



Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem

Eduardo Lalla-Ruiz, Belén Melián-Batista, J. Marcos Moreno-Vega*

Departamento de Estadística, I.O. y Computación Escuela Técnica Superior de Ingeniería Informática Universidad de La Laguna 38271, La Laguna, Santa Cruz de Tenerife

ARTICLE INFO

Article history:

Received 25 May 2011

Received in revised form

1 June 2012

Accepted 2 June 2012

Available online 23 June 2012

Keywords:

Artificial intelligence

Port logistics

Berth allocation

Tabu search

Path relinking

ABSTRACT

This paper considers the Dynamic Berth Allocation Problem, in which vessels are assigned to discrete positions in berths. This problem, whose goal is to minimize the total time the vessels stay at the port, constitutes one of the most important processes at any containers terminal. We propose a hybrid metaheuristic that combines Tabu Search with Path Relinking, $T^2S^* + PR$. The results reached by this hybrid algorithm are compared with the optimal values given by the best mathematical model that appears in the literature for this problem, $GSPP$, and with a tabu search algorithm from the literature, T^2S . For small instances, the algorithm $T^2S^* + PR$ is able to obtain most of the optimal solutions in an amount of computational time that is lower than the time required to solve the $GSPP$ model. For medium and large size instances, $GSPP$ cannot be solved to optimality, whereas the proposed hybrid algorithm outperforms T^2S . Moreover, the computational experiments carried out in this paper confirm the robustness of the proposed algorithm with respect to both the parameters governing the procedure and the problem size.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

According to the data provided by the [Eno Transportation Foundation](http://www.eno.org), more than 60% of the products that travel over the sea are stored in containers. This percentage rise up to almost 100% over some routes connecting economically strong stable countries. The total amount of cargo reaches several millions of TEUs (measure unit equivalent to the dimensions of a standard container). Ports compete to become interchange points (hubs) or origin–destination points of the transportation routes. Regional and national governments consider a strategic objective to have ports where to locate container terminals since they are sources of economic growth. The competitiveness in this area is huge. The main criteria used by the operators to choose a port as operations base are the geographic location, politic and social stability, as well as operational costs.

Broadly speaking, the loading and unloading processes in a container terminal consist of several phases as indicated in [Stahlbock and Voß \(2008\)](#), [Steenken et al. \(2004\)](#), [Vis and de Koster \(2003\)](#). Once a vessel reaches the port, it is allocated to a berth. Then, its containers are unloaded and driven to the yard, where they are temporally stored. Finally, they are moved to

trucks, trains or ships to proceed with the delivery phase. Complex planning and management problems, which are very important for the transport operators and port authorities, arise in each of the previous phases. Their effective resolution drives to important savings, being the berthing one of the tasks with the highest impact in the final costs.

Given a set of vessels, the goal of a berth allocation problem is to determine how to allocate them to the berths in order to optimize some cost function. The features of the vessels and the port or its resources determine the constraints that have to be satisfied to obtain feasible solutions. One of the most used cost functions to measure the efficiency of a port terminal is the sum of the waiting and working times of each vessel. Other cost functions are the workload of terminal resources ([Lim, 1998](#)) and the number of vessels which cannot be attended in it ([Imai et al., 2008](#)).

[Bierwirth and Meisel \(2010\)](#) describe different integration schemes. In a functional integration by a feedback loop, the output solution of one problem is given as input to the other problem in a loop that is executed until a solution satisfying the decision maker expectations is reached. Therefore, it is quite relevant to design and develop procedures that provide high quality solutions (no necessarily optimal) with a low computational cost. An example of these designs can be seen in [Giallombardo et al. \(2010\)](#), where a two levels heuristic is proposed to solve the joint problem of allocating vessels and assigning quay cranes to them. In the first level, a quay crane

* Corresponding author.

E-mail addresses: elalla@ull.es (E. Lalla-Ruiz), mbmelian@ull.es (B. Melián-Batista), jmmoreno@ull.es (J. Marcos Moreno-Vega).

profile is assigned to each vessel and in the second level, the corresponding berth allocation problem is solved. These steps are then repeated until a stopping condition is met.

In this work, we propose a hybrid heuristic that combines the Tabu Search metaheuristic with Path Relinking (T^2S^*+PR) (artificial intelligence methods) to solve the Dynamic Berth Allocation Problem. This procedure incorporates a Tabu Search algorithm based on T^2S , proposed by Cordeau et al. (2005), a set of elite solutions which consists of a subset of all the local optima found along the tabu search process. This set is then used to build new starting solutions for the tabu search by executing a path relinking algorithm. Moreover, we include an additional neighborhood structure that allows swaps of vessels both among berths and in the same berth. Therefore, we obtain a tabu search procedure that uses two neighborhood structures to guide the search, T^2S^* . The results obtained by the proposed hybrid algorithm are then compared with the results given by the exact resolution of the mathematical model $GSPP$ (Generalized Set Partitioning Problem) presented by Buhrkal et al. (2011) and with the results reached by the T^2S proposed by Cordeau et al. (2005). The computational experience corroborates that T^2S^*+PR outperforms T^2S . It obtains optimal or near-optimal solutions in most cases in a smaller amount of time than $GSPP$. Moreover, for medium and large size instances, for which $GSPP$ cannot be solved to optimality, T^2S^*+PR significantly outperforms T^2S .

Note that the tabu search algorithm used in this paper is based on the tabu search T^2S , proposed by Cordeau et al. (2005) to solve the Discrete Dynamic Berth Allocation Problem (DBAP), but that it is not exactly the one proposed by these authors. The main difference between both tabu search algorithms is the fact that our algorithm uses two different neighborhood structures, instead of the single one used by Cordeau et al. Since several neighborhood structures are considered, it is also required to define the pattern that has to be followed to perform the different moves. This constitutes one of the contributions of our paper and is explained in Section 4. Moreover, our method combines tabu search with path relinking. It is also important to notice the fact that it is not guaranteed that a hybrid method combining several heuristics will reach better solutions than its single parts. In this work we obtain a hybrid algorithm that outperforms T^2S considering both the solution quality and the computational times.

In the following, we present the outline of the paper and highlight its contributions. The literature review of the dynamic berth allocation problem as well as its description are provided in Section 2. The paper contribution relies upon the solution algorithm introduced in Section 2 that combines a tabu search algorithm, which makes use of two different neighborhood structures, with path relinking, so providing a hybrid algorithm. The computational experience carried out in this paper is summarized in Section 4. This experience confirms that the proposed method is effective since it reaches the optimal solution for most of the solved instances. Furthermore, the algorithm is significantly more efficient than the resolution of the mathematical model $GSPP$ studied in Buhrkal et al. (2011). Therefore, it can be used as an efficient method for solving the berth allocation problem in integrated designs as the one proposed by Giallombardo et al. (2010). Finally, we draw our conclusions in Section 5.

2. Berth allocation problem

In many multi-users container terminals, the quay is divided into a finite set of berths to which the vessels can be assigned for loading and unloading purposes. The problem associated to this assignment scheme is referred to as Discrete Berth Allocation

Problem. The goal of this problem is to minimize the total time that vessels stay at the port.

