INT J COMPUT COMMUN, ISSN 1841-9836
8(3):384-394, June, 2013.

An Agent-Based Solution for the Berth Allocation Problem

C. Cubillos, R. Díaz, E. Urra, D. Cabrera-Paniagua, G. Cabrera, G. Lefranc

Claudio Cubillos, René Díaz, Enrique Urra,
Guillermo Cabrera, Gastón Lefranc
Pontificia Universidad Católica de Valparaíso
Av. Brasil 2241, Valparaíso, Chile
claudio.cubillos@ucv.cl, imrede@gmail.com, enrique.urra@gmail.com,
guillermo.cabrera@ucv.cl, glefranc@ucv.cl

Daniel Cabrera-Paniagua
Escuela de Ingeniería Comercial
Universidad de Valparaíso
Pasaje La Paz 1301, Viña del Mar, Chile
daniel.cabrera@uv.cl

Abstract: This work presents the development of MABAP, a decision support system based on the agent technology that helps in solving the problem of berth allocation for ships within a port. The Berth Allocation Problem (BAP) regards the logistics involved in planning and controlling the berthing of vessels. A software architecture in terms of agents is presented; Berths and Ships representing the actors in the system, BerthRequest and BerthPlanner as representatives of ships and berths in the planning process, and finally the Dock and Central agents representing the dock or pier. The architecture modeling was done using PASSI methodology for the design of agent-oriented systems, and the implementation was done in JADE, a Java-based development environment for multiagent systems. To validate the resulting support system, tests were carried out in which the user can choose different port-policy scenarios, ranging from maximizing vessels throughput to maximize berths use.
Keywords: Artificial Intelligence, Decision support system, Multiagent architecture, Ports planning.

1 Introduction

Maritime traffic is of significant importance to our country and the world, as reported by the United Nations Conference on Trade And Development (UNCTAD) [1], which places Chile in the Top 20 of container traffic. However, while container traffic has increased, the percentage of change from year to year has been declining and it is expected that this trend will continue. The latter, is largely due to the world economical state. Under such settings, an enormous potential can be developed through the optimization of related processes, obtaining competitive advantage in the market. The operations involved in maritime traffic and more precisely the activities carried out in ports are one of the most complex ones within the transportation industry. Such complexity is due to three reasons: 1) the wide variety of involved actors, 2) the interaction under a highly dynamic environment and 3) the distributed nature of the problem. Of the vast number of problems present in port logistics, there is one referred as the Berth Allocation Problem (BAP), which can be defined as a tactical planning problem, where the objective is to assign, ideally optimally, the position of ships inside the port for loading and unloading, so as to minimize the costs of container movements within the port. The problem, in the most complete form, involves two stages: A first stage of assigning the ships to each section of the port, and then a second stage of sequencing each vessel in the temporary space of the different sections. This work focuses on the second stage, considering the presence of only one big section in the

Copyright © 2006-2013 by CCC Publications

port and its further division into berths. Given the complex, distributed and dynamic nature of the problem, the use of multi-agent technology reveals useful to build a distributed software architecture and get a flexible, adaptable and robust decision support system solution capable to deliver a better quality of service together with attempting to obtain competitive results close to the pareto-optimal frontier. The main contributions of the present work are 1) to provide the design of a decision support system based on agent technology for the berth allocation planning, 2) provide a distributed planning process based on a greedy insertion heuristic solver [15] and the contract net protocol [4], and 3) provide an implementation of such and its validation through tests that show how diverse port policies can be chosen.

