v. 3, n. 6, 2023

Journal of
Engineering
Research

A HEURISTIC APPROACH FOR SOLUTION OF THE BERTH ALLOCATION PROBLEM IN BULK PORTS

Eolo Aparecido Caetano Rosa
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Belo Horizonte, MG, Brazil

Gustavo Campos Menezes
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Belo Horizonte, MG, Brazil

Abstract: The international trade of raw materials in bulk is deeply dependent on maritime transport and that is why ports have been induced to invest in infrastructure and logistics to improve the supply of ships. In this context, the present work addresses the Berth Allocation Problem (BAP): where the objective is to allocate ships to berths in order to minimize operating costs. The PAB can be considered as one of the main optimization problems in maritime terminals, due to the growing demand of ships that transport products in bulk. Thus, an algorithm that combines the LNS heuristic method and simple permutation will be proposed. The results of the computational tests show that the PAB-HE heuristic was able to produce a relative GAP lower than 10.66% in relation to the lower limit obtained by the CPLEX® solver for instances with speed of the cradles with a penalty of 25% and 17.18% GAP for instances with cradle speed with a 50% penalty, in a computational time of less than 20 seconds.
Keywords: Berth Allocation Problem, LNS Heuristic, Simple Permutation.

INTRODUCTION

Maritime transport is one of the fundamental pillars of the growth of the global economy throughout history. The increase in the demand for raw materials, the growth of international trade transactions and the advances in techniques for building large bulk carriers lead to an increase in the frequency of transport operations between ports and, consequently, to an increase in their importance. in the results of the trade balance of the countries.

Data from the statistical yearbook of the National Waterway Transport Agency (ANTAQ) indicate that Brazilian ports increased their cargo handling by 4.2% in fiscal year 2020 compared to 2019, reaching a total of 1.151 billion tons [ANTAQ, 2021]. Most of this volume is solid bulk, which accounts for about 60% of all cargo handled by the terminals. The advance of this segment in fiscal year 2020 was around 1.2%.

Low cost ceased to be a competitive advantage and became the rule, there was a sudden appearance in a large number of logistics operators and their hiring by most companies as highlighted in [Fioroni, 2008].

In this research context, this article intends to propose a PAB solution in solid bulk port terminals (iron ore, coal and grain).

LITERATURE REVIEW

The BAP consists of allocating the ships to the mooring positions in such a way that the maximum space on the pier is used, minimizing the service time. The decisions to be taken concern the position (where?) and the time (when?) in which the ship must dock [Imai et al., 2001].

In Imai et al. [1997, 2001], a PABD was presented, where all ships that have an arrival time within the planning horizon are considered in the model. The PABD model assumes that all ships can be allocated to any berth, which is not usual in practice. The authors developed a heuristic method based on a subgradient method with Lagrangian relaxation to find solutions for the DBAP. Nishimura et al. [2001]; Xu et al. [2012] expanded the dynamic version of the PABD, delimiting the pier where each ship can berth, and improved its approach considering different supply priorities between ships. Nishimura et al. [2001], proposed an algorithm Genetic (GA) as a solution method and Xu et al. [2012], modeled the PABD as a machine programming problem where assignment of ships to berths takes into account depth constraints and tidal conditions.

In research by Banos et al. [2016], they presented a variant of the PABD proposed by [Cordeau et al., 2005]. They proposed a mutilmatic model and a metaheuristic Simuluntild Annealing (SA) for the Berth Allocation Problem with Multiple Loads (PBMC). The proposal by Cordeau et al. [2005] is based on the premise that the operating times of ships in the berths are known a priori and in the multi-purpose model proposed by the authors, the operating time is calculated and depends on the relationship between the cargo transported by the ship and the berth it will dock. Therefore, the operating time is not known a priori.

In Correcher et al. [2019], they presented a PABD variant, a Mixed Integer Linear Programming (MILP) formulation, a heuristic based on Iteruntild Local Search (ILS) and a Ruin & Recreuntil framework, which were able to obtain optimal solutions or almost optimal for this new variant of the problem.

In the PABC, the area (pier) destined to carry out loading and/or unloading operations does not have any subdivision and the ships can berth in any position according to [Bierwirth and Meisel, 2010].

And PABH has the characteristics of both PABD and PABC according to [Imai et al., 2005] and [Meisel and Bierwirth, 2009]. In the PABH, the area (pier) destined to carry out loading and/or unloading operations is divided into a set of sections or berths; however, there is the possibility that a larger ship occupies more than one berth and, furthermore, that a berth is occupied by more than one ship.

PROBLEM DESCRIPTION

Considering the BAP formulation proposed by Barros [2010], modeled in discrete form as a transport problem in which N ships are seen as consumers and L berths and tide windows favor Tidal Time Windows (TTW) M, as providers. Where each ship i must be allocated to a berth l in a subset of TTW's windows whose cardinality depends on the number of required tides so that the loading and/or unloading operation is completed.

The input set (Table 1), decision variables (Table 2) and formulas of expression 1 of the model are presented below with a brief description. The phorometer hil represents the processing time necessary for a ship i to complete the loading and/or unloading operation at berth l [Barros, 2010].