In the static version of the problem, all vessels arrive at the terminal before the starting planning time, while the dynamic version takes into account the vessels that arrive at any time of the planning horizon. The static problem has been studied in Imai et al. (1997, 2001, 2008) and Lee and Chen (2009). The dynamic problem has been studied in Imai et al. (2001, 2003, 2007, 2008), Cordeau et al. (2005), Monaco and Sammarra (2007), Hansen et al. (2008), Silva et al. (2008), Liang et al. (2009) and Giallombardo et al. (2010).

Imai et al. (2001) propose a heuristic based on the Lagrangian relaxation of the proposed model and develop an extensive computational experience that corroborates the effectiveness of the proposed heuristic in real applications. Imai et al. (2003) consider a variant of the problem that includes service priority associated to the vessels. These priorities let distinguishing the vessels according to their size, work volume, etc. The resulting model is then solved using a Genetic Algorithm. Cordeau et al. (2005) propose a new formulation that includes the weighted sum of the service times and time windows of the berths. They also develop a Tabu Search to solve the problem. Imai et al. (2007) tackle the berth allocation problem in which the vessels can be loaded and unloaded from both sides of the berth. The problem is solved by using a Genetic Algorithm. Monaco and Sammarra (2007) propose a more compact mathematical formulation of the problem, which is then solved using a Lagrangian heuristic. The joint problem of berth allocation and quay cranes assignment is introduced by Imai et al. (2008), who solve it by means of a Genetic Algorithm. Liang et al. (2009) also consider the previous joint problem. In their model, the position and time for berthing and the number of quay cranes must be determined in order to minimize the amount of working time, waiting time and delay of each vessel. Hansen et al. (2008) propose a Variable Neighborhood Search for a variant of this problem in which each vessel has both a reward and a penalty for finishing earlier or later the preestablished time, respectively. Silva et al. (2008) propose a heuristic algorithm based on Genetic Algorithm to solve a particular real case in a Brazilian port. Giallombardo et al. (2010) propose a heuristic that combines Tabu Search and Mathematical Programming to solve a new model of berth allocation and quay cranes assignment.

The Discrete Dynamic Berth Allocation Problem was first formulated as a mixed integer programme by Imai et al. (2001) as an extension of the formulation proposed by Imai et al. (1997) for the discrete static berth allocation problem. Alternative formulations for the dynamic problem have been proposed and studied by Monaco and Sammarra (2007), Cordeau et al. (2005) and Buhrkal et al. (2011). These models are described and compared in Buhrkal et al. (2011). The main conclusion is that the model by Christensen and Holst is superior to the other models, since it is able to reach the optimal solutions for the set of instances used by all these authors.

We report an example of the berth allocation problem. Fig. 1 shows feasible solutions for the static and dynamic berth allocation problems, for which the working times (C_{ij}), berths availability (S_i) and arrival times (A_j) of the vessels are summarized in Table 1. Note that in the dynamic version of the problem, which is the version considered in this paper, idle times may appear in the berths planning. Note that in both cases, static and dynamic, berths 1 and 2 are opened in times 4 and 3, respectively. In the static case, all the vessels arrive at the port before the opening time of the berths, while in the dynamic case, the vessels can arrive at any time along the planning horizon.

Let us focus on the dynamic part of this example, since the problem tackled in this paper is the dynamic one. In this case, vessels v_4 , v_1 and v_2 are assigned to berth 1, while vessels v_3 and

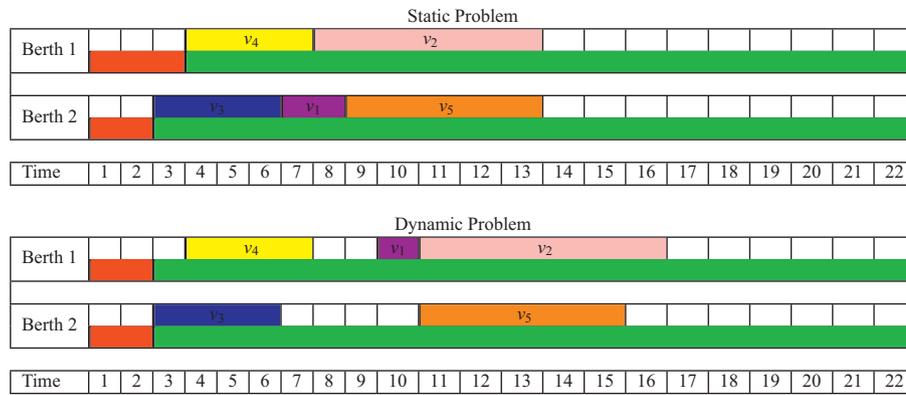


Fig. 1. Berth allocation problems.

Table 1
Vessels arrival times, berths availability and working times.

Vessels arrival times		Berths availability		Working times	
<i>Static problem</i>					
j	1 2 3 4 5	i	1 2	C_{ij}	j 1 2 3 4 5
A_j	1 2 2 1 1	S_i	4 3		i 1 6 4 4 6
					2 5 4 3 5
<i>Dynamic problem</i>					
j	1 2 3 4 5	i	1 2	C_{ij}	j 1 2 3 4 5
A_j	10 6 3 4 11	S_i	4 3		i 1 6 4 4 6
					2 5 4 3 5

v_5 are assigned to berth 2. Vessel v_4 , which arrives at time 4, is assigned to berth 1 at this time and it stays for 4 time units as indicated in Table 1. The next vessel assigned to berth 1 is v_1 , that arrives at time 10 and stays 1 time unit. Vessel v_2 is then assigned to berth 1 after vessel v_1 has finished. Note that, since vessel v_2 arrives at time 6 and is assigned to berth 1 at time 11, it has to wait for 5 time units. Then, it requires six more time units to perform the loading and unloading operations. Therefore, the total time vessel v_2 stays at the port is equal to 11 time units, which are counted in the objective function value. The same analysis can be considered to determine the stays of vessels v_3 and v_5 at the port. Therefore, in this example, vessels v_1 to v_5 stay at the port for 1, 11, 4, 4 and 5 time units, respectively and the objective function value is equal to 25 time units.

Finally, in order to make this paper self-contained, we describe the mathematical formulation presented in Buhrkal et al. (2011). According to these authors, the dynamic berth allocation problem can be modeled as a Generalized Set Partitioning Problem GSPP. In the constraint matrix, a column represents a feasible assignment of a ship to a berth at a time. The set of columns is denoted by $|\Omega|$. Two matrices A and B, both containing $|\Omega|$ columns are defined. Matrix $A = (A_{i\omega})$ contains a row for each ship, and $A_{i\omega} = 1$, if and only if column ω represents an assignment of ship $i \in N$. Each column of A contains exactly one non-zero element. Matrix $B = (B_{p\omega})$ contains a row per (berth,time) position. The rows of B are indexed by the set P, with $|P| = \sum_{k \in M} (e^k - s^k)$. The entry $B_{p\omega}$ is equal to 1, if and only if, position $p \in P$ is contained in the assignment that column ω represents. The cost c_ω of any column $\omega \in \Omega$ is the service time of the respective position assignment and can be multiplied by the priority factor v_i if necessary. A binary variable x_ω is equal to 1, if column ω is used in the solution, and 0 otherwise. With these definitions the GSPP formulation of the BAP presented in Buhrkal et al. (2011) is stated as follows:

$$\min \sum_{w \in \Omega} c_w x_w \tag{1}$$

subject to

$$\sum_{w \in \Omega} A_{iw} x_w = 1, \quad \forall i \in N \tag{2}$$

$$\sum_{w \in \Omega} B_{pw} x_w \leq 1, \quad \forall p \in P \tag{3}$$

$$x_w \in \{0, 1\}, \quad \forall w \in \Omega \tag{4}$$

The objective function (1) minimizes the service time of the vessels. The set of constraints (2) ensures that all vessels are served. Finally, the constraints (3) guarantee that at a time interval, in a berth, only one vessel can be served.

