Model and Heuristic for Berth Allocation in Tidal Bulk Ports with Stock Level Constraints

Victor Hugo Barros, Tarcísio Souza Costa, Alexandre C.M. Oliveira, Luiz A.N. Lorena

PII: S0360-8352(11)00006-4
DOI: 10.1016/j.cie.2010.12.018
Reference: CAIE 2890

To appear in: Computers & Industrial Engineering

Received Date: 26 May 2009
Revised Date: 24 September 2009
Accepted Date: 27 December 2010


This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
Model and Heuristic for Berth Allocation in Tidal Bulk Ports with Stock Level Constraints

Victor Hugo Barros\textsuperscript{a} Tarcísio Souza Costa\textsuperscript{b} Alexandre C. M. Oliveira\textsuperscript{c} Luiz A. N. Lorena\textsuperscript{d}

\textsuperscript{a}Universidade Federal do Maranhão - UFMA, São Luís MA, Brasil
bs.victorhugo@gmail.com

\textsuperscript{b}Universidade Federal do Maranhão - UFMA, São Luís MA, Brasil
priestcp@gmail.com

\textsuperscript{c}Universidade Federal do Maranhão - UFMA, Departamento de Informática, São Luís MA, Brasil
acmo@deinf.ufma.br

\textsuperscript{d}Instituto Nacional de Pesquisas Espaciais - INPE, Laboratório Associado de Computação e Matemática Aplicada, São José dos Campos SP, Brasil
lorena@lac.inpe.br

Abstract

We consider the problem of allocating berth positions for vessels in tidal bulk port terminals. A berth is defined as a specific location alongside a quay where a ship loader is available for loading or unloading vessels, accommodating only one vessel at a time. In tidal ports, draft conditions depend on high tide conditions, since available depth at low tide is not adequate for the movement of ships. Some port terminals are associated with important transnational enterprises which maintain strong control over the stock level of their goods. Since the stock level sometimes depends on a continuous process of consumption or production of minerals, the decision to load or unload vessels must consider the amount of the bulk cargo stored in the port yards. Therefore, a basic criterion for decision making is to give priority to the vessels related to the most critical mineral stock level. A second basic criterion is to decide what sequence of vessels reduces the overall demurrage within a given planning horizon. This paper presents an integer linear programming model based on the transportation problem to represent the Berth Allocation Problem in Tidal Bulk ports with Stock level conditions (BAPTBS). Problem instances are solved by a commercial solver and by a Simulated Annealing-based algorithm (SA). The SA employs a problem-specific heuristic, becoming a valid alternative for finding out good solutions for difficult instances.
1 Introduction

The importance of the port grows with increasing progress in the technology of construction of big ships and improved trade between nations resulting from globalization. In Brazil, ports that handle dry bulk cargo, called bulk ports, are responsible for the outgoing of great part of agricultural exports, including coffee and soy, minerals such as iron ore, and imports of petrol derivatives. Bulk ports play an important role in the Brazilian economy and its greater efficiency is being sought through administrative and operational efforts.

In port planning task, the operational teams work with four concepts: Expected Time of Arrival (ETA), Expected Time of Berthing (ETB), Expected Time of Completion (ETC) and Expected Time of Sailing (ETS). ETA is previously known, but the others depend on certain operational conditions, like tidal conditions, berth availability, handling time and the relative importance of the vessel. Vessel priority is strongly associated with logistic issues in trading management and industrial plant workflow.

In this paper, the problem of allocating berth positions for vessels in tidal bulk port terminals is considered. A berth is a quay location, equipped with one or more ship loaders. It is usual to say that a berth may accommodate one vessel at a time. In tidal ports, even when a berth position is available, vessels may need to wait for mooring. At low tide, available depth in such ports is not adequate for the movement of ships. The transit from waiting areas to the berth position is done in time windows during high tide which happens at 12-hour intervals, approximately.

Bulk ports essentially operate with bulk cargo. Bulk cargo is transported unpackaged in large quantities, classified as liquid (e.g., petroleum, gasoline, caustic soda and chemicals) or dry (e.g., coal, grain, iron ore and bauxite ore). Vessels are loaded using either excavators and conveyor belts or pipelines. Silos or stockpiles for the bulk cargo are often alongside the berth.

This research is motivated by a problem at the maritime industrial port complex of São Luís, formed by Itaquí port and the private terminals of Ponta da Madeira (Vale Mining Company) and Aluminum Consortium of Maranhão (Alumar Aluminum refinery). The private terminals of Vale and Alumar, large transnational enterprises, maintain a strong control over the raw material stock level. The three port terminals are together responsible for the second largest grain cargo handling in Brazil (Vale, 2008).
The mining company moves cargo along 9,820 km of its railroad network to its 6 port terminals. About 81.7 million tonnes of cargo are loaded through the Ponta da Madeira Complex (based on 2006 reports), most of which are minerals as iron ore and pellets (Vale, 2008). Furthermore, Vale also loads more than 60 different types of products, such as steel, coal, limestone, granite, containers, pig iron, agricultural products, wood, pulp, vehicles and various cargoes. For bulk operations, stock level must be sufficient for loading onto the ships.

