



HAL
open science

Solving a Berth Assignment Problem

Serigne Gueye, Sophie Michel, Adnan Yassine

► **To cite this version:**

Serigne Gueye, Sophie Michel, Adnan Yassine. Solving a Berth Assignment Problem. 2009. hal-02457954v2

HAL Id: hal-02457954

<https://hal.science/hal-02457954v2>

Preprint submitted on 6 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Solving a Berth Assignment Problem

Serigne Gueye Sophie Michel Adnan Yassine

*Université du Havre,
Laboratoire de Mathématiques Appliquées du Havre,
BP 540,
76058 Le Havre cedex, France*

Abstract

The Berth Allocation Problem (BAP) is the problem of allocating berthing spaces and scheduling container vessels on these spaces so as to minimize total weighted time. We study a version of BAP in which containers are moved between vessels and berth space is abundant. Thus, the problem reduces to optimally assign vessels to berths. We call it the Berth Assignment Problem (BASP). We formulate it as a non standard Quadratic Assignment Problem, and we show that BASP is NP-Hard. The formulation is simplified, linearized, and valid inequalities are found. Numerical results are shown.

Key words:

Vessels Assignment, Port Terminal, 0-1 Quadratic Programming, Linearization, Linear Programming, Constraint Programming

1 Introduction

The Berth Allocation Problem (BAP) is the problem of allocating berth spaces to vessels in a container terminal. The general BAP problem stands as follows. Along the wharf of a terminal, n container vessels, with known arrival dates a_i , have to be berthed. Depending on their sizes, as well as the number of containers in the vessel, each of them needs a known length along the wharf. To operate (uploading and downloading operations) a vessel a processing time p_i is necessary. The problem is then to find **when** (date u_i) and **where** a vessel has to be assigned in order to minimize some costs without exceeding the wharf capacity (length). Several Berth Allocation Problems have appeared in the literature differing on the assumptions made.

One may consider the wharf as a continuous line in which a vessel may berth

anywhere, or as a collection of sections (i.e subdivisions) in which vessels are allowed to berth. The first case is known as the continuous case while the second is the discrete one. The costs to optimize is in general the vessel waiting times computed as follows

$$\min \sum_{i=1}^n w_i(u_i + p_i - a_i) \text{ i.e sum of waiting times}$$

But other objectives may be considered. For instance, H. Wai[6] try to solve a multiobjective problem in which one of the objective is the waiting time and the other a container transshipment cost. In this type of problem, it is supposed that between vessels arriving in the port flows of containers exist. These flows are summarized in a flow matrix of containers giving how many containers a vessel has to transfer to another one and vice versa. For a complete survey of models and solution methods for the Berth Allocation Problem, the reader may be referred to Guan and Cheun[2] , Cordeau et al[1], Lim[5], Wang and Lim[7] among others.

In the problems described above, the operational details are omitted. Indeed, the way that containers are exactly handled are not taken into account. For instance, in transshipment of containers, a direct transshipment (from vessel to vessel) is not always necessary . A container can be uploaded on the yard area, then stocked and moved latter in another yard area in order to be downloaded in another vessel. These operational aspects are relaxed. As a consequence, allocating vessels on berths may be viewed as a tactical step of a port logistic management.

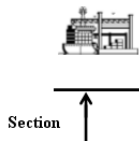
We are interested in another special case of BAP. This case was proposed by the Information Technology Company GTI (Le Havre) as a subproblem embedded in a Transport and Logistic package. We consider the discrete case in which the wharf is divided in sections of equal size. n vessels have to be moored and each of them needs a number b_i ($i = 1, 2, \dots, n$) of sections. We suppose that there are enough sections for all the vessels and that some containers have to be moved between vessels. As in the Wai problem the container flows are summarized in a matrix. Since the number of sections is sufficient, the time dimension dropped and the problem consists only in assigning optimally vessel to sections in such a way to minimize container transshipment costs. That is the reason why we call this problem a Berth Assignment Problem (BASP) instead of a Berth Allocation Problem (BAP). Because of the number of vessels to consider in practice (says ≥ 10) and the combinatorial complexity of the problem ($n!$ possible assignments), solving such problems in a "reasonable" amount of time is only possible using heuristics. We have proposed in 2005 a greedy heuristic (see Gueye [3]). Nevertheless, the evaluation of the solution quality is possible only in comparison with the optimal one's. Because lower is the gap between the heuristic solution value and the optimal value and better

is the heuristic. In this paper, we are interested in solving exactly this problem.