3. Tabu search with path relinking

Tabu Search (TS) (Glover and Laguna, 1993) is a metaheuristic procedure whose philosophy is to derive and exploit a collection of intelligent problem solving strategies, based on implicit and explicit learning procedures to exploit an adaptive memory framework, which is an important feature of the artificial intelligence. From the standpoint of tabu search, adaptive memory embodies the dual functions of creating and exploiting memory structures, H , for taking advantage of the history of the problem-solving process. The memory structures of tabu search operate by reference to four principal dimensions, consisting of recency, frequency, quality, and influence.

TS may be conveniently characterized by reference to neighborhood search, though neighborhood search has a broader meaning in TS than in some other parts of the metaheuristic literature. Tabu search employs a different philosophy for going beyond the criterion of terminating at a local optimum. Randomization is deemphasized, and generally is employed only in a highly constrained way, on the assumption that intelligent search should be based on more systematic forms of guidance. The notion of exploiting certain forms of adaptive memory to control the search process is the central theme underlying tabu search. The effect of such memory may be envisioned by stipulating that TS maintains a selective history H of the states encountered during the search, and replaces the neighborhood of solution s , $N(s)$, by a modified neighborhood which may be denoted by $N(H,s)$. History therefore determines which solutions may be reached by a move from the current solution, selecting s' from $N(H,s)$.

In TS strategies based on short-term considerations, $N(H,s)$ characteristically is a subset of $N(s)$, and the tabu classification serves to identify elements of $N(s)$ that will be excluded. In the intermediate and longer term strategies, $N(H,s)$ may contain solutions not in $N(s)$, generally consisting of selected elite

solutions reached during the search process. Such elite solutions typically are identified as elements of a regional cluster in intermediate term intensification strategies, and as elements of different clusters in longer term diversification strategies. In addition, elite solution components, in contrast to the solutions themselves, are included among the elements that can be retained and integrated to provide inputs to the search process.

Path Relinking (PR) (Glover et al., 2000) provides a fundamental means for pursuing the goals of intensification and diversification when its steps are implemented to exploit strategic choice rule variations. It is initiated by selecting two solutions s and s' from a collection of elite solutions produced during previous search phases. A path is then generated from s to s' , producing a solutions sequence $s = s(1), s(2), \dots, s(r) = s'$, where $s(i+1)$ is created from $s(i)$ at each step by choosing a move that leaves the fewest number of moves remaining to reach s' . Finally, once the path is completed, one or more of the solutions $s(i)$ is selected as a solution to start a new search phase, which can be based on tabu search as proposed in this paper. A number of alternative moves can be used to produce a next solution from $s(i)$ by the “fewest remaining moves” criterion, consequently obtaining a set of possible paths from s to s' . Selecting unattractive moves relative to the objective function value at each step will tend to produce a final series of strongly improving moves, while selecting attractive moves will tend to produce lower quality moves at the end.

3.1. Tabu search proposed by Cordeau et al. (2005)

Cordeau et al. (2005) propose a tabu search approach (T^2S) for solving the discrete dynamic berth allocation problem, in which a solution s is described by means of a set of attributes, *Attributes*, as indicated below:

$$Attributes(s) = \{(i, j) : \text{ship } j \text{ is assigned to berth } i\}.$$

The neighborhood $N(s)$ of a solution s consists of all the solutions \bar{s} which are obtained from s by substituting an attribute $(i, j) \in Attributes(s)$ by another one (i', j) , with $i \neq i'$. This move corresponds to the reallocation of vessel j from its current berth i to a new berth i' . When vessel j is removed from berth i , the sequence of vessels in that berth is reconnected by linking the predecessor and successor of j . Moreover, vessel j is allocated to berth i' in the position that minimizes the objective function value. This move is then forbidden for θ iterations, assigning a tabu-active status to attribute (i, j) .

T^2S employs an aspiration criterium by global objective. This criterium revokes the tabu-active status of an attribute if including it into the solution let us obtain a solution that improves the best solution found along the search process.

With the purpose of diversifying the search and drive it to less explored regions, Cordeau et al. propose the use of frequencies. Let $\bar{s} \in N(s)$ be the neighbor solution of s obtained by eliminating the attribute (i, j) from s . Let ρ_{ij} be the number of times that attribute (i, j) has been part of the solutions visited during the process, ζ the current iteration, and λ a parameter to control the intensity of the diversification. Let us consider the following modified objective function $c(H, \bar{s})$, which is defined as $f(\bar{s})$, if $f(\bar{s}) < f(s)$, and as stated below, otherwise:

$$c(H, \bar{s}) = f(\bar{s}) + \lambda f(\bar{s}) \rho_{ij} / \zeta$$

where $f(s)$ is the objective function value of solution s .

The set of neighbors of solution s , $N(H, s)$, consists of all the solutions in $N(s)$ that are non-tabu or that satisfy the aspiration criterium.

In order to generate the initial solution in T^2S , Cordeau et al. propose two procedures: $R-G$ (*Random Greedy*) and $FCFS-G$ (*First*

Come, First Served - Greedy). In $R-G$, given a random vessels permutation, they are assigned one at a time to the best possible berth following the given order. In $FCFS-G$, the vessels are ordered according to their arriving times and assigned as in $R-G$.

T^2S consists of the consecutive application of the tabu search starting from both solutions. In both cases, the search stops after 10,000 iterations. The remaining parameters used by T^2S were set to the following values: $\theta = \lfloor 7,5 \log n \rfloor$ and $\lambda = 0,015$. A more detailed explanation of this procedure can be found in Cordeau et al. (2005).

3.2. Tabu search with path relinking for the DBAP

In order to improve the effectiveness and efficiency of the tabu search described above, several alternatives can be considered. One of them consists of incorporating advanced memory structures that keep high quality solutions to be then used to intensify the search over promising regions of the search space. Another alternative is to use several neighborhood structures during the search process.

In this work, we propose to create an elite set of solutions, ES , which consists of a subset of the local optima encountered during the process. Moreover, in combination with the local search proposed by Cordeau et al. that uses the reallocation move, we implement a swap move, according to which vessels allocated to either the same or different berths can be exchanged. The new proposed tabu search, which makes use of two different neighborhood structures to guide the search, will be referred to as T^2S^* . The method proposed in this paper, that combines T^2S^* with path relinking, will be called $T^2S^* + PR$.

In a swap move, given a solution s represented by the set $attributes(s)$, the attributes (i_1, j_1) and (i_2, j_2) are changed by (i_2, j_1) and (i_1, j_2) . This move corresponds to the reallocation of vessels j_1, j_2 from their berths, i_1 and i_2 , to the berths i_2 and i_1 , respectively.

