

Decentralized Cooperative Metaheuristic for the Dynamic Berth Allocation Problem

Eduardo Lalla-Ruiz, Christopher Expósito-Izquierdo, Belén Melián-Batista, J. Marcos Moreno-Vega

Department of Computer Engineering and Systems
{elalla, cexposit, mbmelian, jmmoreno}@ull.es

Abstract The increasing demand of maritime transport and the great competition among port terminals force their managers to reduce costs by exploiting its resources accurately. In this environment, the Berth Allocation Problem, which aims to allocate and schedule incoming vessels along the quay, plays a relevant role in improving the overall terminal productivity. In order to address this problem, we propose Decentralized Cooperative Metaheuristic (DCM), which is a population-based approach that exploits the concepts of communication and grouping. In DCM, the individuals are organized into groups, where each individual shares information with its group partners. This grouping strategy allows to diversify as well as intensify the search in some regions by means of information shared among the individuals of each group. Moreover, the constrained relation for sharing information among individuals through the proposed grouping strategy allows to reduce computational resources in comparison to the 'all to all' communication strategy. The computational experiments for this problem reveal that DCM reports high-quality solutions and identifies promising regions within the search space in short computational times.

Keywords: Berth Allocation Problem, Maritime Container Terminal, Decentralized Cooperative Metaheuristic

1 Introduction

Maritime container terminals are infrastructures built with the goal of facing the technical requirements arising from the increasing volume of containers in the international sea freight trade. They are aimed at transferring and storing containers within multi-modal transportation networks. The main transport modes found at a maritime container terminal are container vessels, trucks, and trains. In this regard, according to the UNCTAD¹, the international maritime container trade has greatly grown over the last decades. One of the most widespread indicators for assessing the competitiveness of a maritime container terminal is the time required to serve the container vessels arriving to the port [15]. For this reason, an inefficient utilization of some key resources, like berths, could produce delays of yard-side and land-side operations, giving rise to a poor overall productivity of the container terminal.

The aforementioned issue leads to the definition of the Berth Allocation Problem (BAP). Its main goal is to assign berthing positions along the quay to incoming vessels. In this process, container terminal managers must consider several factors such as the vessels and berth time windows, number of loaded/unloaded containers, water depth, and tide conditions. In this paper, we study the Dynamic Berth Allocation Problem (DBAP) introduced in [5], which considers berth and vessel time windows as well as heterogeneous vessel service times stemming from the assigned berth.

¹United Nations Conference on Trade And Development, <http://unctad.org>

In order to solve the DBAP, this work proposes Decentralized Cooperative Metaheuristic (DCM). This algorithm is a population-based approach in which a set of individuals is organized into groups that exchange information among them while the search is performed. In this regard, as indicated in [7], the ‘all to all’ communication in working systems is not appropriate because it demands too many computational resources. Therefore, the way the information is shared in DCM pursues a decentralized grouping strategy. Namely, during the search, the individuals only share information with their group partners. Each group has its own leader and rules regarding how to exchange information.

The goals of this work are, on the one hand, to assess the behaviour of DCM as well as provide high-quality solutions by means of short computational times for the berth allocation at maritime container terminals. On the other hand, we seek to evaluate the effectiveness of DCM by comparing its computational results with those reported by the mathematical model proposed in [3] and the results obtained by the best algorithms from the related literature for the DBAP. In this regard, as discussed in the relevant section, the computational results provided by DCM indicate it requires lesser computational time than the best solution approach recently proposed in the literature for the DBAP.

The remainder of this paper is organized as follows. A short literature review of the BAP is presented in Section 2. Then, the DBAP addressed in this work is described in Section 3. In Section 4, the algorithm proposed for addressing the DBAP is described. Later, the computational experience carried out and a comparative summary are presented in Section 5. Finally, some conclusions and several lines for further research are drawn in Section 6.

