scm>>.
- (4) Hübner, J.S. (2003). Um Modelo de Reorganização de Sistemas Multiagente. PhD Thesis, Universidade de São Paulo, São Paulo.
- (5) Hubner, J.F. & Sichman, J.S. (2000). SACI: Uma Ferramenta para Implementação e Monitoração da Comunicação entre Agentes. *IBERAMIA 2000*, SBIA 2000.
- (6) ILOG (2002). *ILOG OPL Studio 3.6.1: Language Manual*. ILOG Corporation, France.
- (7) Jennings, N.R. & Wooldridge, M. (1999). Agent-Oriented Software Engineering. *Nineth European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1999.
- (8) Kraus, S. (2001). Automated negotiation and decision making in multiagent environments. Book: *Multi-agents systems and applications*. Springer-Verlag New York, p.150-172.
- (9) Kwon, O.; Paul Im, G.; & Lee, K.C. (2007). MACE-SCM: A multi-agent and case-based reasoning collaboration mechanism for supply chain management under supply and demand uncertainties. *Expert Systems with Applications: An International Journal*, **33**(3), 690-705.
- (10) Más, R. (2001). Otimização da Programação de Suprimento de Petróleo. Master Thesis. Universidade de São Paulo, São Paulo.

- (11) Neiro, S.M.S. & Pinto, J.M. (2004). A general modeling framework for the operational planning of petroleum supply chain. *Computers and Chemical Engineering*, **28**, 871-896.
- (12) Simchi-Lev, D.; Kaminsky, P. & Simchi-Levi, E. (2000). *Designing and Managing the Supply Chain*. McGraw-Hill Higher Education, 2000.
- (13) Smith, R.G. (1980). The Contract Net Protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, **29**(12), 1104-1113.
- (14) Veiga, S.L. (2006). Modelo Hierarquizado para Scheduling de Suprimento de Petróleo. Master Thesis. Universidade Tecnológica Federal do Paraná, Curitiba.
- (15) Verdicchio, M. & Colombetti, M. (2002). Commitments for Agent-Based Supply Chain Management. *SIGecom Exch – ACM*, p.13-23.
- (16) Wang, M.; Liu, J.; Wang, H.; Cheung, W.K. & Xie, X. (2008). On-demand e-supply chain integration: A multi-agent constraint-based approach. *Expert Systems with Applications: An International Journal*, v.43.
- (17) Zarandi, M.H.F.; Pourakbar, M. & Turksen, I.B. (2008). A Fuzzy agent-based model for reduction of bullwhip effect in supply chain systems. *Expert Systems with Applications: An International Journal*, v.34.