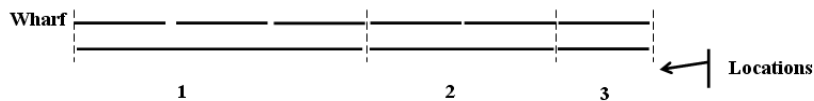
The BASP is explained in detail in section 2. We modelize it in section 3 as a non standard version of the well known Quadratic Assignment Problem. This version is NP-Hard as the QAP. We propose a linearization of the model in the same section. This linearization leads theoretically to a very poor lower bound. To improve it some valid inequalities are proposed in section 4. The integer linear problem presents in section 3 and the valid inequalities are used together in an algorithm. This algorithm has been implemented and tested on instances derived from QAPLIB Benchmark. Numerical results are shown in section 5. Finally, some perspectives about this work are point out in section 6.

2 The Berth Assignment Problem

Let us define a section as an area, of known length, along the wharf, in which some containers may be downloaded or uploaded from a vessel. This definition is schematically illustrated in the figure below where the bold line represents a section.



We define a "location" as a contiguous series of sections. The size of a location is the number of its sections. If n is the number of vessels (resp. locations) then each vessel (resp. location) may be identified by an index $i = 1, 2, \dots, n$ (resp. $k = 1, 2, \dots, n$). If we consider a wharf as the horizontal line below, locations are arranged starting from the left-hand-side, to the right-hand-side, in increasing order of the indices. In this example the size of the location 1, 2, 3 are respectively 3, 2, 1.



Vessels arriving in the port have to be assigned in locations. For each vessel the length of its location depends on its size, but also on the number of containers to handle in the vessel. We will notice b_i the location size of the vessel i . If S is the total number of sections of the wharf we must have

$$\sum_{i=1}^n b_i = S.$$

It is important to observe that if k is the index of a location then its size depends on the vessel assigns to it. In other words, the size of a location is not known a priori but it is determined only when a vessel is assigned to the location. Therefore, let π be an assignment of vessels on locations, we will notice $s(\pi, k)$ the size of the location k .

Vessels, assigned along the wharf, exchange some containers, mathematically expressed by a traffic matrix $F = \{f_{ij}\}_{i,j=1,2,\dots,n}$ where i and j are vessel indices, and $f_{ii} = 0$ for all i .

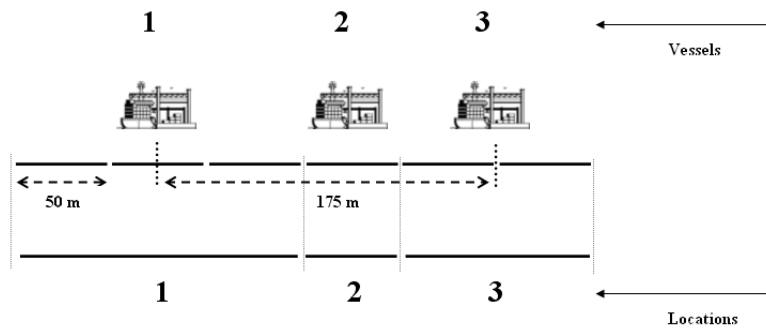
Let π be an assignment where $\pi(i) = k$ (i.e vessel i is assigned to location k), $\pi(j) = l$ (i.e vessel j is assigned to location l). The cost of moving f_{ij} containers from k to l is $f_{ij}d_{kl}(\pi)$, where $d_{kl}(\pi)$ is the distance (depending on the assignment π) between locations k and l . This distance is evaluated as follows.

Let us notice α (meters) the length of a section. For $k < l$, we have

$$d_{kl}(\pi) = \left(\frac{s(\pi, k)}{2} + \sum_{u=k+1}^{l-1} s(\pi, u) + \frac{s(\pi, l)}{2} \right) \alpha \text{ meters.} \quad (1)$$

The overall cost of the assignment π is then : $\sum_{i=1}^n \sum_{j=1}^n f_{ij}d_{\pi(i)\pi(j)}(\pi)$ meters.

Example 2.1 *Let us consider the following example to illustrate mathematical expressions above.*



In this example, 3 vessels (1, 2, 3) are assigned to 3 locations (1, 2, 3). This assignment corresponds to the identity permutation $\pi(i) = i$, $i = 1, 2, 3$.