2 Literature Review

The Berth Allocation Problem (BAP) has been extensively studied in the literature. In this regard, due to the large variety of maritime terminal layouts, research has produced multitude of variants of this problem. Depending on how the quay is modelled, the BAP can be referred to as discrete (the quay is divided into segments called berths) or continuous (the quay is not divided, thus the vessels can berth at any position in the quay). Moreover, in some related works (*e.g.*, [5], [14]) there is also a hybrid consideration of the quay (the quay is divided into a set of berths and a vessel can occupy more than one berth at a time or share its assigned berth with other container vessels). Depending on the arrival time, the BAP can be classified into static (the vessels are already in port when the berths become available) or dynamic (the vessels arrive during the planning horizon). For detailed descriptions, the reader is referred to the works [1] and [4].

One of the most relevant variants is the Dynamic Berth Allocation Problem (DBAP). It was first formulated in [9] as an extension of the model proposed in [8] for the Static Berth Allocation Problem. Alternative formulations for the dynamic problem have been proposed and studied in [12], [5], and [3]. These models are described and compared in [2]. The main conclusion extracted from the latter work is that the model presented in [3] is superior to the other models when considering the temporal behaviour. In this regard, it is able to reach the optimal solutions within short computational time for the set of instances used by all the previous authors.

Recently, [11] presented an efficient Tabu Search metaheuristic with Path-Relinking for solving the DBAP. The authors also proposed a benchmark suite of instances for which the model from [3] does not provide feasible solutions within a time limit. [6] presents a Clustering Search (CS-SA) with Simulated Annealing for solving the DBAP. This algorithm provides the optimal solutions for all the largest instances proposed in [5]. In this regard, [13] propose a Particle Swarm Optimization algorithm for addressing the DBAP, which reports optimal solutions within shorter computational times than CS-SA.

3 Dynamic Berth Allocation Problem

In this work, we address the Dynamic Berth Allocation Problem (DBAP) proposed in [5], which is modeled as a Multi-Depot Vehicle Routing Problem with Time-Windows (MDVRPTW). The vessels are seen as customers and the berths as depots at which one vehicle is located. The goal of the DBAP is to determine the berthing position and berthing time of $|N|$ incoming vessels along the quay, which is

divided into $|M|$ berths. In order to make this paper self-contained, the description of the model proposed in [5] is included. The following parameters are defined in the problem:

- N , set of vessels
- M , set of berths
- t_i^k , handling time of vessel $i \in N$ at berth $k \in M$
- a_i, b_i , arrival and departure time of vessel $i \in N$
- l^k, e^k , start and end of the availability of the berth $k \in M$
- v_i , the service priority of each vessel $i \in N$

Let us define a graph, $G^k = (V^k, A^k) \forall k \in M$, where $V^k = N \cup \{o(k), d(k)\}$ contains a vertex for each vessel as well as the vertices $o(k)$ and $d(k)$, which are the origin and destination nodes for any route in the graph. The set of arcs is defined as $A^k \subseteq V^k \times V^k$, where each one represent the handling time of the vessel. The decision variables are as follows:

- $x_{ij}^k \in \{0, 1\}, \forall k \in M, \forall (i, j) \in A^k$, set to 1 if vessel j is scheduled after vessel i at berth k , and 0 otherwise
- $T_i^k, \forall k \in M, \forall i \in N$, the berthing time of vessel i at berth k , *i.e.*, the time when the vessel berths
- $T_{o(k)}^k, \forall k \in M$, starting operation time of berth k , *i.e.*, the time when the first vessel berths at the berth
- $T_{d(k)}^k, \forall k \in M$, ending operation time of berth k , *i.e.*, the time when the last vessel departs at the berth

The assumptions considered in the mathematical model are the following:

- (a) Each berth $k \in M$ can only handle one vessel at a time
- (b) The service time of each vessel $i \in N$ is determined by the assigned berth $k \in M$
- (c) Each vessel $i \in N$ can be served only after its arrival time a_i
- (d) Each vessel $i \in N$ has to be served until its departure time b_i
- (e) Each vessel $i \in N$ can only be berthed at berth $k \in M$ after k becomes available at time step l^k
- (f) Each vessel $i \in N$ can only be berthed at berth $k \in M$ until k becomes unavailable at time step e^k