2 Multiagent Technology

In literature there is no exact definition of what an agent is. However, one of the most cited is Wooldridge's definition [2]: "An agent is a computer system located in an environment that is capable of performing actions independently to achieve its design goals". While this definition identifies some characteristics of an agent, it is not clearly distinguishable from a conventional distributed system. However, the main mentioned feature in which there is a consensus is autonomy. In the rest, there is often little agreement as many of the features have different relevance depending on the domains. For example, the characteristic of learning can be desirable for a certain application, while for another one it would not only be unimportant, but even undesirable. Despite the differences, one can forget about a precise definition in order to identify certain common properties, such as [2]:

- Autonomy: agents are capable of task selection, prioritization, goal-oriented behavior, and decision making without human intervention.
- Social ability: they can "contract" other components through some communication and coordination, and cooperation in solving any given task.
- Reactivity: they can perceive the context in which they operate and react appropriately.

Multiagent Systems

Multiagent architectures arise from the need to develop advanced complex applications consisting of a number of subsystems that interact to achieve distribute intelligence among various actors, giving rise to the creation of multi-agent systems (MAS). The use of a multiagent architecture appears as an appropriate solution when dealing with physically distributed problems, or where experience is required for integrating heterogeneous technologies or where the problem in question is defined on a computer network. The use of a multiagent architecture features in these cases as the most suitable alternative to leverage a distributed solution, adaptable to changes in structure and environment. In addition, an associated approach will build a total system from different autonomous units.

In a MAS, subsystems are defined with absolute local decision-making, and therefore a definition of policies to cooperate, negotiate and coordinate their actions is necessary. Therefore we can find different types of agents according to their capabilities [8].

The PASSI Methodology

PASSI (a process for Agent Societies Specification and Implementation) [3] is a step-by-step methodology that allows to go from the requirements to the system code. Aimed at designing and developing multi-agent societies, it integrates design models and concepts from software engineering and object-oriented multi-agent systems using UML notation. Thus allows providing the engineering formalism required in such systems' design. The PASSI methodology consists of five phases at various design levels, and twelve steps in the process of building a multi-agent system. Examples of its use can be found in literature devoted to the design of a bookstore system [3], a virtual enterprise for transportation [16] and an open agent system [17], among others.

3 The Berth Allocation Problem

In container shipping, the Hub and Spoke model is widely adopted. The ocean-going ships, also called mother ships (mother vessels) operate between a limited number of transfer terminals (hubs). Smaller vessels (feeders) link the hubs to the other ports (spokes). In recent years, the mother vessels have increased markedly in size making transport up to 8000 TEU (Twenty foot Equivalent Units) and larger sizes are planned. Transshipment ports are large intermodal platforms and only a limited number of them handle a significant part of world traffic. However, when a ship arrives at a port, it must wait for a space to tie up to the dock. For this reason it has different sections, mooring points or Berths. In addition, there are Gantry Cranes available for loading/unloading and smaller vehicles for the rest of the transportation work, such as forklifts. Once docked at the berth, containers destined to the port must be unloaded and new containers directed to other ports must be loaded before the ship can continue its course. The demand on those ports for the load/unload procedure to run with maximum efficiency will become greater, as the shipping companies will continue to increase their fleet size and the capacity of new ships. The logistics involved in planning and controlling the docking of ships is called the Berth Allocation Problem (BAP). In the BAP, the port management should try that:

- The ships dock as soon as possible to ensure a rapid turnover.
- The forklifts load/unload the required containers in the shortest time possible.
- The cost of transshipment of containers is minimal.

A frequent problem associated with the latter has to do with the allocation of forklifts to each of the arriving ships, which is directly associated with the handling time and has a strong impact on the BAP, known as the Crane Scheduling Problem [5]. The problem can be represented in a two-dimensional space as shown in Figure 1, where the rectangles represent the ship whose dimensions are its length, including a safety margin and the handling time. These rectangles are positioned in the decision space without overlapping and meet certain restrictions. In the spatial dimension, there are restrictions on water depth (draft allowance) and the maximum distance in relation to the most favorable location along the pier, calculated considering the outbound-containers location and the space reserved for the inbound containers. In the temporal dimension, restrictions are expressed as time windows. Some of them are soft and can be relaxed by an appropriate penalty cost.