Let $N_1(s)$ and $N_2(s)$ be the neighborhoods of the solution s obtained by applying the reallocation and swap moves, respectively. Moreover, let $N_1(H, s)$ and $N_2(H, s)$ be the modified neighborhoods of s obtained from $N_1(s)$ and $N_2(s)$. A solution \bar{s} belongs to $N_1(H, s)$ if $\bar{s} \in N_1(s)$, none of its attributes is tabu-active or satisfy the aspiration criterium. The modified neighborhood $N_2(H, s)$ is defined in the same way.

In the algorithm proposed in this paper, T^2S^* , the move from a solution s to a solution s_{next} is carried out as follows. Let s' be the best non-tabu neighbor solution of s that can be reached by using the reallocation move, as indicated below:

$$s' = \arg \min_{\bar{s} \in N_1(H, s)} c(H, \bar{s}).$$

If $f(s') \leq f(s)$, the search for the best non-tabu neighbor solution s'' of s' is performed using the swap movement, as indicated below:

$$s'' = \arg \min_{\bar{s} \in N_2(H, s')} c(H, \bar{s}).$$

If $f(s'') \leq f(s')$, then $s_{next} = s''$; $s_{next} = s'$, otherwise.

The set of elite solutions is then used by the path relinking algorithm to create new starting points for the tabu search. The combination of tabu search with path relinking has been used in other papers from the literature Nguyen et al. (2012). Taking an elite solution s_{elite} as guiding solution, and as initial one, $s_{initial}$, a solution generated by $R-G$, a procedure based on path relinking, that iteratively brings $s_{initial}$ closer to s_{elite} , is applied. From the midpoint of this path, a new tabu search T^2S^* is then run. The new reached local optima are used to update the elite set. These steps are repeated until the stopping condition is met.

Let $s(i)$ be the current solution in the path and $B(s(i))$ the set of vessels with different allocation in $s(i)$ and s_{elite} , which is the guiding solution. In order to get the next solution in the path $s(i+1)$, a vessel $j \in B(s(i))$ is selected and is then allocated to the

```

procedure  $T^2S^* + PR$ ;
begin
   $s_{R-G} = R-G()$ ;
   $s_{FCFS-G} = FCFS-G()$ ;
   $ES = T^2S^*_{min}(s_{R-G})$ ;
   $ES = ES \cup T^2S^*_{min}(s_{FCFS-G})$ ;
  repeat
     $s_{initial} = R-G()$ ;
     $s_{elite} = Select(ES)$ ;
     $s = PR(s_{initial}, s_{elite})$ ;
     $ES = ES \cup T^2S^*_{min}(s)$ ;
  until (stopping condition);
end.

```

Fig. 2. Tabu search with path relinking.

same berth and position as it is in the guiding solution, s_{elite} . These steps are iteratively repeated until reaching s_{elite} . Fig. 3 shows an example of this process for the dynamic berth allocation problem with five vessels and two berths, in which the arriving times of the vessels, the availability of the berths and the processing times are reported in Table 1.

In the example shown in Fig. 3, in order to move from solution $s'(1)$ to solution $s'(2)$, the first difference between $s'(1)$ and the guiding solution is detected. In this case, vessel 5 is allocated to the third position of berth 1 in the guiding solution, while in $s'(1)$ it is allocated to berth 2. Therefore, in order to get closer to the guiding solution, vessel 5 is allocated accordingly, reaching the solution $s'(2)$ shown in Fig. 3. This process is iterated until reaching the guiding solution, s_{elite} .

Let s_{R-G} and s_{FCFS-G} be the solutions obtained from the methods $R-G$ and $FCFS-G$ described above, respectively. Let $T^2S^*_{min}(s_{R-G})$ and $T^2S^*_{min}(s_{FCFS-G})$ be the sets of local optima obtained after executing the tabu search from s_{R-G} and s_{FCFS-G} , respectively. Fig. 2 shows the pseudo-code of $T^2S^* + PR$. First of all, the elite set, ES , is initialized with the sets of local optima $T^2S^*_{min}(s_{R-G})$ and $T^2S^*_{min}(s_{FCFS-G})$. Then, a random solution, $s_{initial}$, is generated by running the Random Greedy approach, $R-G$, and solution $s_{elite} \in ES$ is randomly selected. A path relinking connecting $s_{initial}$ with s_{elite} is performed. Finally, the mid-point solution of this path, s , is selected as starting point to run the tabu search T^2S^* . The local optima reached along this search are also used to update the elite set of solutions. These steps are repeated until the stopping condition is met. In the initialization phase of the elite set, T^2S^* is run until reaching either a maximum number of iterations n_1 or a maximum number of iterations without improvements n_2 . Similarly, in the improvement phase, T^2S^* is executed until either a maximum number of iterations n_3 is met or a maximum number of iterations without improvements is met. $T^2S^* + PR$ stops when a maximum number of path relinkings n_5 is performed.

4. Computational experiments

The computational experience has been carried out following two phases. In the first one, the values for the parameters used by the proposed algorithm $T^2S^* + PR$ are set. In the second one, the method $T^2S^* + PR$ is compared with T^2S , T^2S^* and the exact resolution of the mathematical model $GSPP$ described in Section 2. In order to carry out this experience, we have considered problems of different sizes that range from 25 up to 60 vessels and from 5 up to 13 berths. We have used the 50 problems of the set $I2$ and the 30 problems of the set $I3$ of Cordeau et al. (2005). Moreover, in order to reach performance conclusions regarding the instances, we have generated a new set of realistic test instances¹. Table 2 describes the characteristics of these new

instances, which were generated according to Cordeau et al. (2005) with longer time horizon, higher traffic, and less number of available berths. Under the headings *Vessels* and *Berths*, the number of vessels and available berths are respectively represented. Last column shows the time horizon corresponding to the berths availability. All programs have been coded in Ansi C and the experiments have been performed in a PC with a T4300 processor at 2.10 GHz. The computational times reported in this section are given in seconds.

4.1. Parameters setting

The behavior of $T^2S^* + PR$ depends on several parameters: the elite set size ($|ES|$), maximum number of iterations executed by T^2S^* to construct the elite set at the initialization phase (n_1), number of iterations without improvements of T^2S^* in the initialization phase (n_2), number of iterations run by T^2S^* when it is used as an improvement method after relinking two solutions (n_3), number of iterations without improvements of T^2S^* in the relinking phase (n_4) and maximum number of relinkings (n_5).

The size of the elite set is set to 10 solutions. In order to measure the influence of the remaining parameters over the effectiveness and efficiency of $T^2S^* + PR$, different combinations of values are fixed. Table 3 summarizes these combinations.

With the purpose of carrying out the parameters setting phase, one third of the problems in classes $I2$ and $I3$ are randomly chosen (1/6 of each class; 7 problems of class $I2$ and 5 problems of class $I3$). Each problem is solved 10 times. The mean objective value and mean CPU time are considered as response variables. The analysis has been performed following two steps, in which the effect of the parameters setting over both the mean objective value and the mean CPU time are taken into account, respectively. Table 4 summarizes the obtained results. For each parameters combination C_i , we report the mean objective value and mean CPU time.