The aluminum refinery, in turn, controls the stock level by avoiding production collapse caused by lack of some raw material, such as coke, caustic soda, petroleum pitch and bauxite ore. A raw material with stock below a certain critical level gains high priority for unloading. The stock level is measured in terms of days-on-hand (DOH).

The relative importance of the vessels is also influenced by contractual costs. Trips are associated with transport contracts that define a set of rules and responsibilities for both the contractor and the owner or shipper. Through such contracts, vessels must arrive in pre-established lay-day windows, defined by the logistic enterprise team, considering mineral consumption or production. The lay-day definition is a sufficiently hard problem, solved at a strategic level before an operational one.

The contractual cost is commonly incurred due to delayed ship operation caused by events under contractor responsibility, like unavailability of shiploaders or even management decisions. When vessels at a port are not loaded on time, a charge, called demurrage, is levied against the ship-contractor. If a vessel is loaded ahead of schedule the ship-contractor receives a credit called dispatch. Demurrage or dispatch at the discharge (or charge) port can be estimated during allocation planning.

The problem of allocating berth positions for vessels in tidal bulk terminals considers some conditions as hard constraints and others as penalty costs to be minimized. The problem is to determine the berths so as to minimize the total demurrage incurred given the tidal conditions and the stock level constraints. Berth constraints may not be considered, i.e., berth positions are similarly equipped (homogeneous berth positions) and they may load or unload any bulk cargo. The term bulk cargo is employed in this work to represent any kind of ingoing or outgoing good. Figure 1 illustrates the operational scenario in a tidal bulk port with homogeneous berth position.

**Fig. 1 - Operational scenario in a tidal bulk port with homogeneous berth position**

Several berth allocation models have been proposed in the literature. They differ in terms of the assumptions being made, such as whether the vessel
waiting is allowed (Guan et al., 2002, Imai et al., 2001, Kim and Moon, 2003), whether multiple mooring vessels at a berth is possible, whether vessel arrival times are considered, whether the processing times are proportional to vessel size (Lim, 1998), whether the berth positions are continuous or discrete (Cordeau et al., 2005), and so on. Vessels do not need to wait when berth space is abundant, when parallel mooring is possible, or there are not any tidal conditions to be considered.

A port can operate in an efficient way and the allocation and programming of ships to berths can have a primary impact on the efficiency of these operations (Hansen et al., 2007). Other decision problems can appear in a container port (Vis and Koster, 2003). Despite relatively ample literature about decision making in container ports, very little space has been devoted to grain or bulk cargo ports (Günther and Kim, 2005).

The Berth Allocation Problem (BAP) is typically defined as optimally scheduling ships to berthing areas along a quay. The objective, from that focus point, is the minimization of the total (weighted) service time for all ships, defined as the time elapsed between the ETA and ETC (Cordeau et al., 2005). The planning horizon of the BAP in general is one week, and the berthing plan is updated every day. This paper is devoted to the formulation of a particular BAP case in which certain assumptions are made in a tidal bulk port environment. A problem instance considers waiting and incoming vessels to be moored (Imai et al., 2001) on a planning horizon.

The main assumptions made in this paper relate to tidal conditions and stock level, commonly observed in the maritime industrial port complex located in São Luís. The waiting and incoming vessels are allocated to time windows that have favorable tidal conditions and berth availability. Depending on a vessel’s draft (the depth of water from the waterline to the bottom of the ship) and the tidal conditions, a sailing and a berthing vessel may share the same tidal time window (simultaneous operation).

A basic criterion for the decision of which vessels must be allocated to a specific time window is, firstly, the stock level of mineral (or raw material) concerning such a vessel. For loading operations, stock must be enough to be loaded onto the vessel. Unloading operations give priority to those vessels carrying raw materials at critical stock level. A second basic criterion for deciding the vessels priority is settled in demurrage rates, attempting to reduce the overall demurrage on the planning horizon.

This paper proposes an Integer Linear Programming model to represent the Berth Allocation Problem in Tidal Bulk ports with Stock level conditions (BAPTBS). Problem instances are defined and solved by a commercial solver and by a simulated annealing-based heuristic. The remainder of this paper
is organized as follows. In Section 2, a brief survey about BAP is presented. The mathematical modeling is proposed in Section 3. In Section 4, the computational results are examined and discussed as well as issues concerning approximate algorithms. The conclusions are summarized in Section 5.

2 Related works

The objective of the BAP is usually to minimize the total service time of all ships. The distinction among a vessel’s importance may be made by employing a pricing scheme based on demurrage over delayed operations. Penalty terms may be explicitly included in the objective function considering demurrage costs when the vessel service time exceeds the contracted value (Imai et al., 2006).

In the discrete case, the BAP can be modeled as an unrelated parallel machine scheduling problem where ships are treated as jobs and berths as machines. In the continuous case, the BAP is a cutting stock problem with additional constraints, placing non-overlapping ships in a two-dimensional plan in time and continuous positions (Pinedo, 1995).