Related Work

In literature, many studies on the problem can be found. First, the BAP can be modeled in a discrete case if the dock is seen as a finite set of berths. In this case, the berths are described as fixed length segments or if the spatial dimension is ignored, as dots. When considering that the length of vessels is very variable, one could divide the dock in sections to make the assignment,

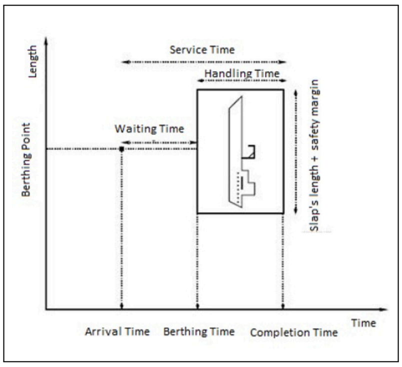


Figure 1: BAP two-dimensional (space vs. time).

although it would be difficult because the requirements vary dynamically. If large segments are used, space sub-use will happen in some cases, while smaller segments will make it difficult to find feasible solutions. To overcome this, a dynamic model is considered which defines that the ships can tie up anywhere in the dock. For the discrete case, the BAP can be modeled as an Unrelated Parallel Machine Scheduling problem [10], where each boat is seen as a work and each docking point as a machine. The time of arrival of each vessel is the release time of the work. In the continuous case, the BAP is a two-dimensional problem of the Cutting Stock Problem with additional constraints. In either case, BAP complexity turns out to be NP-hard [11]. In [12] has been proposed a dynamic formulation of the BAP or Dynamic Berth Allocation Problem (DBAP), which is represented as a finite set of docking berths and the arrival of the ships is considered random. In other words, the spatial dimension of the ships and berths is not considered. This formulation is called "dynamic" compared to the previous one called Static Berth Allocation Problem (SBAP) [13] which considers that all the ships are already in port when the berths are available. The SBAP can be solved in polynomial time with the Hungarian method proposed in [14], since it can be reduced to an assignment problem. For the DBAP, the authors take advantage of this feature and propose a Lagrange relaxation for reaching a desirable allocation sub problem. The computational results show that DBAP is relatively easy to solve while the cases are "close" to the static case, in the sense that most of the ships are already in port when the berths are becoming available. The objective function is the sum of service times of the ships and do not consider the existence of certain vessels priority over others. On the other hand, in literature there is no much research covering the BAP from a software architecture perspective, nor using agent technology as modeling paradigm. In the following are mentioned the two most relevant works in the area for the present research. The first is MADARP [6], an agent software architecture for the Dial-a-Ride Problem (DARP), because BAP can be seen as a Passenger Transportation Problem, in which ships correspond to passengers and the berths to the vehicles. MADARP architecture identifies different layers: interface, planning, service and ontology and agents associated with each of the actors involved: customers and vehicles. In [9] the work was further advanced and a novel solver for DARP was leveraged based on Genetic Algorithms. A second alternative found tackling partially BAP is [7], which proposes a multiagent architecture as solution to a port terminal operations automation problem. The system architecture divides the problem into sub-problems, which are determined by specific agents. However, the above solutions do not provide enough details on the solvers used and on how the resulting planning & control support systems aid port decision-makers to implement diverse planning policies according to their needs.

4 The MABAP Planning System

This chapter presents the functional specification of the multiagent decision support system and its technical design, in terms of software engineering artifacts. Regarding the BAP, some of the specifications for the requirements and restrictions considered for the problem are described in the following:

- It is considered the discrete case of the problem, in which the docking points are considered fixed in size and number.
- Each ship must have its ETA (Estimated Time of Arrival), the estimated processing time and allowed time windows for its processing.
- The berthing points have time windows that limit their availability.
- The implementation must consider the existence of dynamism, due to events that may occur, such as a ship's cancellation of arrival, delays and/or closure of any berth.
- The length of the ship must not exceed the length of the largest section in the dock.
- A section can only process the ships one by one, and as we know they should not overlap in the time-space diagram.