In order to determine the differences among the combinations of parameters, we use the Friedman nonparametric statistical test (see, for instance, Daniel, 1990). In the cases in which the null hypothesis of equality of treatments is rejected, we use the multiple comparisons test of Friedman to determine the differences among the combinations.

Table 5 summarizes the results obtained when applying the Friedman test for the comparison of equality in the objective function values and CPU times. The results corroborate that there are not significant differences in the mean objective function values when the parameters change. However, significant differences are detected in the mean CPU times. Table 6 shows the results obtained when applying the multiple comparisons test of Friedman at $\alpha = 0.05$ significance level for the CPU times. The first row reports the ranks of the different parameters combinations. Then, for each combination C_i , the combinations that have the same behavior of the given row are shown; i.e. those combinations C_j whose ranks are not higher than the critical difference CD ($|R_i - R_j| \geq CD$).

From the obtained results, we conclude that there are not statistically significant differences between the combinations of parameters C_1 , C_3 and C_5 , regarding the computational time. For the computational experience that follows in this section, we have chosen the combination of parameters C_1 . The rationale behind this election is the fact that its rank is the lowest, although there are not significant differences between C_1 , C_3 and C_5 .

In contrast with other path relinking procedures, the algorithm proposed in this paper does not consider as initial and guiding solutions elements from the elite set. The rationale behind this fact is that we have experimentally seen that this strategy does not provide good solutions for the considered problem. This may

¹ These instances are available in <https://sites.google.com/site/gciports/berth-allocation-problem>



Fig. 3. Example of execution of path relinking.

Table 2
Characteristics of the generated instances.

Set	Vessels	Berths	H
A1	30	3	600
A2	30	5	600
B1	40	5	600
B2	40	7	600
C1	55	5	600
C2	55	7	600
C3	55	10	600
D1	60	5	600
D2	60	7	600
D3	60	10	600

Table 3
Values of the parameters n_1, n_2, n_3, n_4, n_5 .

Combination	n_1	n_2	n_3	n_4	n_5
C ₁	50	10	20	5	5
C ₂	50	10	20	5	10
C ₃	100	20	10	5	5
C ₄	100	20	10	5	10
C ₅	100	20	20	5	5
C ₆	100	20	20	5	10

be due to the fact that the solutions in the elite set belong to the same subregion of the search space.

Table 7 shows the results obtained when solving with $T^2S^* + PR$ three instances randomly selected from the classes I2 and I3 of Cordeau et al. (2005) using both path relinking strategies (Strategy 1: selects the initial and guiding solutions from the elite set at random; Strategy 2: uses a solution generated with the R-G method as initial solution and one solution from the elite set as guiding one). For each path relinking strategy, several combinations of parameters are used. The first eight columns of Table 7 indicate the number of vessels ($|V|$), number of berths ($|B|$), problem (*inst.*), and the values to which the parameters n_1, n_2, n_3, n_4 and n_5 are set. The four next columns summarize the mean objective function values and the CPU times required for each strategy. Last column reports the percentage differences between strategies 1 and 2. These results corroborate that in all cases the second strategy provides better objective values than the first one.

4.2. Comparative analysis

The goal of the computational experiments reported in this section is to determine the performance of the proposed heuristic procedure $T^2S^* + PR$ when compared to the tabu search procedures T^2S and T^2S^* , and to the resolution of the exact model $GSPP$. For this purpose, the problem instances of the classes I2

Table 4
Parameters setting results.

$ V $	$ B $	inst.	C_1	C_2	C_3	C_4	C_5	C_6	
25	5	4	693.60 0.192	691.50 0.259	692.90 0.234	690.70 0.320	691.60 0.236	694.00 0.324	
		6	1129.00 0.192	1129.00 0.261	1129.00 0.242	1129.00 0.308	1129.00 0.235	1129.00 0.298	
		9	752.60 0.214	752.30 0.286	752.50 0.258	752.60 0.337	752.60 0.259	752.60 0.336	
		7	10	825.00 0.249	825.00 0.322	825.00 0.334	825.00 0.397	825.00 0.333	825.00 0.389
		10	3	764.60 0.306	764.10 0.391	764.70 0.320	763.50 0.486	763.90 0.384	764.70 0.484
	8	855.00 0.283	855.00 0.402	855.00 0.392	855.00 0.502	855.00 0.452	855.00 0.528		
	35	10	1	1125.10 0.908	1124.90 1.156	1125.00 1.509	1125.00 1.625	1125.10 1.436	1124.90 1.705
	60	13	10	1213.60 5.786	1213.20 8.770	1213.30 6.847	1213.20 9.861	1213.50 7.159	1213.00 9.913
			16	1367.50 5.962	1370.30 8.931	1367.10 7.566	1365.90 11.178	1368.60 7.713	1366.40 10.864
			18	1345.00 5.363	1345.00 8.108	1345.00 8.241	1345.00 10.636	1345.00 8.641	1345.00 10.606
25			1383.40 6.952	1380.20 9.463	1379.40 9.238	1379.40 12.864	1379.90 8.495	1379.80 12.220	
29			1281.80 5.908	1281.50 8.569	1281.70 7.203	1281.40 10.660	1281.20 7.674	1281.60 10.845	

Table 5
 p -values associated to the statistic test of Friedman.

Variable	Statistic test value	Theoretical value	p -value
Objective value	4.7083	9.24	≥ 0.1
CPU time	52.6190	9.24	≤ 0.005

Table 6
Multiple comparisons of Friedman for the CPU time. $\alpha = 0.05$, $CD = 27.1284$.

Rank	12	31	36	41	66	66
C_1	C_1	C_3	C_5	C_2	C_4	C_6
C_3	C_1	C_3	C_5	C_2		
C_5		C_3	C_5	C_2		
C_2			C_5	C_2	C_4	C_6

and 13 of Cordeau et al. (2005) as well as the new instances generated in this paper are solved with all the procedures. The heuristic procedures are executed 30 times for each problem instance.

Tables 8–10 illustrate the results obtained for small, medium and large instances, respectively. In these tables, under the headings CPLEX-GSPP, T^2S , T^2S^* and $T^2S^* + PR$, the results provided by the mathematical model, the tabu search from the literature, the tabu search proposed in this paper as well as the combination of our tabu search with path relinking are reported, respectively. Moreover, *Opt Val* indicates the optimal value for a given instance, while *Obj Val* gives the objective function value obtained by the corresponding method.

In Table 8, the gaps corresponding to T^2S and T^2S^* are calculated with respect to the optimum values provided by the mathematical model GSPP. On the other hand, the gaps corresponding to $T^2S^* + PR$, *Gap1* and *Gap2* are calculated with respect to the optimum values and to the T^2S^* objective values, respectively. In Tables 9 and 10, the gaps corresponding to T^2S^* are calculated with respect to T^2S objective values obtained with the procedure by Cordeau et al. (2005), while the gaps corresponding to $T^2S^* + PR$, *Gap1* and *Gap2* are calculated with respect to the T^2S

and T^2S^* objective values, respectively. In each case, CPLEX was run until optimality was proven or it run out of memory.