Set	Description
N	set of ships
M	set of *ttw's*
L	set of mooring positions (berths)
K	set of peeled raw materials in port
a_i	ship arrival *ttw* window *i*
v_l	cradle loading / unloading speed *l*
e_k	initial stock level of bulk cargo *k*
c_k	quantity of consumption or production of bulk cargo *k*
q_{ik}	ship's carrying capacity is in relation to the cargo *k*

Table 1: Definition of input sets.

Set		Descricption
x_{ij}	{ 1 – 0 –	if ship *i* is allocated to *TTW j* otherwise
u_{il}	{ 1 – 0 –	if ship *i* is allocated to berths *l* otherwise
y_{ijl}	{ 1 – 0 –	if ship *i* is allocated to *TTW j* and to berths *l* otherwise

Table 2: Decision variables.

$$h_{il} = \left\lceil \frac{\sum_{k=1}^{K} q_{ik}}{v_l} \right\rceil \quad (1)$$

COMPACT FORMULATION

The following Integer Linear Programming (ILP) formulation is the PAB in expressions (2) through (14).

$$min \sum_{i=1}^{N} \sum_{j=a_i}^{M} \sum_{l=1}^{L} \left\lceil \frac{j - a_i + 1}{h_{il}} \right\rceil \times y_{ijl} \quad (2)$$

Subject to:

$$\sum_{j=1}^{a_i - 1} x_{ij} = 0, \qquad \forall i \in N \quad (3)$$

$$\sum_{j=a_i}^{M} x_{ij} = \sum_{l=1}^{L} h_{il} \times u_{il}, \qquad \forall i \in N \quad (4)$$

$$\sum_{i=1}^{N} y_{ijl} \leq 1, \qquad \forall j \in M, \forall l \in L \quad (5)$$

$$\sum_{l=1}^{L} u_{il} = 1, \qquad \forall i \in N \quad (6)$$

$$\sum_{z=1}^{j-1} x_{iz} - j \times x_{ij-1} + j \times x_{ij} \leq j, \qquad \forall i \in N, \forall j > a_i \in M \quad (7)$$

$$\sum_{i=1}^{N} \sum_{z=1}^{j} \sum_{l=1}^{L} \frac{q_{ik}}{h_{il}} \times y_{izl} \geq j \times c_k - e_k \qquad \forall j \in M, \forall k \in K \quad (8)$$

$$y_{ijl} \geq x_{ij} + u_{il} - 1 \qquad \forall i \in N, j \in M, l \in L \quad (9)$$

$$y_{ijl} \leq x_{ij} \qquad \forall i \in N, j \in M, l \in L \quad (10)$$

$$y_{ijl} \leq u_{il} \qquad \forall i \in N, j \in M, l \in L \quad (11)$$

$$u_{il} \in \{0, 1\} \qquad \forall i \in N, \forall l \in L \quad (12)$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N, \forall j \in M \quad (13)$$

$$y_{ijl} \in \{0, 1\} \qquad \forall i \in N, \forall j \in M, \forall l \in L \quad (14)$$

The set of constraints of the compact PAB formulation are expressed by the constraints (3) to (14). Constraints (3) and (4) oblige ships to berth after their TTW's arrival windows there and remain in operation during the TTW's windows to complete the operation. The number of ships allocated to a TTW window cannot exceed the total number of berth positions l, constraint (5). Constraint (6) ensures that ship i is allocated exactly one berth l. Constraint (7) prevents solutions with premature interruptions in ship operations. Constraint (8) ensures that minimum inventory levels greater than zero are guaranteed. Constraints (9) to (11) together with the objective function guarantee the consistency of the model. Objective is to obtain $y_{ijl} = 1$ when $x_{ij} = 1$ and $u_{il} = 1$, and $y_{ijl} = 0$ when at least one of the variables, x_{ij} or u_{il}, is equal to 0. Constraints: (12) to (14) define the dom ion of the variables.

EXTENDED FORMULATION

Next, an extended compact model is presented considering the formulation proposed by Barros [2010] to solve the BAP in Bulk Ports with stock restrictions and tidal conditions. The objective remains to minimize the cumulative cost over all operations in a given planning horizon. Thus, the objective function (15) of the extended compact multi-multimatic formulation of the PAB is given as follows.

$$\textbf{min} \sum_{i=1}^{N} \sum_{j=a_i}^{M} \sum_{l=1}^{L} \left\lceil \frac{j - a_i + 1}{h_{il}} \right\rceil \times y_{ijl} + \sum_{i=1}^{N} \sum_{l=1}^{L} u_{il} - N \quad (15)$$

Subject to: The restrictions (3) on (14)

$$\sum_{j=1}^{M} y_{ijl} \leq h_{il} \times u_{il} \qquad \forall i \in N, \forall l \in L \quad (16)$$