Proposed Multiagent Architecture

Figure 2 shows the proposed conceptual architecture, based on the MADARP architecture [6]. Therefore, the paper considers making an adaptation of this architecture to the particular problem of BAP maintaining its structure, leading agents to have similar functions, i.e. the vessels are considered as "clients" that request a service, not of transportation in this case but of a docking point assignment. On the other hand, berthing points or its subsections are considered as "vehicles" that offer a service, not of transportation but of vessels stowage. The following Figure 2 depicts the main agents of the MABAP architecture. In the first layer, Berth agents and Ship agents leverage as interface agents, plus the BerthPlanner and BerthRequest agents tackling the planning & scheduling tasks.

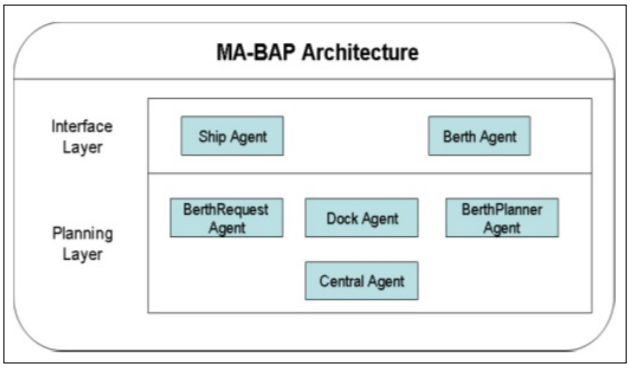


Figure 2: Multiagent Architecture for BAP.

There is also another agent called Dock agent, representing the management of the port company to act as intermediary between Berths and Ships. The Central Agent is a matchmaker, responsible for delivering the list of Berth agents that meet the requirements of a given Ship agent. The agents participating in the system and their main responsibilities are described in the following:

- Ship Agent: It is the one who communicates with the operator and takes the tasks of ship registering and the reporting of events that take place with the ship. It also initiates the docking process with a request, being entitled to cancel and change such request. The created requests are sent to the BerthRequest agent.
- BerthRequest Agent: This agent represents each ship in the negotiation process when requesting a new berthing point. It is responsible for receiving orders to process a new request and terminate its execution. Together with the Ship agent, they embody each of the vessels requiring stowage.
- Berth Agent: Represents each docking point or berth in the system, creating its corresponding BerthPlanner agent. It also manages the berth registration with the Central agent.
- BerthPlanner Agent: It corresponds to the planner agent for each docking point or berth. It keeps track of the ships stowage sequence and it is involved in the process of allocation of new docking points, evaluating the diverse berth requests from ships as they arrive. It also provides its actual schedule to other agents in the society upon request.
- Central Agent: Together with the Dock agent, this agent embodies the third main actor, the dock or pier. It keeps up a registry of the berths available at the dock and is responsible for providing the list of berths that happen to be candidates for a particular request coming from a ship.
- Dock Agent: This agent is devoted to manage the process of allocation of berthing points, handling incoming requests and initiating negotiations with the various berths of the dock.

The main interaction scenario (Figure 3) is when a ship calls for the allocation of a new docking point. The sequence starts with the ship which sends a docking request through the BerthRequest agent to the Dock agent. Such request has a profile, wherein the information includes the ship's name, its approximate time of processing and estimated dates (time) of arrival and departure.

The Dock agent sends the request to the Central agent, requesting a list of candidate sections that meet the profile of the request. The latter generates the list and sends it back to the Dock agent. The Dock initiates a contract-net [4] with the BerthPlanner agents; therefore, it sends a request for berthing to all the BerthPlanner agents from each of the sections in the matching list. These latter agents perform a greedy algorithm for evaluating the insertion of the new ship in its space-time diagram and send back a proposal profile.

From the arrived proposals, the Dock agent selects the best alternative according to its objective function (e.g. minimum total time of stay of the ship in port). Then, the agent communicates to the selected BerthPlanner agent to insert the ship in its sequence and informs the other BerthPlanner agents the rejection of its proposals. Finally reports the ship agent on the result of its request.