In (Lim, 1998), the quay is represented as a continuous line and the problem is heuristically solved by deciding the allocation of berthing points to ship berthing times, also assuming constant handling times. That approach does not solve the general problem in which the berthing time is a decision variable and the handling time varies along the quay.

In (Nishimura and Imai, 2001), a non-linear integer programming model and genetic algorithm are presented. The metaheuristic is based on a different representation of the spatial dimension in which the quay is a collection of segments, allowing up to two ships sharing the same segment at the same time if their lengths are berth-segment compatible. Additional constraints regarding berth water depth are also introduced.

In (Park and Kim, 2003), another non-linear integer programming model considering quay crane assignments (QCAP) has been introduced. The main assumption that allows integrating the BAP and the QCAP is that handling times vary linearly with the number of quay cranes assigned to a vessel. The authors recognize that this is an approximation of reality. The objective function is to minimize the sum of penalty terms over all vessels, assuming that an optimal berthing point is known and a penalty may be applied whenever a different choice is made. The algorithm employs Lagrangean relaxation and a subgradient optimization method.
In (Kim and Moon, 2003), a mixed integer linear programming model for the continuous case was formulated. A commercial solver is able to find the optimal solution for instances involving seven ships on a 3-day planning horizon. A heuristic based on simulated annealing is proposed to solve instances of realistic dimensions.

More recently, in (Lorenzoni et al., 2006), the problem of attending ships within agreed time limits at a port under first come first served conditions has been formulated. In addition, the use of the developed tool based on a mathematical model of a multi-mode resource-constrained scheduling problem is indicated, with an extension of a differential evolution algorithm. The computational tests with data generated on the basis of the characteristics of a real port environment are also presented. The authors make an allusion to the tidal conditions in a port, which can restrict the port entrance to vessels at certain time intervals.

3 Problem modeling

In this work, the Berth Allocation Problem in Tidal Bulk ports with Stock constraints (BAPTBS) is modeled in discrete form as a transportation problem in which $N$ ships are seen as suppliers and $M$ favorable tidal condition windows (henceforward named as tidal time window - TTW) as consumers. Each ship $i$ must be allocated to a subset of TTW whose length corresponds to the handling time $h_i$ necessary for operation completion. For convenience, since such TTWs happen in a regular and known time frequency $TF$ ($TF \approx 12$ hours), the planning horizon of the BAPTBS is divided into $M$ TTWs. For this reason, the continuous time scale is changed into a discrete tidal scale, in a way that it is easy to compute the operation real time by calculating $T = TF \times j$, for a specific $j^{th}$ TTW.

The decision variable $x_{ij}$ is given by:

$$
\begin{cases}
1 & \text{if ship } i \text{ is allocated to TTW } j \\
0 & \text{otherwise}
\end{cases}
$$

Figure 2 illustrates the BAPTBS modeled as Transportation problem.

Fig. 2 - BAPTBS modeled as transportation problem
3.1 Input data

The data for BAPTBS is given by:

- \( N \): set of ships, \( n = |N| \);
- \( M \): set of TTWs, \( m = |M| \);
- \( L \): set of berth positions, \( l = |L| \);
- \( a_i \): arrival TTW of ship \( i \);
- \( h_i \): handling TTW of ship \( i \);
- \( t_i \): amount of TTWs contractually allowed for ship \( i \);
- \( d_i \): demurrage or dispatch for ship \( i \);
- \( e_k \): the initial stock level for bulk cargo \( k \);
- \( w_k \): the amount of consumption or production for bulk cargo \( k \);
- \( q_{ik} \): the cargo capacity of ship \( i \) with respect to bulk cargo \( k \).

Bulk cargo \( k \) may be imported or exported (ingoing or outgoing good). Hence, the operation type (unload or load) is defined by the signal of \( w_k \) and \( q_{ik} \):

- importation: \( w_k, q_{ik} > 0 \);
- exportation: \( w_k, q_{ik} < 0 \);

3.2 Objective function

The objective function is calculated by the cost accrued over all operations on a given planning horizon:

\[
\text{cost}_i = (s_i - t_i) \times d_i
\]

(2)

where \( s_i \) is the service time given by:

\[
s_i = c_i - a_i
\]

(3)

and the completion time, \( c_i \), is given by:

\[
c_i = b_i + h_i
\]

(4)

The berthing time, \( b_i \), is obtained via the decision variable, \( x_{i,j} \). The cost function of ship \( i \) becomes:

\[
\text{cost}_i = (b_i + h_i - a_i - t_i) \times d_i
\]

(5)
The objective function in Eq. 6 represents the overall cost, taking all $|N|$ ships on the planning horizon of $|M|$ TTWs. One can observe that the overall cost is expressed in the $Ax + C$ form, where the $C$ term is necessary for obtaining the real cost value, but not for the decision process. The $A$ term works as a median TTW of berthing, $\frac{j_{\text{med}}}{M}$, weighting the ship priority, $d_i$.