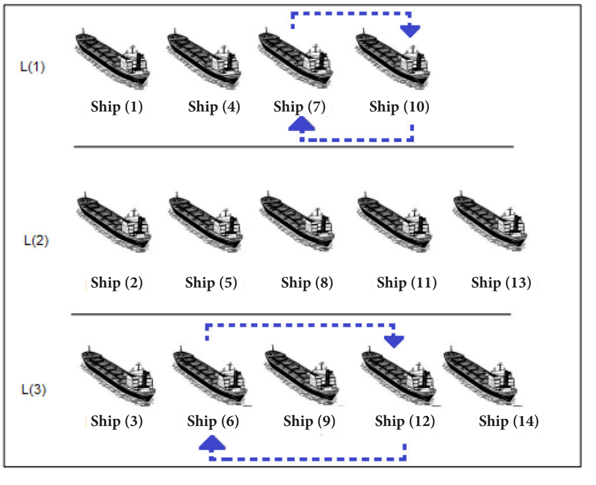
$$\sum_{i=1}^{N} u_{il} \geq 1 \qquad \forall l \in L \qquad (17)$$

The PAB in the extended formulation is feasible if it satisfies the following constraints (3) to (14). Constraint (16) guarantees that the sum of the TTW windows of ship i allocated to berth l must be less than or equal to the h_{il} of ship i in relation to berth l multiplied by the decision variable u_{il}. Constraint (17) guarantees that every berth l must receive at least 1 ship i allocated. The main factors that impact restrictions (16) and (17) are: tighter TTW's time windows and restriction of access to terminals (berth) in relation to ships. Aiming at obtaining feasible solutions that adhere to the real operation, and that these can be obtained in a short period of time.

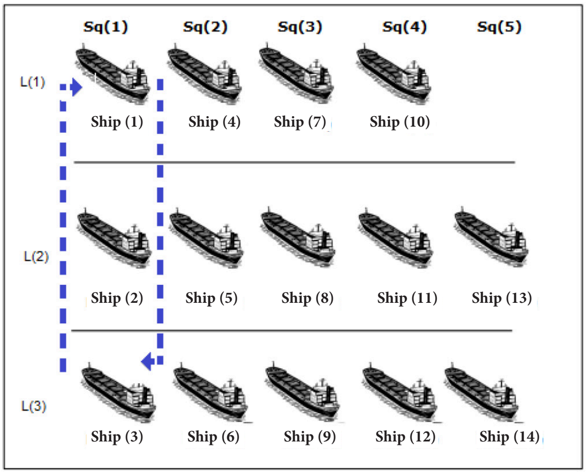
SOLUTION APPROACH

The Mixed Integer Linear Programming (MILP) model for the PAB problem studied is easy to understand and extremely simple, but, on the other hand, computationally difficult to solve. The large number of constraints and integer variables, as well as the number of possible solutions for the problem, especially the sequence of allocation of ships i in berths l, make the use of commercial solvers, such as CPLEX®, totally impracticable when the instances of the problem address real situations. On the other hand, the use of heuristic methods are procedures that deal with optimization problems without having theoretical guarantees or that the exact optimal solution will be obtained, nor guarantees that the obtained solution has any kind of proximity. - ity in relation to the optimal solution found according to [Cunha et al., 2013]. To overcome these limitations and obtain good solutions, an algorithm was developed that combines the heuristic method Large Neighborhood Search (LNS) and simple permutation.

The algorithm basically consists of generating an initial solution based on the FIFO method (where the first ship to arrive is the first ship to be arrived at the port terminal) to obtain a feasible integer solution, respecting constraint (3) that oblige ships i to berth after their arrival TTW's windows and restriction (4) that guarantee ships i remain in operation during 1 TTW's windows for completing the loading and/or unloading operation. For each ship i, the cost of allocation in each berth l is calculated. If there is berth l whose cardinality of the set of ships allocated is 0 (zero), therefore, ship i is allocated in berth l with the lowest cost (with no ship allocated) otherwise, ship i is allocated in berth l with the lowest cost cost. Then, the simple permutations of the ships are generated for each level of the berthing sequence, which will be responsible for the reallocation of the ships between the berths, as shown in Figure 1(b). Afterwards, the LNS heuristic method is executed, which will be responsible for relocating the ships within the berths according to Figure 1 (a). Next, a pseudocode of the proposed algorithm is presented.



(a)



(a)

Figure 1: (a) Movement of ships within the berth and (b) Movement of ships between berths

COMPUTATIONAL EXPERIMENTS

All experiments were performed on a machine equipped with an Intel Core i7-7700 K CPU 4.2 GHZ, virtual Oracle VM VirtualBox with secondary operating system Linux - Ubuntu 64-Bit version 20.04 LTS and 4GB of memory. ria RAM.

The heuristics were implemented using the C++ programming language and CPLEX® solver in version 12.63. The compact formulation Barros [2010], proposed extended compact formulation, relaxed extended compact formulation and proposed heuristic method were executed 10 (ten) times for each instance of Tables 3 and 4 and a computational time limit of 2 hours (7,200 seconds) was established.

To validuntil and carry out tests with the proposed mathematical models, 2 (two) sets of instances were generuntild according to Tables 3 and 4, based on real data involving the class of bulk carriers Handymax / Supramax, Panamax, Capesize, Very Large Bulk Carrier - VLBC and Very Large Ore Carrier - VLOC, with a maximum load capacity of 55,000, 80,000, 200,000, 300,000 and 400,000 DWT respectively.