5 Insertion Algorithm

Within the MABAP system, the evaluation of the inclusion of a new ship into a berth sequence and to choose within the proposals from each berth which is most suitable, are the fundamental objectives of the problem and, therefore, a key aspect of the application implementation. The algorithm used in the application is based on the insertion heuristic proposed by Jaw et al. [15] for the DARP problem, and used on a past research [9] which has been adapted to fit the BAP

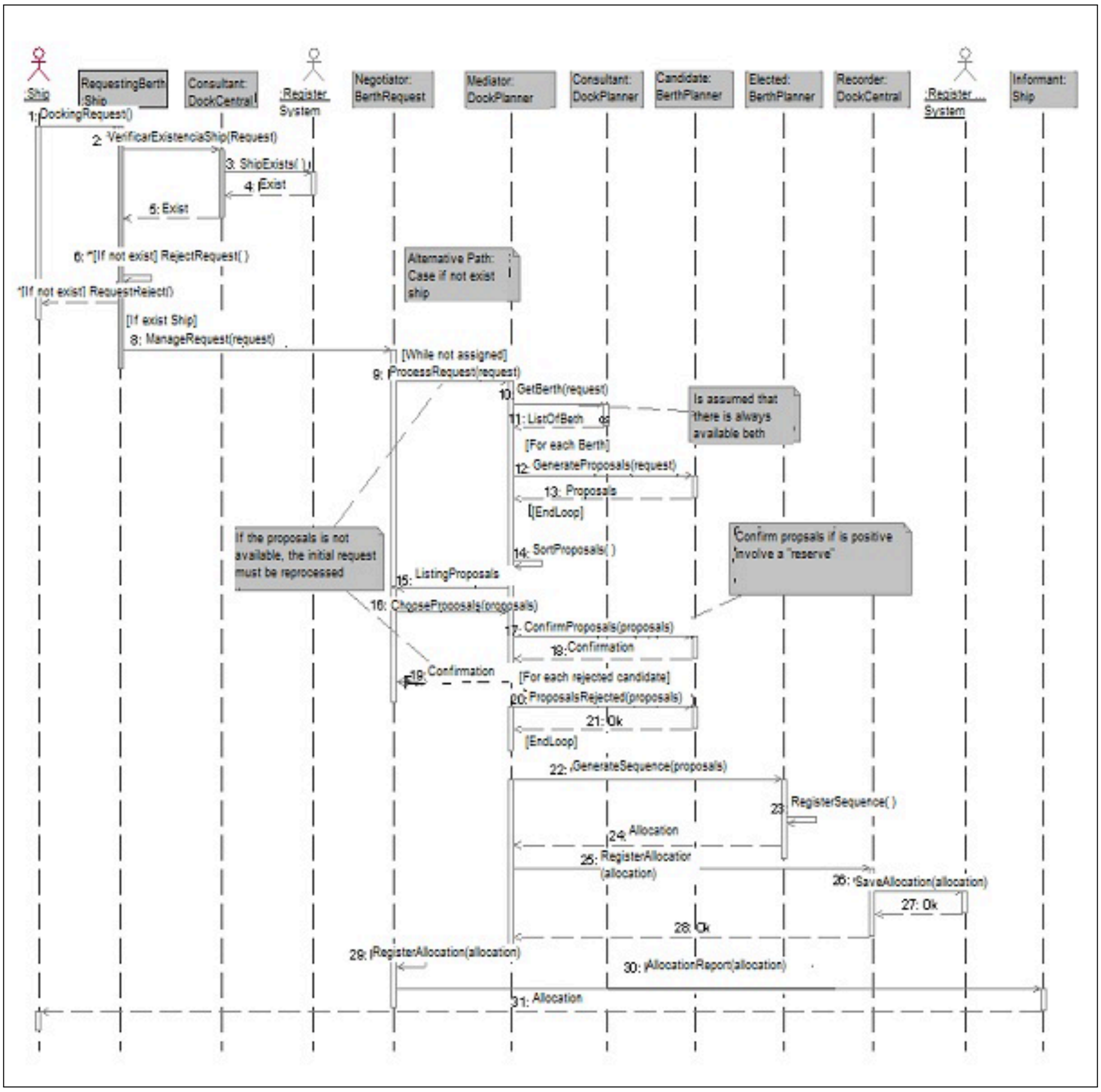
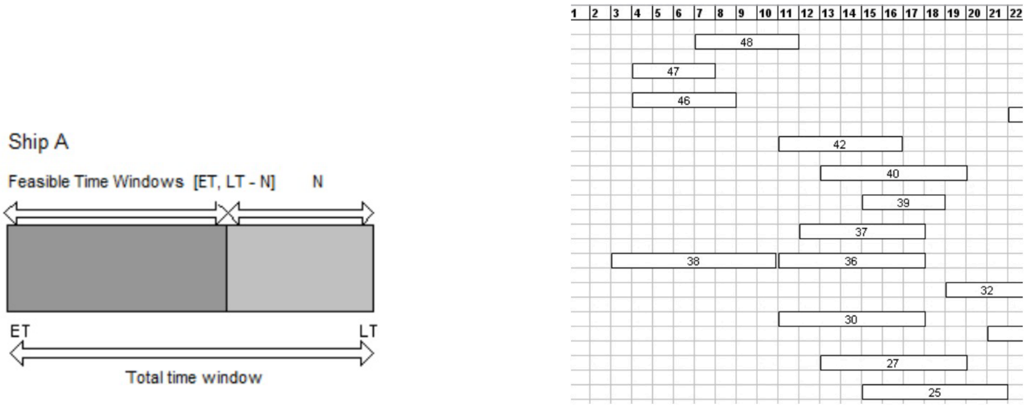