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \frac{d_i}{h_i} x_{ij} + \sum_{i=1}^{N} d_i \left( \frac{1}{2} h_i - a_i - t_i + \frac{1}{2} \right)$$  \hspace{1cm} (6)$$

3.3 Constraints

The proposed model considers hard constraints expressed in equations 7 through 11. Eqs. 7 and 8 imposes that ships shall berth after their arrival TTWs, $a_i$, and keep unloading (or loading) during $h_i$ TTWs to complete the operation. The number of ships allocated to a given TTWs may not exceed the total number of berth positions, $|L|$ (Eq. 9).

$$\forall i \in N: \sum_{j=1}^{a_i-1} x_{ij} = 0$$  \hspace{1cm} (7)$$

$$\forall i \in N: \sum_{j=a_i}^{M} x_{ij} = h_i$$  \hspace{1cm} (8)$$

$$\forall j \in M: \sum_{i=1}^{N} x_{ij} \leq |L|$$  \hspace{1cm} (9)$$

Eq. 10 prevents the interruption of a low demurrage ship operation for berthing another one sufficiently more expensive. In real port operations, service interruptions are undesirable, but they can occur due to many factors like equipment malfunction, loss, or damage. From the planner’s point of view, service interruptions are unforeseeable and, once started, the berth operation must not be interrupted for any reason, including advantages in operational final cost. Eq. 10 searches for no consecutive 1’s in the decision ship bit stream, which would mean an undesirable operation interruption. If they were found such a solution would be considered infeasible.

$$\forall i \in N, \forall j > 1 \in M: \sum_{l=1}^{j-1} x_{il} - j x_{ij-1} + j x_{ij} \leq j$$  \hspace{1cm} (10)$$

Eq. 11, finally, considers the stock level conditions, obliging ship allocations in which any bulk cargo storage level drops below zero. In industrial refineries,
safety storage level is used as threshold for which emergency actions must be triggered immediately. For simplicity, zero stock level is used in this work for discarding infeasible berth allocation.

\[ \forall j \in M, \forall k \in K : \sum_{i=1}^{N} \sum_{l=1}^{j} \frac{q_{ik}}{h_i} x_{il} \geq jw_k - e_k \]  

(11)

4 Computational experiments

Several problem instances, corresponding to real scenarios, were randomly generated in order to validate the proposed model. Some are considered hard instances because they can take a long time to be solved by a branch-and-bound like method. It is not possible to exactly determine when an instance is hard, but there are two points that play an important role in this question. First, obviously, it is related to the number of vessels, \( N \), and, consequently, the number of tides, \( M \), on the planning horizon. The more ships and tides more decisions should be taken, more variables and constraints as well. The second point is that as high as the number of berths more options of berthing exist and, therefore, less decisions need to be made.

In Table 1, the problem instances used in this work are presented. The instance features (number of ships, berths and minimal TTWs needed to solve it) as well as the solver time (in seconds) and global optimal found are shown (Eq.6). The optimal values were obtained by CPLEX, a high performance Linear Programming (LP) and Mixed Integer Programming (MIP) solver from ILOG (CPLEX,2006).

The found solutions are all optimal, including the hardest one: the 20-ship instance 14 for which CPLEX has taken execution over time (above 13 hours or 48 604.41 seconds). Concerning the time issues, the experiments were conducted in a Pentium D (2.80GHz), 1GB RAM. CPLEX has taken over 3 hours to obtain 7 671.0 in 15-ships instance 6, for example, and less than 1 hour to reach the optimal value in 30-ship instance 18. Despite the number of ships, the instance 6 is definitely more difficult to solved with CPLEX than instance 18.

Table 1 - Problem instance features and optimal values

As shown in Table 1, the execution times tend to increase with the number of ships. However, sometimes, instances of 10 vessels (less complex) were harder than instances of 15 and even 20 ships, with same number of berths. Finally, the more berths the instance has, the faster the solution is found. Some instances were created with more berths to obtain better times. With respect
to the TTWs, minimizing the cost, the solver is able to find the exact number of TTWs needed for berthing all ships on the planning horizon.

4.1 Model validation

Some special experiments are now presented for further remarks. In Figure 3, the ship berthing sequence for the instance 17 is shown. With 3 berths, this instance takes only about 5 seconds to solve. The allocation of ships to berths is denoted by a number-symbol association. The ship sequence for berth 1, for example, is formed by ships which have numbers in circles, following the specific dashed line: \{1, 4, 7, 8, 15, 13, ..., 28, 30\}.

Fig. 3 - Berthing decision in instance 17.

For example, the stock level behavior through the planning horizon 11 is depicted in Figure 4. The stock level of 4 bulk cargoes is always positive when constraints of Eq. 11 are applied (Figure 4a). Without stock level constraints, the bulk cargo 4 assumes negative values after TTW 51 which represents a stock underflow (Figure 4b).