For small size instances, CPLEX was able to solve the instances to optimality in all cases. However, for medium and large size instances CPLEX terminated with a memory fault and no solution was found. The disability of the mathematical model GSPP to solve more realistic instances had already been noticed in Buhrkal et al. (2011).

Table 8 shows that T^2S obtains 4 out of 30 optimal solutions, whereas T^2S^* and $T^2S^* + PR$ reach 20 and 27, respectively. In addition, $T^2S^* + PR$ improves the results given by T^2S^* in the remaining three instances. Furthermore, T^2S^* and $T^2S^* + PR$ are significantly more efficient than both GSPP and T^2S . Note that the time reductions of T^2S^* with respect to GSPP and T^2S are 98.67% and 87.60%, respectively. Finally, the time reductions of $T^2S^* + PR$ with respect to GSPP and T^2S are 98.30% and 84.15%, respectively. Note that $T^2S^* + PR$ improves in 10 out of 30 instances the results reached by T^2S^* with a light increment in the computational time.

Table 9 shows that for medium size instances, $T^2S^* + PR$ obtains the best objective function value in all cases, while T^2S^* and T^2S obtain 15 and 0 out of 40, respectively. The time reductions of T^2S^* and $T^2S^* + PR$ with respect to T^2S are 82.77% and 78.37%, respectively. For large size instances, whose results are summarized in Table 10, $T^2S^* + PR$ also obtains the best objective function value in all cases, while T^2S^* and T^2S obtain 10 and 0 out of 30, respectively. The time reductions of T^2S^* and $T^2S^* + PR$ with respect to T^2S are 79.48% and 77.37%, respectively.

The Wilcoxon signed-rank non-parametric statistical test is then used to statistically corroborate the above-mentioned conclusions. The performance of the different heuristics implemented in this paper for solving the DBAP has been compared. The results are reported in Table 11. The first column corresponds to the experiment, the second column to the p -value and the third column corresponds to the error-value α . The comparison between T^2S^* and $T^2S^* + PR$ with T^2S resulted in a extremely small p -value, so that we can reject the null hypothesis of performance equality and conclude that both T^2S^* and $T^2S^* + PR$ perform significantly better than T^2S , that is the tabu search from the literature. Furthermore, the test also revealed that $T^2S^* + PR$ presents significant differences with respect to T^2S^* at a

Table 7

Strategy 1 (two solutions randomly chosen from the elite set) vs. strategy 2 (the guiding solution randomly chosen from the elite set and the initial solution generated with R-G).

V	B	inst.	n ₁	n ₂	n ₃	n ₄	n ₅	Strategy 1		Strategy 2		Strategy 1 vs. Strategy 2 Gap (obj.1,obj.2)
								Obj.	CPU	Obj.	CPU	
25	5	4	50	10	20	5	5	709.80	0.075	693.60	0.192	-2.28
								711.00	0.078	691.50	0.259	-2.74
								717.50	0.109	692.90	0.234	-3.42
			100	20	10	5	5	710.70	0.113	690.70	0.320	-2.81
								714.60	0.119	691.60	0.235	-3.21
								717.90	0.120	694.00	0.323	-3.32
35	10	1	50	10	20	5	5	1127.30	0.451	1125.10	0.907	-0.19
								1127.00	0.464	1124.90	1.156	-0.18
								1130.80	0.591	1125.00	1.509	-0.44
			100	20	10	5	5	1137.30	0.611	1125.00	1.624	-1.05
								1134.80	0.589	1125.10	1.435	-0.85
								1135.70	0.648	1124.90	1.704	-0.95
60	13	16	50	10	20	5	5	1386.00	1.908	1367.50	5.962	-1.33
								1386.00	1.903	1370.30	8.931	-1.13
								1386.00	3.113	1367.10	7.565	-1.36
			100	20	10	5	5	1386.00	3.113	1365.90	11.178	-1.45
								1386.00	3.170	1368.60	7.712	-1.25
								1386.00	3.183	1366.40	10.864	-1.41

Table 8

Small size experimental results.

Size	Inst.	CPLEX-GSPP		T ² S			T ² S*			T ² S*+PR			
		Opt Val	Time	Obj Val	Gap	Time	Obj Val	Gap	Time	Obj Val	Gap ₁	Gap ₂	Time
30 × 03	1	1763	11,25	1763	0,00	1,90	1763	0,00	0,25	1763	0,00	0,00	0,25
	2	2090	22.59	2090	0,00	1,91	2090	0,00	0,14	2090	0,00	0,00	0,26
	3	2186	13.68	2188	0,09	2,15	2186	0,00	0,23	2186	0,00	0,00	0,26
	4	1538	12.98	1544	0,39	1,83	1538	0,00	0,23	1538	0,00	0,00	0,28
	5	2114	16.12	2114	0,00	1,82	2114	0,00	0,18	2114	0,00	0,00	0,22
	6	2185	36.97	2187	0,09	1,83	2187	0,09	0,25	2185	0,00	-0,09	0,34
	7	1845	23.29	1849	0,22	1,85	1847	0,11	0,25	1845	0,00	-0,11	0,34
	8	1271	9.77	1278	0,55	1,84	1271	0,00	0,17	1271	0,00	0,00	0,31
	9	1595	28.25	1595	0,00	1,84	1595	0,00	0,26	1595	0,00	0,00	0,25
	10	2195	9.64	2197	0,09	1,82	2195	0,00	0,25	2195	0,00	0,00	0,29
30 × 05	1	1149	17.93	1153	0,35	2,68	1149	0,00	0,28	1149	0,00	0,00	0,47
	2	1475	30.59	1480	0,34	2,61	1476	0,07	0,26	1475	0,00	-0,07	0,47
	3	1542	26.43	1547	0,32	2,62	1542	0,00	0,28	1542	0,00	0,00	0,50
	4	1075	15.72	1077	0,19	2,74	1075	0,00	0,30	1075	0,00	0,00	0,48
	5	1463	23.60	1475	0,82	2,65	1463	0,00	0,31	1463	0,00	0,00	0,40
	6	1580	30.84	1587	0,44	2,54	1581	0,06	0,39	1580	0,00	-0,06	0,48
	7	1276	19.39	1279	0,24	2,66	1276	0,00	0,23	1276	0,00	0,00	0,46
	8	870	22.06	877	0,80	2,64	870	0,00	0,31	870	0,00	0,00	0,42
	9	1134	22.31	1156	1,94	2,64	1153	1,68	0,34	1134	0,00	-1,65	0,50
	10	1527	18.80	1536	0,59	2,69	1527	0,00	0,29	1527	0,00	0,00	0,46
40 × 05	1	2301	41.51	2317	0,70	5,82	2307	0,26	0,84	2303	0,09	-0,17	0,90
	2	2829	59.89	2839	0,35	5,76	2835	0,21	0,94	2834	0,18	-0,04	1,09
	3	2880	99.20	2886	0,21	5,79	2880	0,00	0,58	2880	0,00	0,00	0,50
	4	2001	39.78	2033	1,60	5,80	2001	0,00	0,64	2001	0,00	0,00	0,84
	5	2815	74.14	2834	0,67	5,90	2815	0,00	0,72	2815	0,00	0,00	0,76
	6	2934	66.46	2945	0,37	5,91	2934	0,00	0,84	2934	0,00	0,00	0,87
	7	2632	40.97	2656	0,91	5,86	2632	0,00	0,79	2632	0,00	0,00	0,79
	8	1835	40.11	1852	0,93	6,01	1836	0,05	0,96	1835	0,00	-0,05	1,28
	9	2086	47.70	2119	1,58	5,99	2095	0,43	0,89	2089	0,14	-0,29	1,07
	10	2962	52.28	2976	0,47	5,97	2964	0,07	0,63	2962	0,00	-0,07	1,06
		1904.93	32.48	1914.30		3,47	1906.57		0,43	1905.27			0,55

Bold numbers indicate that the optimal solution is reached.

confidence level of $\alpha = 0.05$. Therefore, we conclude that $T^2S^* + PR$ outperforms both T^2S and T^2S^* .