Figure 3: Role Identification Diagram for Berth request scenario.

problem. These changes consider using a single event that is associated with the ship service at the dock, which involved working with a single time window. In the case of DARP are considered two events; one for the pickup of the passenger and another for his delivery. The creation of the initial time window for each ship is related to the processing time associated with it, as shown in Figure 4a. For the ship A, with estimated arrival and departure times of ET and LT respectively, and processing time N, its maximum time window corresponds to [ET, LT] and the feasible initial time to [ET, LT-N]. The idle time or "slack" in DARP is associated with the time in which the vehicle is idle; either stopped or traveling without having to go to pickup or deliver a passenger. In the case of BAP, it is determined by the difference of the time the berth is ready to receive a new ship and the time of the ship actual arrival to the berth.



(a) Time Windows.

(b) Input file example with distribution of ships arrival.

Figure 4: Initial parameters of MABAP.

Therefore, unless the port has a very high ship arrival rate, there will always be idle times at the berthing points of the port. Hence, it was decided to remove the restriction of the original heuristic as to allow slack times between ship arrivals, providing a more realistic scenario. Furthermore, slack times will also allow discriminating better solutions among the proposals provided by each of the berths.

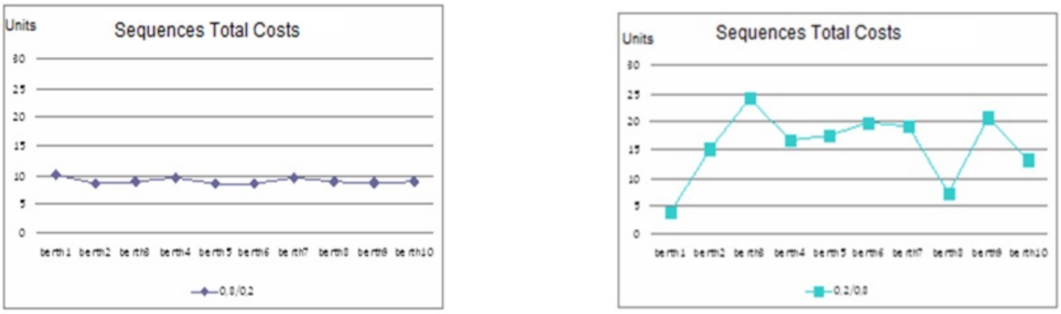
6 Tests and Results

This section presents the results obtained from the tests. To gather data a typical scenario that considers a dock with 10 berthing points has been considered together with the arrival of 50 ships within a time horizon of 60 time units. Each docking point takes 1 time unit to be ready to serve the arrival of a new ship (getReadyTime). No space restrictions are considered as any docking place can attend any ship.

The generation of instances was performed by an automated process, generating ships data with random time windows of 5 time units maximum. Figure 4b shows a graphical representation of the ships arrival within the time horizon for a sample input file. Each box represents a ship and its time windows width. The objective function considered the weighted sum of two factors: on one hand, the number of processed ships and on the other hand, the idle time (slack) of berths between ships services.

Results

The tests were carried out under two scenarios for planning the distribution of ships on berths, which considered different values for weighting the two factors of the objective function; the ShipsNumberFactor and the SlackFactor. A first scenario had weights of 80% - 20% and a second scenario had weights of 20% - 80% respectively. To obtain the results on each scenario, 10 input files were used and 20 runs were made under each scenario for a total of 200 executions. The graphics show the total estimated cost of the ten sequences under the two scenarios. By starting with the scene of strong restriction on the number of ships (Figure 5), the total cost is in principle almost a straight line and keeps controlled equally for each docking point.