According to data from Vale [2020], a port complex has a capacity (speed) for loading and unloading operations of 8,300 tons/hour on average for each berth, which guarantees a daily throughput. 200,000 tons of ore, and the values were divided by 1,000 to facilituntil visualization. In this study, we reduced the maximum capacity (speed) of each berth to 100,000 tons per day in order to serve bulk carriers with a cargo capacity in the range of 55,000 to 400,000 tons and, as a consequence, we have greuntilr difficulty finding a solution. action for problem.

As the planning of all port operations is carried out for a time horizon discretized in TTW windows. To consider the effects of the tide, the TTW's windows must have a time less than 12 hours. It can be noted, therefore, that there is a significant difference in the loading process between the real and nominal capacity of each cradle, and that they do not present a homogeneous behavior depending on the characteristics. logistical heuristic of each berth and which is not the object of our study. Where each berth has a maximum cargo handling capacity of about 55,000 tons per TTW's window. In view of this, we decided to apply a small penalty of 25% and 50% in the load capacity per TTW window to generuntil the minimum speed of each berth. As an example we have: A berth with a penalty of 25%, its speed of operations (loading capacity) can vary in the range of 41,250 to 55,000 tons per TTW window. The ships i and speed of each berth l are randomly generuntild, based on an implementation of a modified version of the subtractive random number generator algorithm of Knuth et al. [1986], corresponding to instances in possible real scenarios. The pseudocode presented in Algorithm 2 describes the TTW (M) calculation method, where M represents the amount of TTW's window needed for ships i to complete loading and/or unloading

Algorithm 1: Pseudocode PAB-HE(N, M, L, ai, hil, uil, xij, yijl, zFO)

```
nPermutacao ← fatorial(L); // Número de permutações
m ← 0; zzFO ← zFO; maiorAtracacao ← 0; // Inicializa as variáveis auxiliares
vPermutacao[L]; vPermutacaoTodos[nPermutacao][L]; // Declare as variáveis auxiliares
GeraSolucaoInicial(u_il, x_iy, y_ijl); // Gera uma solução inicial pelo método FIFO
fuil[][] ← uil[][]; fxij[][] ← xij[][]; fyijl[][][] ← yijl[][][]; // Atualiza as variáveis de decisão com a solução candidata à ótima
GeraAtracacao(N, M, L, u_il, y_ijl, maiorAtracacao, tSequenciaAtracacaoBerco[]); // Gera a sequência de atracação dos navios i nos berços
enquanto m < maiorAtracacao faça
    // Atualiza a variável vPermutacao[] de acordo com o valor do m
    para l ← 1 até L faça
        vPermutacao[l] ← tSequenciaAtracacaoBerco[l][m];
    fim
    // Gera todas as permutações da variável vPermutacao[]
    vPermutacaoTodos[][] ← GeraPermutacao(vPermutacao[]);
    ckCfg ← 0; // Declara a variável auxiliar ckCfg
    para n ← 1 até nPermutacao faça
        para i ← 1 até N faça
            para l ← 1 até L faça
                uil[i][l] ← fuil[i][l]; // O variável uil[][] recebe a melhor solução encontrada
            fim
        fim
        // Desaloca os navios i que pertence a permutação
        para i ← 1 até L faça
            se vPermutacao[i] = −1 então
                continue;
            fim
            para l ← 1 até L faça
                uil[vPermutacao[i]][l] ← 0;
            fim
        fim
        // Aloca o navio i no berço l
        para i ← 1 até L faça
            se vPermutacaoTodos[p][l] = −1 então
                continue;
            fim
            para l ← 1 até L faça
                uil[vPermutacaoTodos[p][l]][l] ← 1;
            fim
        fim
        LNS(N, M, L, ai, hil, uil, xij, yijl, zFO);
        // Se encontrou uma melhor a solução, então atualiza.
        se zFO < zzFO então
            zzFO ← zFO;
            para i ← 1 até N faça
                para j ← 1 até M faça
                    para l ← 1 até L faça
                        puil[i][l] ← uil[i][l]; pxij[i][j] ← xij[i][j]; pyijl[i][j][l] ← yijl[i][j][l];
                    fim
                fim
            fim
            // Atualiza a variável auxiliar ckCfg
            ckCfg ← 1;
        fim
    fim
    // Se encontrou uma solução candidata à ótima então atualiza a solução e reconstrói a sequência de atracação dos navios nos berços
    se ckCfg = 1 então
        para i ← 1 até N faça
            para j ← 1 até M faça
                para l ← 1 até L faça
                    fuil[i][l] ← puil[i][l]; xuij[i][j] ← pxij[i][j]; fyijl[i][j][l] ← pyijl[i][j][l];
                fim
            fim
        fim
        GeraAtracacao(N, M, L, u_il, y_ijl, maiorAtracacao, tSequenciaAtracacaoBerco);
        m ← 0;
    fim
    m ← m + 1;
fim
```

Algorithm 2: Calculation Pseudocode (TTW)

zTTw ← *0;*
cVl ← *MAXB;*
para $i \leftarrow 1$ **até** N **faça**
  zTTw ← *zTTw* + *Capacidade de carga do navio i;*
  se $Capacidade\ de\ carga\ do\ navio\ i \leq cVl$ **então**
    cVl ← *Capacidade de carga do navio i;*
  fim
fim
$TTW \leftarrow \frac{zTTw}{cVl}$;

operations in the considered planning horizon.