To sum up this section, we indicate that the computational experience carried out in this paper corroborates the effectiveness and efficiency of the proposed procedure, $T^2S^* + PR$, since it

reaches the optimal solution for most of the instances that Cplex is able to solve being more efficient than the resolution of the mathematical model GSPP. Moreover, it outperforms the tabu search proposed in the literature. This fact let us state that the algorithm can be used as an efficient method for solving the berth

Table 9
Medium size experimental results.

Size	Instance	T^2S		T^2S^*			$T^2S^* + PR$			
		Obj Val	Time	Obj Val	Gap	Time	Obj Val	Gap ₁	Gap ₂	Time
40	1	1489	8.16	1467	-1.48	1.20	1460	-1.95	-0.48	1.11
	2	1423	8.26	1381	-2.95	1.01	1375	-3.37	-0.43	1.32
	3	2149	8.13	2119	-1.40	0.84	2119	-1.40	0.00	1.17
	4	1618	7.99	1600	-1.11	1.18	1597	-1.30	-0.19	1.78
	5	1885	8.37	1849	-1.91	1.11	1847	-2.02	-0.11	1.45
	6	2104	8.11	2080	-1.14	0.86	2080	-1.14	0.00	1.37
	7	1863	7.91	1845	-0.97	1.25	1841	-1.18	-0.22	1.56
	8	2040	8.16	2026	-0.69	1.18	2026	-0.69	0.00	1.70
	9	1901	8.01	1888	-0.68	1.06	1880	-1.10	-0.42	1.48
	10	1922	7.89	1905	-0.88	0.71	1892	-1.56	-0.68	1.59
55 × 05	1	4701	14.33	4693	-0.17	1.65	4689	-0.26	-0.09	2.82
	2	5496	14.25	5483	-0.24	1.37	5467	-0.53	-0.29	2.81
	3	5523	14.20	5499	-0.43	1.92	5499	-0.43	0.00	2.67
	4	4249	14.27	4189	-1.41	1.76	4179	-1.65	-0.24	3.65
	5	5590	14.39	5484	-1.90	1.39	5478	-2.00	-0.11	2.73
	6	5609	14.25	5599	-0.18	1.45	5595	-0.25	-0.07	2.56
	7	4914	14.44	4902	-0.24	1.90	4882	-0.65	-0.41	3.82
	8	3585	14.00	3565	-0.56	1.54	3552	-0.92	-0.36	2.79
	9	4301	14.06	4277	-0.56	1.67	4275	-0.60	-0.05	2.75
	10	5831	14.14	5739	-1.58	1.85	5739	-1.58	0.00	2.65
55 × 07	1	2871	19.22	2846	-0.87	3.60	2846	-0.87	0.00	4.78
	2	2941	19.13	2887	-1.84	3.03	2883	-1.97	-0.14	4.94
	3	3853	19.44	3840	-0.34	4.31	3833	-0.52	-0.18	5.57
	4	3022	19.88	2977	-1.49	2.51	2971	-1.69	-0.20	4.31
	5	3845	19.01	3803	-1.09	3.75	3801	-1.14	-0.05	3.56
	6	3833	19.36	3783	-1.30	2.59	3783	-1.30	0.00	3.70
	7	3844	19.36	3774	-1.82	2.43	3774	-1.82	0.00	3.84
	8	3893	19.34	3864	-0.74	2.09	3863	-0.77	-0.03	3.95
	9	3627	18.91	3597	-0.83	3.03	3591	-0.99	-0.17	5.26
	10	3699	19.44	3658	-1.11	2.61	3635	-1.73	-0.63	4.73
55 × 10	1	2777	26.66	2745	-1.15	8.09	2745	-1.15	0.00	7.31
	2	2577	29.89	2549	-1.09	8.23	2534	-1.67	-0.59	6.10
	3	2570	29.66	2545	-0.97	6.20	2545	-0.97	0.00	6.53
	4	3351	28.25	3315	-1.07	7.00	3315	-1.07	0.00	5.59
	5	3157	30.19	3147	-0.32	7.66	3123	-1.08	-0.76	6.12
	6	2293	29.42	2283	-0.44	6.48	2283	-0.44	0.00	6.54
	7	2172	28.78	2146	-1.20	5.04	2146	-1.20	0.00	9.17
	8	2783	30.56	2743	-1.44	4.98	2726	-2.05	-0.62	5.18
	9	2212	29.69	2162	-2.26	6.50	2162	-2.26	0.00	6.50
	10	2894	30.97	2815	-2.73	5.45	2815	-2.73	0.00	6.05
		3260.18	17.76	3226.73		3.06	3221.15			3.84

Bold numbers indicate the best known solutions.

Table 10
Large size experimental results.

Size	Instance	T^2S		T^2S^*			$T^2S^* + PR$			
		Obj Val	Time	Obj Val	Gap	Time	Obj Val	Gap ₁	Gap ₂	Time
60 × 05	1	5763	18.35	5761	-0.03	1.99	5753	-0.17	-0.14	3.12
	2	6932	18.03	6884	-0.69	2.67	6884	-0.69	0.00	3.20
	3	6795	18.09	6782	-0.19	2.17	6780	-0.22	-0.03	4.25
	4	5172	18.52	5105	-1.30	2.30	5105	-1.30	0.00	2.30
	5	6747	17.93	6715	-0.47	2.47	6715	-0.47	0.00	3.18
	6	6637	18.01	6618	-0.29	2.45	6616	-0.32	-0.03	3.53
	7	6073	18.43	6011	-1.02	2.66	6011	-1.02	0.00	4.75
	8	4415	18.08	4406	-0.20	2.64	4385	-0.68	-0.48	3.77
	9	5263	17.95	5235	-0.53	2.17	5235	-0.53	0.00	3.99
	10	7350	18.14	7281	-0.94	2.22	7281	-0.94	0.00	3.62
60 × 07	1	3735	24.30	3724	-0.29	4.40	3715	-0.54	-0.24	9.26
	2	4279	24.99	4191	-2.06	6.39	4172	-2.50	-0.45	6.70
	3	4291	24.36	4290	-0.02	6.78	4281	-0.23	-0.21	5.90
	4	3926	24.44	3916	-0.25	5.32	3916	-0.25	0.00	7.15
	5	4294	24.64	4264	-0.70	3.99	4261	-0.77	-0.07	6.23
	6	5741	24.17	5731	-0.17	6.56	5729	-0.21	-0.03	4.39
	7	3825	24.32	3749	-1.99	6.61	3743	-2.14	-0.16	8.28
	8	4735	25.47	4600	-2.85	6.96	4586	-3.15	-0.30	6.96
	9	4049	25.73	4011	-0.94	5.39	4004	-1.11	-0.17	5.39
	10	4222	24.91	4125	-2.30	5.66	4115	-2.53	-0.24	7.37
60 × 10	1	2204	23.06	2190	-0.64	5.37	2190	-0.64	0.00	5.37
	2	2395	24.09	2359	-1.50	4.69	2358	-1.54	-0.04	3.84
	3	1592	23.84	1577	-0.94	4.72	1575	-1.07	-0.13	5.42
	4	2632	22.80	2597	-1.33	6.13	2578	-2.05	-0.73	3.78
	5	2975	23.72	2961	-0.47	6.10	2954	-0.71	-0.24	4.53
	6	2258	23.88	2180	-3.45	6.31	2180	-3.45	0.00	5.20
	7	2235	23.73	2177	-2.60	4.62	2177	-2.60	0.00	4.62
	8	2216	24.14	2151	-2.93	6.06	2148	-3.07	-0.14	4.34
	9	2681	23.74	2661	-0.75	4.92	2660	-0.78	-0.04	5.21
	10	2089	23.68	2048	-1.96	5.88	2034	-2.63	-0.68	4.90
		4250.70	22.18	4210		4.55	4204.70			5.02