The time windows of the vessels and berths are defined by (c)-(f). The objective function (1) aims to minimize the total (weighted) service time of all the vessels, defined as the time elapsed between their arrival to the port and the completion of their services. When vessel i is not assigned to berth k , the corresponding term in the objective function is zero because $\sum_{j \in N \cup d(k)} x_{ij}^k = 0$ and $T_i^k = a_i$. A detailed mathematical formalization of the model can be consulted in [5].

$$\min \sum_{i \in N} \sum_{k \in M} v_i \left[T_i^k - a_i + t_i^k \sum_{j \in N \cup d(k)} x_{ij}^k \right] \quad (1)$$

In order to improve the understanding of the DBAP, we provide in Figure 1 an example of a berthing schedule. In the figure, a schedule and an assignment plan are shown for 6 vessels and 3 berths. The rectangles represent the vessels and inside each rectangle we display its corresponding service priority (p_i), used for establishing vessels priorities. The time windows of the vessels are represented by the lines

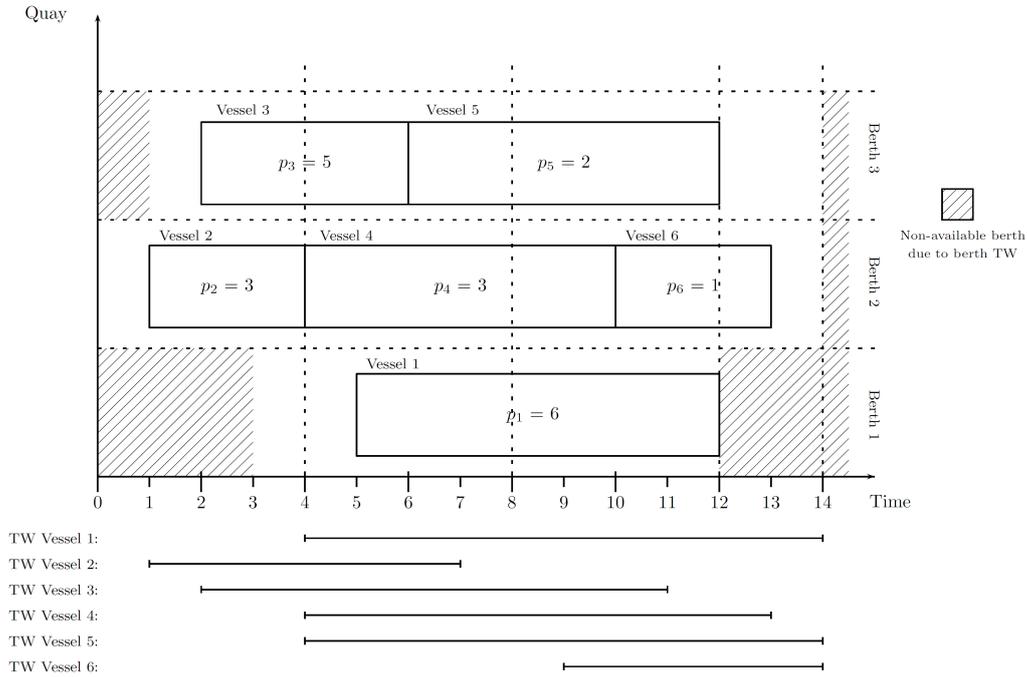


Figure 1: Example of solution for the DBAP with 6 vessels and 3 berths

at the bottom of the figure, starting from its arrival to the maximum departure time. Furthermore, the time window of each berth is limited by the non-hatched areas.

Table 1 reports the different handling times of each vessel depending on its assigned berth. For example, if vessel 1 is assigned to berth 1, its handling time would be equal to 7, which is shorter than the handling time of 8 that it would have at berth 2.

In the example depicted in Figure 1, the service priority of vessel 5 and 6 is 2 and 1, respectively. In this regard, as can be checked with their time windows, they have to wait for berthing in their respective assigned berths. This wait is justify if we consider that their wait for berthing have lesser impact on the objective function value than delaying other vessels like 3 and 4, for which their service priorities are 5 and 3, respectively.