(a) Graphic of total costs for scene 80%-20%.

(b) Graphic of total costs for scene 20%-80%.

Figure 5: Graphical results.

Discussion

The following graphics show how the multiagent system can be tuned according to the policies of the port decision-maker. In a first case, the user can prefer solutions in which a higher number of ships are served per berth (80%-20%) while in another case the user may prefer to minimize the idle time of berths (20%-80%). Of course the proposed solution can consider other performance measures either for the operator (e.g maximizing throughput) or for the ships it serves (e.g. minimizing ships' stay).

Other aspect to analyze refers to the distributed nature of the planning. In this case the planning process involves two steps: 1) the assignment of ships to available berths and 2) the scheduling of the assigned ships inside each berth. Our actual prototype of support system provides a distributed solution by using the Contact-Net Protocol [4] as coordination/connection mechanism. Other protocols can be used, such as Dutch or English auctions, however in a general case no significant improvement should be obtained as in such cases there is no bids selection process. As early described, for each berth request coming for a ship the Dock agent makes a call-for-bids to the available BerthPlanner agents in charge of managing the ships' schedule of each berth. The underlying optimization problem is known to be NP-Hard, hence the actual prototype uses a greedy insertion heuristic which ensures less than a second per request. Such solver used inside the BerthPlanners can be changed according to the given needs (by incorporating soft computing techniques) for best solutions alternatives at a cost of more processing time. Finally, from a more political perspective, the ship arrival into a port and its berth allocation triggers a transversal process that involves diverse entities and decision-makers (port operator, port authority, berth operators, etc.). Therefore, a centralized system is unrealistic. The multiagent system for berth allocation enables such integration of the different actors involved in the berth planning process. In addition, the resulting software architecture is flexible enough to permit the inclusion of other actors (e.g. Customs Authority, customs brokers, crane & floor/yard operators, etc.) in a transparent way. In this sense, an interesting approach would be to make it an open agent system by incorporating what developed in [17]. The integration of the diverse port enterprises through a multiagent system capable of wrapping the diverse enterprises' systems, leverages as a feasible alternative in port operations in general and in the berth planning process in particular.

7 Conclusion

This paper has tackled one of the many problems that can be found in the daily operation of a port, the berth allocation problem (BAP). The complexity of the problem, its mathematical modeling and its variants have been reviewed. A multiagent system has been presented which solves the underlying optimization problem by making use of a greedy insertion heuristic. Good results were obtained, allowing the decision-maker to choose solutions according to the port authority policies.