Fig. 4 - Stock level on the planning horizon for instance 11: a) with stock level constrained b) without stock level constraints

The instance 11 is interesting in this point of the paper because both figures are quite similar. Examining the optimal solution for both cases, the following vessel sequence can be extracted:

(1) 1 5 4 12 7 16 17 14 8 13 15 18 9 11 [20 3] 10 19 [6 2] (applying Eq. 11)
(2) 1 5 4 12 7 16 17 14 8 13 15 18 9 11 [3 20] 10 19 [2 6]

There are two differences in the sequence: exchanges between vessels 20 and 3 and vessels 6 and 2. The first exchange does not affect the stock level nor the remaining subsequence: following vessel 10 and 19 are berthed at same TTW (44 and 47 respectively) in both sequences. Thus, this exchange is not relevant for the objective function value. However, the second exchange causes a little difference in feasibility and in the optimal cost. Since Eq. 11 is applied, the vessel 6 is advanced for keeping the stock level positive which does not happen in Figure 4b. However, such a necessary decision increases the operational cost, leading the optimal objective function value to 14 671.0 against 14 663.0 if stock level constraints were not applied.
4.2 Simulated Annealing-based heuristic

The average time for obtaining the optimal solution for the instances of Table 1 is about 1 hour. However, the hardest operational scenarios can lead to running times of about 8 hours in a high performance commercial solver. Hence, in this section, an approximate algorithm is presented as a valid alternative for finding out good solutions for hard instances. A Simulated annealing (SA) algorithm is held for fast solving instances due to its simplicity, robustness and ease of further commercial implementation.

4.2.1 Foundations

Simulated annealing (SA) (Kirkpatrick, 1983) is an approximate algorithm based on the physical phenomenon of annealing of solids. This phenomenon is responsible for obtaining perfect states of a solid with low energy by warming the solid up until the fusion state is reached, followed by careful cooling. The careful cooling allows non-deformed objects to be obtained at the end of the process, gathering solid particles more precisely. A SA simulates this phenomenon fed by the following parameters: a) initial temperature, b) cooling rate and c) number of iterations per temperature level.

The kernel of SA is a decision rule: if a new solution $s'$ is better than the current one, $s$, the decision is to accept $s'$, replacing $s$. Otherwise, the Boltzman machine equation is applied, i.e., $s'$ is accepted depending on two factors: how much worse $s'$ is or how warm the system is. The probability of $s'$ is accepted can be expressed by:

$$p(s') = \frac{1}{1 + e^{(f(s') - f(s)) / T}}$$

(12)

Hence, the algorithm accepts or not a new solution, depending on the temperature level and the difference of objective function value between $s'$ and $s$. The difference between simulated annealing and hill-climbing methods consists of the former allows sidewards and backwards movements (depending on the temperature parameter), while the latter only accepts upward ones (improvement moves).

The simulated annealing algorithm can accept backward moves to avoid getting stuck on a local minimum of functions. But the acceptance of these solutions is made in the majority of cases at high values of temperature, because the degree of agitation of particles (solutions), in this case, is more intense. In case of low temperature, the algorithm is more selective.
4.2.2 High Demurrage Ship heuristic

A solution in the proposed simulated annealing is represented by a sequence of ships that is an order of service, not necessarily the berthing sequence, but the priority order (from high priority, on the left, to low). The first ship $s_1$ is the first to be considered, respecting the arrival time, in the berth programming. Evidently, subsequent ships in the permutation can have a berth conflict with previous ones. Once a berth conflict happens, the less priority ship is delayed, probably, causing demurrage.

The current solution, in general, is the base for generating a new solution in the SA algorithm. This is made through a neighborhood procedure $\mathcal{P}$. The basic idea is to keep some features fixed and to try change the other ones. The modified solution $s'$ is considered a neighbor solution of $s$ by one movement of $\mathcal{P}$. For example, a common procedure for solutions represented by permutation, as $s = \{1, 2, 3, 4, 5\}$, is to exchange two positions in permutation becoming $s' = \{5, 2, 3, 4, 1\}$ (2-Change neighborhood).

In this paper, a specially developed procedure for BAPTBS is proposed: the High Demurrage Ship heuristic (HDSh). HDSh is based on the inherent knowledge about the problem: an ill-positioned ship causes a high demurrage. Following this idea, a non-blind ship exchange should take into account such knowledge about the composition of objective function to try improve solutions.

The HDSh works calculating the costs of each ship in a given solution. The cost $k$ is a contribution the ship $s_k$ gives to the total cost of the objective function and it is given by the previously presented Eq. 5. For example, if a random ship $s_k$ moors exactly in its arrival TTW, its cost is probably dispatch (negative value), i.e., the ship does not wait for a berth when it arrives. The basic idea is to try to improve the solution by exchanging a high-cost ship for a low-cost one in the permutation.

Initially, a random positive-cost ship, $s_h$, is chosen, not necessarily the highest. A second step is to find out another random ship, $s_l$, to which $cost_l < cost_h, \forall l \neq h$. As one can see through the $l < h$ condition, the ship priority is taken into account in this procedure, avoiding delay for an already high-cost ship.

Figure 5 illustrates the High Demurrage Ship heuristic in 3 iterations, considering a 5 ship permutation.