Table 1: Vessels handling times depending on the allocated berth

Vessel	Berth 1	Berth 2	Berth 3
1	7	8	5
2	2	3	4
3	5	5	4
4	4	6	5
5	5	8	6
6	4	3	5

4 Decentralized Cooperative Metaheuristic

In this work, we propose Decentralized Cooperative Metaheuristic (DCM), which is a population-based approach. The individuals (*i.e.*, solutions of the problem at hand) of the population, S , exchange information among them following a line structure. That is, each individual has an adjacent individual on

each side (left and right). It should be noted that with exception of the first and last individuals that only have one adjacent, the remaining individuals have two adjacent individuals (one at the left and one at the right). Hence, individual 1, which is the first individual generated, is directly related to individual 2. On the other hand, individual 2 is directly related to individual 1 (left) and individual 3 (right), and so on. Moreover, as explained below, in DCM depending on the way the individuals share information among them, the population is distributed into groups, this is termed as *Relationship scheme*. DCM consists basically of two main components that are summarized as follows:

- (i) *Sharing*. Since the population is connected following a line structure. Each individual shares information with their adjacent individuals. The way the information is shared is based on a one-direction strategy, *i.e.*, for each pair of individuals, one of them gives information and the other receives it. The direction of the information flow, namely, determining who shares and who receives, is determined by means of the objective function value. That is, if an individual presents a better objective function value than its adjacent individual, it will directly share information with it. However, if both have the same objective function value, there will not be any information exchanged. Finally, if the individual has a worse objective function value, it will receive information from its adjacent. The information shared in the DCM approach applied to the DBAP consists of the best discarded solutions.
- (ii) *Distribution*. As explained above, once the individuals are created and their objective function values are calculated, a comparison of each individual with its adjacent according to the line structure is then performed. This comparison will allow to determine groups, that is, when an individual presents a worse objective function value than its adjacent individual, then it will directly form part of its group. However, if both individuals have the same objective function value, there will no exist any communication. Hence, when an individual receives information from another individual and this latter from another till a peak (the best individual in its group) this will conform a group. Hence, the groups are composed of those individuals that follow a better individuals around a peak solution in terms of objective function value.

Due to this, different types of individuals arise:

- A set of leading individuals (S_{lea}): This set is made up by the best individuals of each group
- A set of independent individuals (S_{ind}): This set includes all the individuals that have the same objective function value than their adjacent individuals. Thus, they do not exchange information with any other individuals
- A set of follower individuals (S_{fol}): This set contains the individuals that are neither leaders nor independents

The complete scheme of the population in terms of relationship among individuals is termed as *relationship scheme*. In this regard, in DCM, this complete distribution is not always the same, it can be re-determined if a given stopping condition is met. When a certain criterion is met, the distribution of the population is re-designed. In that case, all the individuals are compared again and a new division of the population into groups is performed. For the proposed solution approach, the stopping condition is met when the best solution found is improved or all group leader individuals could not improve their objective function value in the current iteration.

The pseudocode of DCM is depicted in Algorithm 1. The initial population composed of n_s individuals is randomly generated (line 1). The best solution is initialized to the best individual (line 2). The relationship scheme is determined by comparing the objective function value of each individual with its adjacent ones (line 4). Once the individuals are organized into groups, the search process is performed (lines 5 – 15) until the stopping condition for determining the relationship scheme is met. In this case, the stopping criteria used to recalculate the relationship scheme is set until the best solution known, s_{best} , is improved or any leading solution, $s \in S_{lea}$, is able to improve. In the search process, n_{on} random neighbour solutions are generated for each group leader and independent individual (line 7). If the best neighbour random solution leads to an improvement, the current solution is replaced by that one (line 8). Then, each individual $s \in S_{fol}$ generates $n_{on} - \delta$ neighbours and adds the δ best discarded neighbours received from its adjacent individual (lines 11 – 12). In the special case that an individual belongs to