The traffic matrix between vessels is given by

$$F = \begin{bmatrix} 0 & 10 & 20 \\ 8 & 0 & 2 \\ 10 & 10 & 0 \end{bmatrix}$$

Let us consider vessels 1 and 3 assigned respectively to locations 1 and 3. Then, if the size of a section is 50 meters then the distance between locations 1 and 3 will be

$$d_{13}(\pi) = \frac{150}{2} + 50 + \frac{100}{2} = 175 \text{ m}$$

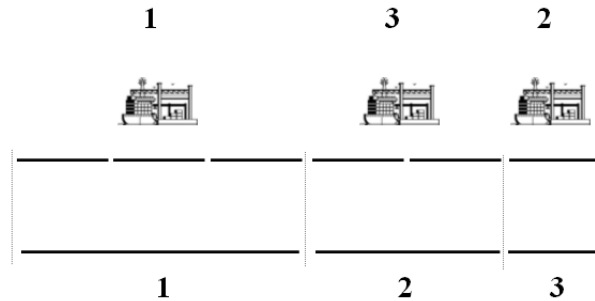
More generally, with the assignment π , we have the following distance matrix between locations :

$$D(\pi) = \begin{bmatrix} 0 & 100 & 175 \\ 100 & 0 & 75 \\ 175 & 75 & 0 \end{bmatrix}$$

And the overall cost of π is

$$(10+8)*100 + (20+10)*175 + (10+2)*75 = 7950 \text{ meters (i.e 7 km 950 m)}$$

Let us observe what happens with the other assignment $\pi^*(1) = 1, \pi^*(2) = 3, \pi^*(3) = 2$ shown below.



In this case, we have the following different distance matrix between locations :

$$D(\pi^*) = \begin{bmatrix} 0 & 125 & 200 \\ 125 & 0 & 75 \\ 200 & 75 & 0 \end{bmatrix}$$

And the overall cost of π^* is

$$(10+8)*200 + (20+10)*125 + (10+2)*75 = 8250 \text{ meters (i.e 8 km 250 m)}$$

□

The movements of containers between locations need human and material resources, and are time-consuming. Higher is the distance between two vessels and higher will be the economic cost induced by the container movements. This imply that lower is the total overall distance to transfer all containers and lower is the associated economic cost for the port manager. Thus, we have to solve the following optimization problem :

$$(BASP) : \underset{\pi \in S_n}{Min} \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{\pi(i)\pi(j)}(\pi).$$

where S_n is the permutation set of $\{1, 2, \dots, n\}$, $F = \{f_{ij}\}_{i,j=1,2,\dots,n}$ is a known traffic flow matrix and $D(\pi) = \{d_{kl}(\pi)\}_{k,l=1,2,\dots,n}$ is a distance matrix whose values depend on the assignment π . This mathematical programming problem (*BASP*) may be viewed as a "non standard" version of the well known Quadratic Assignment Problem (QAP). At the opposed of the "standard" QAP where the distance matrix D is fixed (i.e does not depends on any assignment π), in the *BASP* this matrix may change.

The *BASP* assignment is a very difficult combinatorial optimization problem (NP-Hard). Hence, because of the number of vessels to consider in practice and the complexity of the problem, we have proposed in 2005 (see [3]) a greedy heuristic for the IT consulting company GTI. In the sequel, a formulation and an exact resolution scheme are now developed.

3 Mathematical formulation

Let us denote by $i, j \in \{1, 2, \dots, n\}$ the vessel indices and by $k, l \in \{1, 2, \dots, n\}$ the location indices. As in the standard QAP formulation, we define the following binary variables

$$x_{ik} = \begin{cases} 1 & \text{if vessel } i \text{ is assigned to location } k \\ 0 & \text{otherwise} \end{cases}$$

These variables have to verify the standard QAP constraints :

$$\sum_{i=1}^n x_{ik} = 1 \quad \text{and} \quad \sum_{k=1}^n x_{ik} = 1.$$

Let $x = \{x_{ik}\}_{1 \leq i,k \leq n}$ be the variables matrix. Since an assignment of x corresponds to a permutation, the overall cost that has to be minimized in the

problem is $\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f_{ij} d_{kl}(x) x_{ik} x_{jl}$.