**Fig. 5 - High Demurrage Ship heuristic**

Expertise acknowledge and several computational executions were needed to find a good parameter set for the SA. In these executions, values for initial
temperature, cooling rate and number of solutions generated at each iteration were varied along the range of values commonly used for solving other combinatorial problems. At last, the best parameter set found is: a) initial temperature, 2000, cooling rate, 0.98, number of iterations, 1300. The computational environment for SA is the same for CPLEX.

The computational results can be seen in Table 2.

Table 2 - Simulated Annealing’s results

For each problem instance, SA was executed 20 times. The Optimal solution column recalls the known optimal value obtained by the CPLEX. The $FS \pm \tau^2$ refers to the average and the standard deviation calculated with respect to the best found solutions (FS) encountered in each execution. Over the $FS$, a final solution quality measure, Error%, was taken into account, regarding the error percentage on average for SA. The Hits column shows the number of times that the optimal solutions were reached for each instance. The computational effort of SA is summed up by the number of function calls, FC. With respect to the runtime (RT), CPLEX RT and the average of the SA RT, $\overline{RT}(s)$, are both placed in the last columns of Table 2.

The average of objective function calls and running times were both calculated considering only the well-succeeded executions. For example, in the instance 19, the values 238, 275.4 and 83.3, respectively, were concerned with the 10 executions that had found optimal solutions on average. As one can observe in Table 2, the SA algorithm found the hardest instance’s optimal solution (16, 965.0) (instance 14) in 16 of 20 trials, with a good running time, if compared with CPLEX. On the other hand, in some of the short instances (instances 2 and 3) CPLEX was faster than SA.

The SA performance is considered reasonable both in running time and in quality solutions. Even though in instance 20, for which SA performed poorly, finding the optimal just one time, on average, the final solutions for this instance were close to optimal, giving a low error percentage (0.07%). In fact, the worst error percentage (0.96%) was obtained for instance 19. The SA found optimal solutions in 90% of the trials, running about 30 seconds, on average, and 2.5 minutes for the worst case (instance 20), while CPLEX has peaks of above 13 hours to find the optimal solution (instance 14). For this instance, SA takes about 50 seconds to reach the optimal solution.

The SA’s results, in Table 2, indicate that this heuristic approach is a valid alternative for management (or operational) environment, regarding the best times on average as well as less standard deviation to find good solutions.

In situ experiences, involving a software-based decision making in an important port terminal in São Luís, indicate that scenario simulations can be satis-
factorily examined for planning teams since it can be produced by optimization procedures in up to 5 minutes. For this reason, it is reasonable to conclude that the proposed SA is a valid alternative for solving instances of the BAPTBS here modeled.

5 Conclusion

Brazilian bulk ports are responsible for outgoing goods like coffee, soy, iron ore, and incoming petrol derivatives, for example. Manufactured exportation goods, like aluminum, are also loaded in vessels in private ports of industrial refineries. Bulk ports play an important role in the Brazilian economy and their efficiency is being sought via administrative and operational efforts.

This paper proposes an integer linear programming model to represent the Berth Allocation Problem in Tidal Bulk ports with Stock level conditions (BAPTBS). The main assumptions made in this model are related to tidal conditions and stock level, commonly observed in the maritime industrial port complex located in São Luís.

The waiting and incoming vessels are allocated to tidal time windows (TTW) if berth positions are available. Only similarly equipped berths (homogeneous berths) are represented in this model. The decision of where berthing is not considered important for this approach.

A basic criterion for the decision of which vessel must be allocated to a specific TTW is, firstly, the stock level of grain concerning that vessel. The stock must be kept in safety ranges defined previously by the enterprise logistic team. A second basic criterion for deciding the vessels priority is settled in demurrage rates, attempting to reduce the overall demurrage on the planning horizon.

Computational experiments were performed using a high performance commercial solver over randomly generated problem instances corresponding to real operational scenarios. Preliminary tests indicate that it was difficult to estimate, precisely, the time needed to derivate the better solution using the commercial solver. The average time for obtaining an optimal solution varies from few seconds to 8 hours for instances between 10 and 30 vessels. Some instances were made easier by increasing berth positions but, on average, the commercial solver has taken about 1 hour to find an optimal solution.

A Simulated annealing-based algorithm (SA) was proposed as a valid alternative for finding out good solutions. The SA has solved all the instances generated with reasonable robustness, accuracy and satisfactory runtime. The hardest instance’s optimal solution, for example, was solved in 16 of 20 tri-
als, with a good running time, if compared with the commercial solver. Even though in instances for which SA performed poorly, finding the optimal just one time, on average, the final solutions for this instance were close to optimal, giving an error percentage about 0.96% in the worst case.

The SA was be able to find optimal solutions in 90% of the trials, running about 30 seconds, on average, and 2.5 minutes for the worst case, while commercial solver has reached peaks of above 13 hours to find the optimal solution.

The proposed model and approximate algorithm have given support to the development of a decision support system in a port terminal of São Luís. Insights about which kind of optimization procedure is more suitable for specific planning horizons have been used for the development team. Although greedy algorithms have been the main alternative for planning horizons above one month, they cannot provide reasonable robustness for reliable results. The preliminary results of the proposed SA rather demonstrates reasonable robustness, accuracy and satisfactory runtime.