Further research considers refining the events to be managed by the system (e.g. delays, cancels, breakdowns), introducing multi-objective optimization, thus providing a curve of solutions, and the incorporation of a 3D graphical interface to show the planned solution.

Acknowledgments

This work has been partially funded the Pontifical Catholic University of Valparaíso (www.pucv.cl), through Regular Project No. 037.425

Bibliography

[1] United Nations Conference on Trade And Development (UNCTAD), *Review Maritime Transport*, Cap V, pp. 73-74, Cap VII. 2005.

[2] Wooldridge, M., *An Introduction to Multiagent Systems*, Dept. of Electronic Engineering, Queen Mary & Westfield College, 2002.

[3] Burrafato, P., Cossentino, M. Designing a Multi-Agent Solution for a Bookstore With the PASSI Methodology, In *Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002)*, 2002, pp. 27-28.

[4] FIPA, Contract Net Interaction Protocol Specification. Available at: www.fipa.org/specs/fipa00029/SC00029H.pdf

[5] Daganzo, C., The Crane Scheduling Problem, *Transportation Research B 23B*, 1989, 159-175.

[6] Cubillos, C., Crawford, D., Rodríguez, N., MADARP: A Distributed Agent-based System for On-Line DARP. In: *I. Stojmenovic et al. (Eds.): ISPA 2007, Springer Heidelberg LNCS*, Vol. 4742, 160-169, 2007.

[7] Botti, V.J. Multi-Agent System Technology in a Port Container Terminal Automation, *European Research Consortium for Informatics and Mathematics News*, Vol. January, No 56, 37-39, 2004.

[8] Nwana, H.S., Ndumu, D.T., *An Introduction to Agent Technology*, Re-Drawn by Mobile Computing, Dept. of IECS, Feng Chua University, R.O.C., 2003.

[9] Cubillos, C., Urra, E., Rodríguez, N., Application of Genetic Algorithms for the DARPTW Problem, *INT J COMPUT COMMUN* , ISSN 1841-9836, 4(2):127-136, 2009.

[10] Pinedo, M. Scheduling: *Theory, Algorithms and Systems*, Prentice-Hall, Englewood Cliffs, NJ. 1995.

[11] Garey, M. R., Johnson, D. S., *Computers and Intractability: A guide to the Theory of NP-Completeness*, Freeman, San Francisco. 1979.

[12] Imai, A., Nishimura, F., Papadimitriou, S., The dynamic berth allocation problem for a container port, *Transportation Research 35B*, 401-417. 2001.

[13] Imai, A., Nagaiwa, K. Chan, W. T., Efficient planning of berth allocation for contenedor terminals in Asia, *Journal of Advanced Transportation*, 31, 75-94. 1997.

[14] Papadimitriou, C. H., Steiglitz, K., *Combinatorial Optimization; Algorithms and Complexity*, Prentice-Hall, Englewood Cliff, NJ. 1982.

[15] Jaw, J. Odoni, A. R. Psaraftis, H. N. Wilson, N.M.H. A heuristic algorithm for the Multi-Vehicle Advance-Request Dial-a-Ride Problem with Time Windows, *Transportation Research B*, 20B(2): 243 - 257, 1986.

[16] Cabrera-Paniagua, D., Herrera, G., Cubillos, C., Donoso, M. Towards a Model for Dynamic Formation and Operation of Virtual Organizations for Transportation, *Studies in Informatics and Control*, ISSN 1220-1766, 20 (3): 255-264, 2011.

[17] Cubillos, C., Donoso M., Rodríguez N., Guidi-Polanco F., Cabrera-Paniagua D., Towards Open Agent Systems Through Dynamic Incorporation, *INT J COMPUT COMMUN*, ISSN 1841-9836, 5(5):675-683, 2010.