The expression of $d_{kl}(x)$ is obtained as follows. Let us first recall the equation (1) $d_{kl}(\pi)$ for $k < l$:

$$d_{kl}(\pi) = \left(\frac{s(\pi, k)}{2} + \sum_{u=k+1}^{l-1} s(\pi, u) + \frac{s(\pi, l)}{2} \right) \alpha \text{ meters},$$

where $s(\pi, k)$ is the size of the location k in the case of permutation π . Since, in our notation, x stands now for π , we can also write

$$d_{kl}(x) = \left(\frac{s(x, k)}{2} + \sum_{u=k+1}^{l-1} s(x, u) + \frac{s(x, l)}{2} \right) \alpha \text{ meters}.$$

By definition of $s(x, k)$, x and b_i (number of sections need for vessel i), we have

$$s(x, k) = \sum_{m=1}^n b_m x_{mk}.$$

Thus, for $k < l$, $d_{kl}(x)$ is $d_{kl}(x) = \left(\frac{1}{2} \sum_{m=1}^n b_m x_{mk} + \sum_{u=k+1}^{l-1} \sum_{m=1}^n b_m x_{mu} + \frac{1}{2} \sum_{m=1}^n b_m x_{ml} \right) \alpha$.

Hence, for any values for k and l with $k \neq l$

$$d_{kl}(x) = \left(\frac{1}{2} \sum_{m=1}^n b_m x_{mk} + \sum_{u=\min\{k,l\}+1}^{\max\{k,l\}-1} \sum_{m=1}^n b_m x_{mu} + \frac{1}{2} \sum_{m=1}^n b_m x_{ml} \right) \alpha.$$

When $k = l$, we have $d_{kl}(x) = 0$.

The notations above give the complete mathematical program of the Berth Assignment Problem (BASP) :

$$\begin{aligned} \text{(BASP)} \quad & \text{Min} \quad \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f_{ij} d_{kl}(x) x_{ik} x_{jl} \\ \text{s-t : (1)} \quad & d_{kl}(x) = \left(\frac{1}{2} \sum_{m=1}^n b_m x_{mk} + \sum_{u=\min\{k,l\}+1}^{\max\{k,l\}-1} \sum_{m=1}^n b_m x_{mu} + \frac{1}{2} \sum_{m=1}^n b_m x_{ml} \right) \alpha, \\ & 1 \leq k, l \leq n, |k-l| \geq 2 \\ \text{(2)} \quad & d_{kl}(x) = \left(\frac{1}{2} \sum_{m=1}^n b_m x_{mk} + \frac{1}{2} \sum_{m=1}^n b_m x_{ml} \right) \alpha, \\ & 1 \leq k, l \leq n, |k-l| = 1 \\ \text{(3)} \quad & d_{kk}(x) = 0, \quad 1 \leq k \leq n \\ \text{(4)} \quad & \sum_{i=1}^n x_{ik} = 1, \quad 1 \leq k \leq n \\ \text{(5)} \quad & \sum_{k=1}^n x_{ik} = 1, \quad 1 \leq i \leq n \\ & x \in \{0, 1\}^{n^2} \end{aligned}$$

This is 0-1 quadratic problems for which we propose the following linearization.

Since α (the section length) is a constant, we can remove this value to constraints (1) and (2). Moreover, a formulation with only variables x , flow matrix F and sizes b can be derived by replacing $d_{kl}(x)$ with its corresponding explicit expressions (constraints (1),(2),(3)). Now, let y_{kml} be the following variables :

$$y_{kml} = \sum_{u=\min\{k,l\}+1}^{\max\{k,l\}-1} x_{mu} , \quad k, l, m = 1, 2, \dots, n \text{ and } |k - l| \geq 2$$

$$y_{kml} = 0, \quad k, l, m = 1, 2, \dots, n \text{ and } |k - l| \leq 1,$$

and $t_{imj} = \sum_{l=1}^n \sum_{k=1}^n y_{kml} x_{ik} x_{jl}$. We have shown that using these variables LBASP may be linearized as follows :

(LBASP)	Min	$\sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n \sum_{\substack{m=1 \\ m \neq i, j}}^n f_{ij} b_m t_{imj}$	
	s-t : (4)	$\sum_{i=1}^n x_{ik} = 1,$	$1 \leq k \leq n$
	(5)	$\sum_{k=1}^n x_{ik} = 1,$	$1 \leq i \leq n$
	(9)	$t_{imj} \geq \sum_{u=\min\{k,l\}+1}^{\max\{k,l\}-1} x_{mu} + x_{ik} + x_{jl} - 2$	$1 \leq i, j, m, k, l \leq n, i \neq j,$ $m \neq i, j, k - l \geq 2$
		$t_{imj} \geq 0$	$1 \leq i, j, m \leq n, i \neq j, m \neq i, j$
		$x \in \{0, 1\}^{n^2}$	

This is the formulation that we seek to solve using linear integer programming techniques, in particular Branch-and-Bound algorithm. It is well-known that the corresponding linear relaxation gives a lower bound of the optimal value of the problem (LBASP). It is also well-known that to reduce the Branch-and-Bound tree size, as well as the processing time, the gap (between the lower bound and the optimal value) should be as weak as possible. Unfortunately, the linear relaxation of LBASP gives the poorest bound that may be expected. Indeed, the optimal value of the relaxation is always equal to 0 as proved in the property below.