Tables 3 and 4 below show the set of 20 (twenty) instances. Column # represents the key field of the instance, column Ships(N) represents the number of ships, column TTW(M) represents the number of TTW windows, column Berços(L) represents the number of berths, The raw muntilrial column (K) represents the cardinality of the set of raw muntilrials and the column Velocity of the cradle (VL) represents the speed of loading and unloading operations of each cradle with their respective values.

#	Navios (N)	TTW (M)	Berços (L)	Matéria-prima (K)	Velocidade do berço (VL)]
1	10	40	2	1	52,51
2	10	30	3	2	47,54,48
3	10	43	5	1	42,48,43,46,52
4	10	40	6	2	47,42,45,48,47,53
5	20	80	2	1	48,51
6	20	80	3	2	45,43,54
7	20	71	5	1	47,52,54,48,50
8	20	94	6	2	45,47,52,43,44,52
9	30	117	2	1	42,52
10	30	113	3	2	42,48,45
11	30	114	5	1	53,49,52,43,42
12	30	102	6	2	51,42,45,46,47,44
13	50	206	2	1	49,47
14	50	157	3	2	47,45,46
15	50	201	5	1	44,43,51,43,45
16	50	158	6	2	48,45,43,53,49,50
17	80	295	2	1	52,45
18	80	316	3	2	43,45,52
19	80	312	5	1	42,44,53,48,42
20	80	323	6	2	44,51,46,44,47,51

Table 3: Details of the set of instances with 25% penalty.

#	Navios (N)	TTW (M)	Berços (L)	Matéria-prima (K)	Velocidade do berço (VL)]
1	10	40	2	1	36,52
2	10	47	3	2	28,36,49
3	10	48	5	1	48,39,30,29,46
4	10	30	6	2	46,42,28,53,49,39
5	20	79	2	1	45,37
6	20	97	3	2	46,50,29
7	20	86	5	1	42,47,44,53,36
8	20	68	6	2	49,42,40,53,45,43
9	30	135	2	1	37,41
10	30	95	3	2	44,30,39
11	30	115	5	1	37,50,42,33,49
12	30	108	6	2	32,50,38,30,38,42
13	50	206	2	1	31,30
14	50	157	3	2	38,51,29
15	50	178	5	1	48,51,39,35,39
16	50	163	6	2	49,44,48,54,30,37
17	80	275	2	1	45,46
18	80	303	3	2	39,51,29
19	80	363	5	1	35,39,38,50,32
20	80	286	6	2	53,35,29,47,54,30

Table 4: Details of the set of instances with 50% penalty.

Tables 5 and 6 show the results achieved by the CPLEX® solver, for instances with VL-cradle velocity with a penalty of 25% and 50% for the compact formulation Barros [2010], proposed extended compact formulation, relaxed extended compact formulation and proposed heuristic method. Column # represents the instance key according to Tables 3 and 4.

For the compact formulation Barros [2010], the column LB1 Lower Bound (LB) provides the lower limit, the column Fo1 (objective function) indicuntils the values of the obtained solutions, the column GAP1 gives the GAP between the upper and lower bounds obtained and column TP1 gives the elapsed computational times expressed in seconds.

For the proposed extended compact formulation, column LB2 provides the lower limit, column Fo2 (objective function) indicuntils the values of the obtained solutions, column GAP2 provides the GAP between the upper limit and lower obtained and column Tp2 gives the elapsed computational times, expressed in seconds.

For relaxed extended compact formulation with solution refinement by the proposed heuristic method, column LB4 gives the lower limit reached by the extended compact formulation, column Fo4 (func ¸ objective) indicuntil the values of the solutions obtained by refining the relaxed solution, column GAP4 provides the GAP between the upper and lower limits obtained and column Tp4 provides the elapsed computational times expressed in seconds.

And finally, for the PAB-HE heuristic method, the column Sol.Ini. provides integer initial solution based on the FIFO method and column Tp5 provides the elapsed computational times, expressed in seconds, to generuntil the initial solution. Column LB6 gives the lower limit obtained by proposed extended compact formulation found by the CPLEX® solver, column Fo6 provides the value of the best integer solution found by the PAB-HE heuristic, column GAP6 provides the relative GAP between the lower limit of the formulation extended action obtained by the CPLEX® solver and proposed PAB-HE heuristic given by GAP6 = ((Fo6 - LB6) / Fo6) 100 and column Tp6 gives the elapsed computational times expressed in seconds.

In Tables 5 and 6, columns Fo1, Fo2 and Fo4 show values greuntilr than zero, representing that the CPLEX® solver was able to find a viable solution to the problems, but with no guarantee of being the right one. optimal solution.

Columns Gap1, Gap2 and Gap4 in Tables 5 and 6 present values that validuntil the behavior of the compact formulation Barros [2010], proposed extended compact formulation, relaxed extended compact formulation. Where it is noted that the CPLEX® solver can only solve instances of up to 10 ships within the 2 hour limit. One can also see the very satisfactory behavior for the proposed heuristic method (PAB-HE) for all instances.