Further improvements are intended to be made by representing heterogeneous berths, for which the decision about where berthing vessels deserves similar attention to decision about when do it. In such complex decision making scenarios, the possibility of integrating reliable approximate optimization procedures is fundamental for valuable well-succeeded experiences with enterprise management systems.

Acknowledgments The authors acknowledge the useful comments and suggestions of the referees. This research is partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), proc. 473205/2008-4, 300692/2009-9, 558084/2009-5 and 470813/2010-5.

References

[Günther and Kim,2005] Günther, H.O., Kim, K.H., Container terminals and
automated transport systems: logistics control issues and quantitative de-

[Hansen et al., 2007] Hansen, P., Oğuz, C., Mladenovic, N. Variable Neigh-
borhood Search for Minimum Cost Berth Allocation, European Journal of

[Imai et al., 1997] Imai, A., Nagaiwa, K., Chan, W.T., Efficient planning of
berth allocation for container terminals in Asia. Advanced Transportation.

[Imai et al., 2001] Imai, A., Nishimura, E., Papadimitriou, S., The dynamic
berth allocation problem for a container port. Transportation Research-B.

[Imai et al., 2003] Imai, A., Nishimura, E., Papadimitriou, S., Berth allocation

[Imai et al., 2006] Imai, A., Nishimura, E., Papadimitriou, S., Berthing ships
at a multi-user container terminal with a limited quay capacity. Transportation

[Kim and Moon, 2003] Kim, K.H. and Moon, K.C., Berth scheduling by sim-


v22. 105-110. 1998

[Lorenzoni et al., 2006] Lorenzoni, L.L., Ahonen, H., Alvarenga, A.G., A
multi-mode resource-constrained scheduling problem in the context of port

[Nishimura and Imai, 2001] Nishimura, E., Imai, A., Papadimitriou, S., Berth
allocation planning in the public berth system by genetic algorithms. European

[Papadimitriou and Steiglitz, 1982] Papadimitriou, C.H., Steiglitz, K., Com-
binatorial Optimization: Algorithms and Complexity. Prentice-Hall, Engle-


[Pinedo, 1995] Pinedo, M., Scheduling: Theory, Algorithms and Systems,

cvrd.com.br, [2008, March 31].

at a container terminal: An overview, European Journal of Operational Re-
Fig. 1. Operational scenario in a tidal bulk port with homogeneous berth position.

Fig. 2. BAPTS modeled as transportation problem
Table 1
Problem instance features and optimal values

<table>
<thead>
<tr>
<th>Instance</th>
<th>Ships</th>
<th>Berths</th>
<th>TTW</th>
<th>time (s)</th>
<th>Eq. 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>1</td>
<td>31</td>
<td>104.70</td>
<td>3926.0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1</td>
<td>28</td>
<td>0.50</td>
<td>3939.5</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>29</td>
<td>0.30</td>
<td>4626.5</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>1</td>
<td>32</td>
<td>230.45</td>
<td>4384.5</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>1</td>
<td>34</td>
<td>57.84</td>
<td>4603.5</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>1</td>
<td>44</td>
<td>12061.09</td>
<td>7671.0</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>1</td>
<td>47</td>
<td>9.64</td>
<td>9844.5</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>1</td>
<td>44</td>
<td>228.64</td>
<td>8202.0</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>1</td>
<td>41</td>
<td>1.30</td>
<td>8052.5</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>2</td>
<td>26</td>
<td>3.30</td>
<td>5451.0</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>1</td>
<td>58</td>
<td>3470.78</td>
<td>14671.0</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>1</td>
<td>57</td>
<td>12.94</td>
<td>15151.5</td>
</tr>
<tr>
<td>13</td>
<td>20</td>
<td>1</td>
<td>60</td>
<td>2596.34</td>
<td>15153.5</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>1</td>
<td>63</td>
<td>48604.41</td>
<td>16965.0</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>1</td>
<td>54</td>
<td>15.52</td>
<td>16087.0</td>
</tr>
<tr>
<td>16</td>
<td>30</td>
<td>2</td>
<td>44</td>
<td>253.83</td>
<td>18480.0</td>
</tr>
<tr>
<td>17</td>
<td>30</td>
<td>3</td>
<td>31</td>
<td>5.73</td>
<td>14456.0</td>
</tr>
<tr>
<td>18</td>
<td>30</td>
<td>1</td>
<td>85</td>
<td>1130.33</td>
<td>30285.0</td>
</tr>
<tr>
<td>19</td>
<td>30</td>
<td>2</td>
<td>47</td>
<td>3388.41</td>
<td>18264.0</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>2</td>
<td>47</td>
<td>2469.52</td>
<td>19244.0</td>
</tr>
<tr>
<td>21</td>
<td>30</td>
<td>2</td>
<td>48</td>
<td>501.00</td>
<td>20910.5</td>
</tr>
</tbody>
</table>
Fig. 3. Berthing decision in instance 17.