Propriety 3.1 *Let (\overline{LBASP}) the linear relaxation of (LBASP), and $V(LBASP)$ the optimal value of (LBASP). We have $V(\overline{LBASP}) = 0$.*

This result show that it remains to find many valid inequalities of the (LBASP) feasible domain. The section 4 below deals with these inequalities.

4 Valid inequalities

We begin this section with the following property. It will be useful to reduce the number of constraints of (*LBASP*), as well as to find some valid inequalities.

Propriety 4.1 .

(i) $t_{imj} = 1$ if and only if vessel m is between vessel i and j in the optimal assignment.

(ii) $t_{imj} = t_{jmi}$

Notice, for the point (i), that when $t_{imj} = 1$ we only know that m is between i and j without any information about the position of i (resp. j) in comparison to m . In other words, we don't know if i (resp. j) is at the left (resp. right) of m or not. With this property it follows that only three cases are (exclusively) possibles : m is between i and j or i is between m and j , or j is between i and m . We thus derive our first valid equalities below

Proposition 4.2 . $t_{imj} + t_{mij} + t_{ijm} = 1 \forall 1 \leq i < m < j \leq n$.

Taking into account these equalities and the symmetry property (ii), (*LBASP*) may be strengthened as follows

(<i>LBASP*</i>)	Min	$\sum_{i=1}^n 2 \sum_{j=i+1}^n \sum_{\substack{m=1 \\ m \neq i, j}}^n f_{ij} b_m t_{imj}$	
	s-t :	(4)	$\sum_{i=1}^n x_{ik} = 1, \quad 1 \leq k \leq n$
		(5)	$\sum_{k=1}^n x_{ik} = 1, \quad 1 \leq i \leq n$
		(9)	$t_{imj} \geq \sum_{u=\min\{k,l\}+1}^{\max\{k,l\}-1} x_{mu} + x_{ik} + x_{jl} - 2 \quad 1 \leq i, j, m, k, l \leq n, i < j,$
			$m \neq i, j, k-l \geq 2$
		(10)	$t_{imj} \geq 0 \quad 1 \leq i, j, m \leq n, i \neq j, m \neq i, j$
		(11)	$t_{imj} + t_{mij} + t_{ijm} = 1 \quad \forall 1 \leq i < m < j \leq n$
			$x \in \{0, 1\}^{n^2}$

Let observe that the linear relaxation of this formulation discard the solution $x_{ik} = \frac{1}{n}$, $t_{imj} = 0$. However, constraints (11) do not described the entire convex hull of (*LBASP*). We have found an exponential number of other valid inequalities, derived from the meaning of variables t_{imj} (see property 4.1 (i)), and by observing the optimal solution of the linear relaxation. Because of the huge number of valid inequalities, it is not possible, for computer memory reasons, to add all of them simultaneously. Inequalities have to be added progressively, using an iterative algorithm for which at each iteration a so-called

”separation problem” is solved and gives as a result a new inequality for the formulation. Such type of algorithm is called a *cut (or constraint) generation algorithm* in the literature. We have implemented this type of algorithm. At each iteration the separation problem is solved using Constraint Programming (CP) Techniques. To obtain accurate details about the algorithm, the separation problem and CP techniques used we refer the reader to the technical report (Gueye [4]).

5 Numerical Tests

With our algorithms, some numerical tests have been performed. Since no practical datas or academic benchmarks were available , we generate ourselves some instances. As the Berth Assignment Problem is a variant of the Quadratic Assignment Problem, our instances have been derived from QAPLIB benchmark. Each instance is define by the container flow matrix, noticed F , and by the number of sections occupied by each vessel, noticed b_i ($i = 1, 2, \dots, n$).