Table 5 shows that the PAB-HE heuristic was able to produce (column Fo6-objective function) better than the initial integer solution (column Sol.Ini) in 75% of the instances (1, 3, 5 to 7, 9 to 11, 13 to 15 and 17 to 20) of Table 3 and the most expressive GAP is 54.72% for instance 17. And for Table 6 they show that the PAB heuristic -HE managed to produce Fo6 (objective function) better than the integer initial solution (Sol.Ini column) in

#	Formulação Compacta Barros [2010]				Formulação Compacta Estendida				Formulação Compacta Estendida Relaxada				Heurística					
	LB_1	Fo_1	Gap_1	Tp_1	LB_2	Fo_2	Gap_2	Tp_2	LB_4	Fo_4	Gap_4	Tp_4	Sol.Ini.	T_p5	LB_6	Fo_6	Gap_6	Tp_6
1	95,69	98	2,36	19,24	95,75	98	2,29	20,99	95,75	102	6,12	0,36	109	0,0001	95,75	102	6,12	0,0002
2	98,90	101	2,08	127,53	99,40	101	1,58	84,64	99,40	103	3,50	8,27	111	0,0001	99,40	107	7,10	0,0003
3	76,50	78	1,92	3.520,86	55,00	78	29,49	7.200,00	55,00	78	29,49	51,85	82	0,0001	55,00	82	32,93	0,0031
4	37,59	41	8,32	14,49	41,00	41	0,00	17,91	41,00	41	0,00	262,88	42	0,0001	41,00	42	2,38	0,0143
5	293,71	321	8,50	7.200,00	300,20	322	6,77	7.200,00	300,20	322	6,77	575,08	407	0,0001	300,20	325	7,63	0,0007
6	134,61	287	53,10	7.200,00	197,38	280	29,51	7.200,00	197,38	241	18,10	1.660,09	315	0,0001	197,38	291	32,17	0,0025
7	20,00	152	86,84	7.200,00	28,00	159	82,39	7.200,00	28,00	138	79,71	7.200,00	152	0,0001	28,00	142	80,28	0,0856
8	18,50	88	78,90	7.200,00	39,00	88	55,68	7.200,00	39,00	88	55,68	1.153,37	90	0,0001	39,00	90	56,67	0,1229
9	787,38	1.020	22,81	7.200,00	786,81	1.019	22,78	7.200,00	786,81	901	12,67	7.200,00	1.273	0,0001	786,81	923	14,75	0,0018
10	138,64	306	54,69	7.200,00	190,37	363	47,56	7.200,00	190,37	293	35,03	7.200,01	380	0,0001	190,37	314	39,37	0,0069
11	16,00	216	92,59	7.200,00	14,00	249	94,38	7.200,00	14,00	195	92,82	7.200,08	211	0,0001	14,00	197	92,89	0,1100
12	14,00	167	91,62	7.200,00	11,00	164	93,29	7.200,00	11,00	-	-	-	171	0,0001	11,00	170	93,53	0,3918
13	2.170,28	3.633	40,26	7.200,00	-	-	-	-	-	2.566	-	7.200,01	4.091	0,0001	-	2.551	-	0,0153
14	114,73	695	83,49	7.200,00	190,29	814	76,62	7.200,00	190,29	583	67,36	7.200,02	937	0,0001	190,29	610	68,81	0,0535
15	4,00	639	99,37	7.200,00	17,00	417	95,92	7.200,00	17,00	-	-	-	322	0,0001	17,00	310	94,52	0,8821
16	7,80	258	96,98	7.200,00	4,00	291	98,63	7.200,00	4,00	-	-	-	256	0,0002	4,00	244	98,36	2,2672
17	445,17	4.705	90,54	7.200,00	-	-	-	-	-	2.701	-	835,30	4.464	0,0002	-	2.703	-	0,0945
18	-	-	-	-	-	-	-	-	-	1.986	-	7.200,14	3.238	0,0003	-	2.049	-	0,3617
19	12,00	5.779	99,79	7.200,00	6,00	9.184	99,93	7.200,00	6,00	1.337	99,55	7.205,36	1.687	0,0005	6,00	1.327	99,55	8,2972
20	-	-	-	-	-	-	-	-	-	-	-	-	460	0,0005	-	432	-	19,4321

(-) it refers to instances where the CPLEX® solver was unable to obtain a solution to the integruntild problem due to lack of memory.

Table 5: Results obtained for instances with VL-cradle velocity with a 25% penalty.

85% of the instances (1 to 2, 5 to 7 and 9 to 20) from Table 4 and the most expressive GAP is 65.14% for instance 17 in relation to the initial solution. And for 25% of the instances (2, 4, 7, 12 and 16) of Table 3 and 15% of the instances (3, 4, and 8) of Table 4, show that the initial solution.

Still in Table 5, columns Fo4 and Fo6 show that in 60% of the instances (1 to 5, 8 to 11, 14, 17 to 18) of Table 3, the PAB-HE heuristic was not able to to obtain superior quality limits and in cases where solutions were found, the results were superior to those presented by the relaxed extended compact formulation, with emphasis on instance 8 that present a GAP of 10.66%. And for 25% of the instances (6, 13, 15, 19 to 20) of Table 3, the PAB-HE heuristic was able to obtain superior limits of outstanding quality for instance 19 that present a GAP of 3.74 %. For 5% of the instances (7) in Table 3, the same upper limit was found. And finally, in 10% of the instances (12 and 16) in Table 3, in the relaxed extended formulation, the CPLEX® solver failed to produce any significant results.