Fig. 4. Stock level on the planning horizon for instance 11: a) with stock level constrained b) without stock level constraints
Fig. 5. High Demurrage Ship heuristic

Table 2
Simulated Annealing’s results

<table>
<thead>
<tr>
<th>Instance</th>
<th>Optimal</th>
<th>FS ± τ²</th>
<th>Error%</th>
<th>Hits</th>
<th>FC</th>
<th>CPLEX RT (s)</th>
<th>SA RT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3926</td>
<td>3926 ± 0</td>
<td>0%</td>
<td>20</td>
<td>168633.7</td>
<td>104.7</td>
<td>6.64</td>
</tr>
<tr>
<td>2</td>
<td>3939.5</td>
<td>3939.5 ± 0</td>
<td>0%</td>
<td>20</td>
<td>28054.5</td>
<td>0.5</td>
<td>1.95</td>
</tr>
<tr>
<td>3</td>
<td>4626</td>
<td>4626 ± 0</td>
<td>0%</td>
<td>20</td>
<td>3700.6</td>
<td>0.3</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>4384</td>
<td>4384 ± 0</td>
<td>0%</td>
<td>20</td>
<td>105408.7</td>
<td>230.45</td>
<td>4.7</td>
</tr>
<tr>
<td>5</td>
<td>4603.5</td>
<td>4603.5 ± 0</td>
<td>0%</td>
<td>20</td>
<td>9872.9</td>
<td>57.84</td>
<td>1.1</td>
</tr>
<tr>
<td>6</td>
<td>7671</td>
<td>7671 ± 0</td>
<td>0%</td>
<td>20</td>
<td>172666.6</td>
<td>12061.09</td>
<td>14.1</td>
</tr>
<tr>
<td>7</td>
<td>9844.5</td>
<td>9844.5 ± 0</td>
<td>0%</td>
<td>20</td>
<td>155665.2</td>
<td>9.64</td>
<td>12.7</td>
</tr>
<tr>
<td>8</td>
<td>8202</td>
<td>8202 ± 0</td>
<td>0%</td>
<td>20</td>
<td>178633</td>
<td>228.64</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>8052.5</td>
<td>8052.5 ± 0</td>
<td>0%</td>
<td>20</td>
<td>114368.7</td>
<td>1.3</td>
<td>9.7</td>
</tr>
<tr>
<td>10</td>
<td>5451</td>
<td>5451 ± 0</td>
<td>0%</td>
<td>20</td>
<td>184902.1</td>
<td>3.3</td>
<td>14.8</td>
</tr>
<tr>
<td>11</td>
<td>14671</td>
<td>14672.5 ± 2.66</td>
<td>0.01%</td>
<td>16</td>
<td>370444.7</td>
<td>3470.78</td>
<td>42.1</td>
</tr>
<tr>
<td>12</td>
<td>15151.5</td>
<td>15151.5 ± 0</td>
<td>0%</td>
<td>20</td>
<td>195914.3</td>
<td>12.94</td>
<td>28.7</td>
</tr>
<tr>
<td>13</td>
<td>15153.5</td>
<td>15153.5 ± 0</td>
<td>0%</td>
<td>20</td>
<td>402075</td>
<td>2596.34</td>
<td>45.4</td>
</tr>
<tr>
<td>14</td>
<td>16965</td>
<td>16971 ± 14.17</td>
<td>0.04%</td>
<td>16</td>
<td>359523.13</td>
<td>48604.41</td>
<td>50</td>
</tr>
<tr>
<td>15</td>
<td>16087</td>
<td>16087 ± 0</td>
<td>0%</td>
<td>20</td>
<td>281006.2</td>
<td>15.52</td>
<td>37.5</td>
</tr>
<tr>
<td>16</td>
<td>18480</td>
<td>18480 ± 0</td>
<td>0%</td>
<td>20</td>
<td>477112.2</td>
<td>253.83</td>
<td>104.2</td>
</tr>
<tr>
<td>17</td>
<td>14456</td>
<td>14456 ± 0</td>
<td>0%</td>
<td>20</td>
<td>174602</td>
<td>5.73</td>
<td>11.3</td>
</tr>
<tr>
<td>18</td>
<td>30285</td>
<td>30285 ± 0</td>
<td>0%</td>
<td>20</td>
<td>291457.8</td>
<td>1130.33</td>
<td>56</td>
</tr>
<tr>
<td>19</td>
<td>18264</td>
<td>18439 ± 216.9</td>
<td>0.96%</td>
<td>10</td>
<td>238275.4</td>
<td>3388.41</td>
<td>83.3</td>
</tr>
<tr>
<td>20</td>
<td>19244</td>
<td>19258 ± 8.94</td>
<td>0.07%</td>
<td>1</td>
<td>15223</td>
<td>2469.52</td>
<td>144.8</td>
</tr>
<tr>
<td>21</td>
<td>20910.5</td>
<td>20933.5 ± 13.33</td>
<td>0.11%</td>
<td>6</td>
<td>494703.5</td>
<td>501</td>
<td>125.6</td>
</tr>
</tbody>
</table>