Bold numbers indicate the best known solutions.

Table 11
Wilcoxon signed ranks test.

Experiment	<i>p</i> -value	α (%)
T2S–T2S*	< 0.001	0.05
T2S–T2S*+PR	< 0.001	0.05
T2S* - T2S*+PR	< 0.001	0.05

allocation problem in integrated designs as the one proposed by Giallombardo et al. (2010).

5. Conclusions

In this paper we propose a hybrid metaheuristic that combines Tabu Search with Path Relinking to solve the Dynamic Berth Allocation Problem. An elite set of solutions, which consists of local optima found along the tabu search process, is used to build, by path relinking, new initial solutions for the Tabu Search. Moreover, the proposed tabu search makes use of two different neighborhood structures to guide the search. This fact implies that defining the pattern that has to be followed to perform the different moves is then required. The computational experience carried out in this paper corroborates the effectiveness and efficiency of the proposed strategy since they are both increased when compared with other methods from the literature. This fact justifies the use of T^2S^*+PR as a procedure to solve the dynamic berth allocation problem; particularly in those integrated designs in which this problem appears as a subproblem. In these cases it is preferable to use an efficient procedure that provides near-optimal solutions.

As future work we propose the use of other adaptive memory structures that let improve the performance of the proposed method, which already outperforms other heuristic algorithm proposed in the literature. Moreover, we intend to explore other alternatives for the construction of the elite set of solutions (eg., using a mechanism for diversification as it is than in the Scatter Search metaheuristic). Finally, we plan to apply the proposed algorithm over integrated models in which the berth allocation problem is combined with the quay crane assignment problem.

Acknowledgement

This work has been partially funded by the European Regional Development Fund, the Spanish Ministry of Science and Technology (projects TIN2009-13363 and TIN2008-06872-C04-01), and

the Canary Government (project PI2007/019). Eduardo Lalla thanks the Canary Government the financial support he receives through his doctoral grant.

References

- Bierwirth, C., Meisel, F., 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* 202 (3), 615–627.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., Lusby, R., 2011. Models for the discrete berth allocation problem: a computational comparison. *Transp. Res. E* 47 (4), 461–473.
- Cordeau, J.-F., Laporte, G., Legato, P., Moccia, L., 2005. Models and tabu search heuristics for the berth-allocation problem. *Transpor. Sci.* 39 (4), 526–538.
- Daniel, W.W., 1990. *Applied Nonparametric Statistics*. PWS-Kent Publishing Company, Boston.
- ENO Transportation Foundation. <<http://www.enotrans.com/>>.
- Giallombardo, G., Moccia, L., Salani, M., Vacca, I., 2010. Modeling and solving the tactical berth allocation problem. *Transp. Res. B Methodol.* 44 (2), 232–245.
- Glover, F., Laguna, M., 1993. *Tabu Search*. Kluwer, Boston.
- Glover, F., Laguna, M., Martí, R., 2000. Fundamentals of scatter search and path relinking. *Control Cybern.* 39 (3), 653–684.
- Hansen, P., Oguz, C., Mladenović, N., 2008. Variable neighborhood search for minimum cost berth allocation. *Eur. J. Oper. Res.* 191 (3), 636–649.
- Imai, A., Nishimura, E., Hattoric, M., Papadimitriou, S., 2007. Berth allocation at indented berths for mega-containerships. *Eur. J. Oper. Res.* 179 (2), 579–593.
- Imai, A., Chen, H.C., Nishimura, E., Papadimitriou, S., 2008. The simultaneous berth and quay crane allocation problem. *Transp. Res. E Logistics Transp. Rev.* 44 (5), 900–920.
- Imai, A., Nagaiwa, K., Chan, W.T., 1997. Efficient planning of berth allocation for container terminals in asia. *J. Adv. Transp.* 31 (1), 75–94.
- Imai, A., Nishimura, E., Papadimitriou, S., 2001. The dynamic berth allocation problem for a container port. *Transp. Res. B Methodol.* 35 (4), 401–407.
- Imai, A., Nishimura, E., Papadimitriou, S., 2003. Berth allocation with service priority. *Transp. Res. B Methodol.* 37 (5), 437–457.
- Imai, A., Nishimura, E., Papadimitriou, S., 2008. Berthing ships at a multi-user container terminal with a limited quay capacity. *Transp. Res. E Logistics Transp. Rev.* 44 (1), 136–151.
- Lee, Y., Chen, C.-Y., 2009. An optimization heuristic for the berth scheduling problem. *Eur. J. Oper. Res.* 196 (2), 500–508.
- Liang, C., Huang, Y., Yang, Y., 2009. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Comput. Ind. Eng.* 56 (3), 1021–1028.
- Lim, A., 1998. The berth planning problem. *Oper. Res. Lett.* 22 (2–3), 100–105.
- Monaco, M.F., Sammarra, M., 2007. The berth allocation problem: a strong formulation solved by a Lagrangean approach. *Transpor. Sci.* 41 (2), 265–280.
- Nguyen, V.-P., Prins, C., Prodhon, C., 2012. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Eng. Appl. Artif. Intell.* 25 (1), 56–71.
- Silva, V.M.D., Coelho, A.S., Mayerle, S.F., 2008. Berthing ships at a multi-user container terminal with a limited quay capacity. *Investigação Oper.* 26, 31–43.
- Stahlbock, R., Voß, S., 2008. Operations research at container terminals: a literature update. *OR Spectrum* 30 (1), 1–52.
- Steenken, D., Voß, S., Stahlbock, R., 2004. Container terminal operation and operations research—a classification and literature review. *OR Spectrum* 26 (1), 3–49.
- Vis, I.F.A., de Koster, R., 2003. Transshipment of containers at a container terminal: an overview. *Eur. J. Oper. Res.* 147 (1), 1–16.