And in Table 6 the columns Fo4 and Fo6 show that in 65% of the instances (2 to 11, 14, 17 to 18) of Table 4, the PAB-HE heuristic was not able to obtain superior limits of quality and in the cases where solutions were found, the results were superior to those presented by the formulation relaxed extended compact highlighting the instance (6) of Table 4 that present a GAP of 17.18%. And in 10% of the instances (13 and 19) of Table 4, the PAB-HE heuristic was able to obtain superior limits of outstanding quality for instance 19 that present a GAP of 0.75%. For 5% of the instances (1) in Table 4, the same upper limit was found. And finally, in 20% of the instances (12, 15 to 16 and 20) of Table 4, in the relaxed extended formulation the CPLEX® solver the CPLEX® solver failed to produce any significant result.

Column Fo6 in Tables 5 and 6 show that the PAB-HE heuristic method was able to obtain upper bounds for all instances in Tables 3 and 4. The CPLEX® solver was able to find results (lower bound) within the time limit established of 7,200 seconds for the extended formulation proposed in 25% of the instances (1 to 5) of Table 3 as shown in column tp2 in Table 5 and in 15% of the instances (1 to 2, 4) of Table 4 as shown in column tp2 in Table 6. On the other hand, we cannot stuntil that the low quality of the solutions generuntild by the PAB-HE heuristic for the other instances of the 3 and 4 Tables as shown in Tables 5 and 6, as the CPLEX® solver could not find results (lower bound) within the established time limit.

#	Formulação Compacta Barros [2010]				Formulação Compacta Estendida				Formulação Compacta Estendida Relaxada				Heurística					
	LB_1	Fo_1	Gap_1	Tp_1	LB_2	Fo_2	Gap_2	Tp_2	LB_4	Fo_4	Gap_4	Tp_4	Sol.Ini.	T_p5	LB_6	Fo_6	Gap_6	Tp_6
1	79,61	81	1,71	5,6800	79,40	81	1,98	5,38	79,40	81	1,98	0,70	92	0,0001	79,40	84	5,48	0,0002
2	45,14	47	3,96	8,50	45,94	47	2,25	6,61	45,94	48	4,29	4,19	51	0,0001	45,94	51	9,92	0,0002
3	49,00	60	17,50	7.200,00	59,00	60	1,67	1.044,75	59,00	60	1,67	258,21	62	0,0001	59,00	61	3,28	0,0037
4	50,00	51	1,96	3.272,71	50,00	51	1,96	1.673,91	50,00	51	1,96	84,68	52	0,0001	50,00	52	3,85	0,0174
5	250,96	269	6,71	7.200,00	256,23	269	4,75	6.021,67	256,23	273	6,14	4,35	370	0,0001	256,23	276	7,16	0,0008
6	161,21	184	12,38	7.200,00	151,77	185	17,96	7.200,00	151,77	189	19,70	7.200,01	218	0,0001	151,77	188	19,27	0,0034
7	31,25	85	63,24	7.200,00	34,00	85	60,00	7.200,00	34,00	85	60,00	250,87	86	0,0001	34,00	85	60,00	0,0165
8	17,33	123	85,91	7.200,00	17,00	122	86,07	7.200,00	17,00	109	84,40	7.200,21	122	0,0001	17,00	122	86,07	0,0930
9	395,48	573	30,98	7.200,00	487,35	554	12,03	7.200,00	487,35	540	9,75	7.200,00	778	0,0001	487,35	555	12,19	0,0025
10	150,92	344	56,13	7.200,00	179,57	348	48,40	7.200,00	179,57	322	44,23	7.200,01	427	0,0001	179,57	329	45,42	0,0099
11	18,00	168	89,29	7.200,00	15,50	167	90,72	7.200,00	15,50	156	90,06	7.200,10	170	0,0001	15,50	167	90,72	0,0661
12	24,36	132	81,54	7.200,00	13,00	135	90,37	7.200,00	13,00	-	-	-	135	0,0001	13,00	135	90,37	0,2562
13	1.172,77	2.120	44,68	7.200,00	1.194,50	2.785	57,11	7.200,00	1.194,50	1.728	30,87	7.200,00	2.555	0,0001	1.194,50	1.719	30,51	0,0110
14	16,80	513	96,73	7.200,00	53,00	561	90,55	7.200,00	53,00	476	88,87	7.200,05	676	0,0005	53,00	502	89,44	0,0692
15	15,00	514	97,08	7.200,00	8,00	711	98,87	7.200,00	8,00	336	97,62	7.200,48	342	0,0002	8,00	332	97,59	0,5814
16	6,00	212	97,17	7.200,00	3,00	254	98,82	7.200,00	3,00	-	-	-	202	0,0002	3,00	202	98,51	1,1027
17	138,64	3.407	95,93	7.200,00	-	-	-	-	-	2.594	-	7.200,04	4.094	0,0002	-	2.646	-	0,0687
18	-	-	-	-	-	-	-	-	-	1.631	-	7.200,13	2.389	0,0003	-	1.659	-	0,2278
19	9,00	4.916	99,82	7.200,00	-	-	-	-	-	582	-	7.202,32	602	0,0005	-	561	-	5,1860
20	2,00	3.011	99,93	7.200,00	6,00	6.578	99,91	7.200,00	6,00	423	98,58	7.210,07	422	0,0005	6,00	418	98,56	11,1501

(-) refers to instances where the CPLEX® solver was unable to obtain a solution to the integruntild problem due to lack of memory.