Numerical experiments have been performed on a Personal Computer Dell, with 2 G0 RAM, Intel Core Processor of 1.86 GHz on linux Suze Operating System. Ilog Cplex 9.3 and Solver 9.3 have been used. The results are reported in table 1. We indicate in column 3 of this table the optimal value of $(\overline{LBASP^*})$ (i.e the linear relaxation of $(LBASP^*)$), and in colum 4 the optimal value of $(BASP)$. In column 5 the number of nodes of the Cplex 9.3 branch and bound tree is reported. We have fixed the Cplex parameters MIPEmphasis to 1 in order to work toward a rapid feasibility. The experiences show that, in our formulation, emphasizing on feasibility is better (less computational time) than emphasizing on optimality. The Cplex MIP heuristic frequency has been switched off. The processing time has been limited to 1 hour. When the processing time exceed this value, no result is given in the table. All other parameters have been leaved at their default values.

The maximal number of vessels that we are able to assign optimally is currently lower or equal to 10. In all cases, the gap between lower bounds and optimal values do not exceed 12% from the optimal values. But, the computational times grow exponentially with the size n . This is explain by the huge amount of constraints, in particular constraints 9, of the formulation $(LBASP^*)$. The instances chr12a.dat and chr12b.dat seem easiers since no gap exists.

Name	n	Lower Bound (LB)	Optimal Value	Nodes	LB Time	Total Time
els19.dat	7	17385.9	17962.0	48	0.5	2.0
	8	37730.1	41016.0	304	3.7	25.6
	9	75023.5	85610.0	4041	16.8	624.9
lipa20a.dat	7	54.0	55.0	1	0.4	0.9
	8	90.3	92.0	35	0.4	8.2
	9	137.7	142.0	190	52.6	314.6
	10	200.2	206.0	219	1485.6	2937.9
rou12.dat	7	9980.6	10552.0	54	0.5	2.0
	8	20731.6	21994.0	111	2.0	11.5
	9	34851.3	38646.0	1561	23.8	352.3
	10	45204.0	48928.0	1449	65.4	1279.6
chr12a.dat	7	86.0	86.0	0	0.3	0.3
	8	106.0	106.0	0	0.2	0.2
	9	106.0	106.0	0	25.2	25.2
	10	106.0	106.0	0	126.0	126.0
chr12b.dat	7	172.0	172.0	0	1.0	1.5
	8	192.0	192.0	0	5.0	5.0
	9	192.0	192.0	0	109.2	109.2
	10	194.0	194.0	0	915.0	915.0

Table 1
Numerical results

6 Conclusion

This paper deals with a difficult combinatorial optimization problem. In this problem vessels have to be assigned along the wharf of a port terminal in such a way to minimize the total transshipment cost of containers. The problem has been formulated as a non standard version of the Quadratic Assignment Problem. It has been shown that this version is NP-Hard. Following many reformulation steps, and a linearization, we have obtained a integer linear formulation. But this formulation leads to poor lower bound. Without adding new valid inequalities, it is not possible to solve optimally the problem within a reasonable amount of time. We have found an exponential number of such valid inequalities. Then, a cut generation algorithm has been implemented to generate these inequalities. At each iteration of our algorithm, Constraint Programming techniques are used to find the cuts.

Currently, the maximal number of vessels that we may assign optimally is 10. This means that it remains further works in order to improve the numerical results. Such improvement may be reach considering another formulation with less constraints than the one presented in this paper.

Following the improvements, we seek also to take into account the arrival time of the vessels. This aspect will lead to a challenging scheduling problem and will make the problem formulation closer to the reality.

References

- [1] J-F. Cordeau, G. Laporte, P. Legato, and L. Moccia. Models and tabu search heuristics for the berth allocation problem. *Transportation Science*, 39(4):526–538, 2005.
- [2] Y Guan and R.K. Cheung. The berth allocation problem : Models and solution methods. *OR Spectrum*, 26(1):75–92, 2004.
- [3] S. Gueye. Optimisation de l’affectation de ”trades”. Technical report, Laboratoire de Mathematiques Appliquees du Havre (LMAH), 2005.
- [4] S. Gueye, S. Michel, and A. Yassine. Solving a berth assignment problem. Technical report, LMAH, September 2009.
- [5] Andrew Lim. The berth planning problem. *Operations Research Letters*, 22:105–110, 1998.
- [6] Heon Wai. Berth allocation and planning. (cover story). *Innovation: The Magazine of Research and Technology, World Scientific*, 6(1):20–21, 2007.
- [7] Fan Wang and Andrew Lim. A stochastic beam search for the berth allocation problem. *Decision Support Systems*, 42:2186–2196, 2007.