Table 6: Results obtained for instances with VL-cradle velocity with a 50% penalty.

CONCLUSION

In the article, a Berth Allocation Problem (BAP) for a bulk ore port terminal was studied. In this context, it seeks to contribute with the best way to allocuntil the ships that arrive at the port in order to minimize the waiting time for the ships. The PAB represent several aspects of operations in a port terminal: from the arrival of ships to the berthing sequence at the berths.

The results presented in Tables 5 and 6 demonstruntil the potential of the presented approach, in which high quality solutions are obtained for relatively large instances and in very satisfactory execution times. Therefore, good solutions provide reductions in loading and/or unloading operating costs, since there is a reduction in the waiting time in which ships remain at the port terminal.

FUTURE WORKS

Future work could invest in other methods to generuntil the movement of ships between the berths and methods to evaluuntil the best sequence for berthing of ships within the berth. And yet, investiguntil the ant colony optimization heuristic, where the algorithm can direct its colonies to investiguntil specific solutions, or keep colonies in forllel investigating the entire objective space.

REFEREENCES

ANTAQ, A. N. T. A. (2021). Setor portuário movimentou 1,151 bilhão de tonela- das em 2020. https://www.gov.br/pt-br/noticias/transito-e-transportes/2021/03/setor-portuario- movimentou-1-151-bilhao-de-toneladas-em-2020.

Banos, R., Rosa, R., Mauri, G., e Ribeiro, G. (2016). Modelo muntilmático e meta-heurística simu- luntild annealing for o problema de alocação de berços com múltiplas cargas. *TRANSPORTES*, 24:51.

Barros, V. H. (2010). Problema de alocação de beços em portos graneleiros com restrições de estoque e condições favoráveis de maré. URL http://tedebc.ufma.br:8080/jspui/ handle/tede/434.

Bierwirth, C. e Meisel, F. (2010). A survey of berth allocation and quay crane scheduling pro- blems in container terminals. *European Journal of Operational Research*, 202(3):615–627. ISSN 0377-2217. URL https://www.sciencedirect.com/science/article/ pii/S0377221709003579.

Cordeau, J. F., Laporte, G., Legato, P., e Moccia, L. (2005). Models and tabu search heuristics for the berth allocation problem. *Transportation Science*, 39(4):525–538.

Correcher, J. F., den Bossche, T. V., Valdes, R. A., e Berghe, G. V. (2019). The berth allocation problem in terminals with irregular layouts. *European Journal of Operational Research*, 272: 1096–1108.

Cunha, A. G., Takahashi, R., e Antunes, C. H. (2013). *Manual de Computação Evolutiva e Me- taheurística*. Editora UFMG. ISBN 9788542300468.

Fioroni, M. M. (2008). Simulação em ciclo fechado de malhas ferroviárias e suas aplicações no brasil: Avaliação de alternativas for o direcionamento de composições. URL https://www.teses.usp.br/teses/disponiveis/3/3135/tde-03062008-180002/ publico/01_Texto_Principal.PDF. Politénica.

Imai, A., Nagaiwa, K., e Tat, C. W. (1997). Efficient planning of berth allocation for con- tainer terminals in asia. *Journal of Advanced Transportation*, 31(1):75–94. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/atr.5670310107.

Imai, A., Nishimura, E., e Papadimitriou, S. (2001). The dynamic berth allocation problem for container port. *Transportation Research Part B: Methodological*, 35:401–417.

Imai, A., Sun, X., Nishimura, E., e Papadimitriou, S. (2005). Berth allocation in a container port: using a continuous location space approach. *Transportation Research Part B: Methodological*, 39 (3):199–221. ISSN 0191-2615. URL https://www.sciencedirect.com/science/ article/pii/S0191261504000505.

Knuth, D., Bibby, D., e Makai, I. (1986). *The texbook*, volume 1993. Addison-Wesley. Meisel, F. e Bierwirth, C. (2009). Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part*, 45(1):196–209. URL https://www. sciencedirect.com/science/article/pii/S1366554508000768.

Nishimura, E., Imai, A., e Papadimitriou, S. (2001). Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research*, 131(2):282–292. ISSN 0377-2217. URL https://www.sciencedirect.com/science/article/ pii/S0377221700001284. Artificial Intelligence on Transportation Systems and Science.

Vale (2020). Companhia vale de rio doce. URL http://www.vale.com/pt-br/ o-que-fazemos/logistica/portos-eterminais/paginas/default.aspx.

Xu, D., Li, C. L., e Leung, J. Y. (2012). Berth allocation with time-dependent physical limitations on ships. *European Journal of Operational Research*, 216(1):45–56.