Algorithm 1: Decentralized Cooperative Metaheuristic

```

1 Generate a set  $S$  of  $n_s$  individuals at random
2  $s_{best} \leftarrow$  best solution  $\in S$ 
3 while (stopping criterion is not met) do
4   Determine the groups according to Eq. 1
5   while (relationship scheme stopping condition is not met) do
6     for ( $\forall s \in S_{lea} \cup S_{ind}$ ) do
7       Generate  $n_{on}$  neighbour solutions for each  $s$ 
8       Move each individual to its best solution if leads to an improvement
9     end
10    for ( $\forall s \in S_{fol}$ ) do
11      Generate  $n_{on} - \delta$  neighbour solutions for each  $s$ 
12      Each  $s$  obtains  $\delta$  unused best neighbours from the solution in the front
13      Move each individual to its best solution if leads to an improvement
14    end
15  end
16 end
17 return  $s_{best}$ 

```

two groups it will receive $2 \cdot \delta$ solutions. If the best solution (generated by the individual or received from an individual in front of it) leads to an improvement, the solution is replaced by that one (line 13). The DCM search process is carried out while a stopping criterion is not met (line 3). For the DBAP, the search is performed until a maximum number of neighbours equal to $|N|^3$ has been generated by the individuals, where $|N|$ is the number of vessels, or a number n_{imp} of consecutive iterations without improvement of any individual has been performed.

4.1 DCM for the DBAP

The DCM implementation for the DBAP considers a solution s as a sequence composed of features, where a *feature*, is defined as indicated below:

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

Figure 2 shows an example of the solution structure for the planning example shown in Figure 1. Each berth is delimited by a 0. Thus, there will be M sub-sequences. The service order of each vessel is determined by its position in the subsequence. As can be seen in Figure 2, only vessel 1 is allocated at berth 1. At berth 2, the vessel 2 is the first vessel to be allocated. Once it departs from the berth, the next vessel to be allocated is vessel 4, and so on.

Figure 2: Solution structure of the BAP

1	0	2	4	6	0	3	5
---	---	---	---	---	---	---	---

The neighbourhoods used in this approach are obtained using the following movements:

- (a) Reinsertion-move, $N_1(s, \lambda)$: λ vessels are removed from a berth i and reinserted into another berth i' ($\forall i, i' \in M, i \neq i'$).
- (b) Interchange-move, $N_2(s)$: It consists of exchanging a vessel j assigned to berth i with a vessel j' assigned to berth i' ($\forall j, j' \in N, j \neq j', \forall i, i' \in M, i \neq i'$).

The individuals belonging to S_{lea} and S_{ind} produce n_{on} random neighbour solutions using the reinsertion movement, whereas the other individuals use the interchange-move. The DCM approach for the DBAP is performed until a maximum number of neighbours equals to $|N|^3$ has been generated, where

$|N|$ is the number of vessels, or a number n_{imp} of consecutive iterations without improvement of any individual has been performed.

5 Computational Results

This section is devoted to present the computational experiments carried out in order to assess the performance of the Decentralized Cooperative Metaheuristic. All the reported computational experiments were conducted on a computer equipped with an Intel 3.16 GHz and 4 GB of RAM. By taking into account the experiments carried out in this work, we identified the following parameter values for DCM: $n_{on} = 20$, $\delta = 3$, number of individuals $n_s = 30$, and stopping criteria of $max_N = |N|^3$ number of generated neighbour solutions by the individuals or $n_{imp} = 20$ consecutive iterations without improvement the best solution known.

The problem instances used for evaluating the proposed algorithm are provided in [5] and [11]. The instances from [5] were generated by taking into account a statistical analysis of the traffic and berth allocation data at the maritime container terminal of Gioia Tauro (Italy). The problem instances from [11] were generated according to [5] and address other realistic scenarios arising at container terminals. Moreover, to assess the performance of DCM, a comparison among the following algorithmic methods for the DBAP is provided:

- Generalised Set-Partitioning Problem mathematical model (GSPP) [3]
- Clustering Search with Simulated Annealing for generating initial solutions (CS-SA) [6]
- Particle Swarm Optimization (PSO) [13]
- Decentralized Cooperative Metaheuristic (DCM)

Table 2 shows the computational results obtained by applying these solution approaches. The mathematical formulation GSPP implemented by using CPLEX² in [2] provides the optimal solution in 17.92 seconds in the worst case. However, as highlighted in [11], GSPP can require large amounts of memory and computational time, depending on the complexity of the instances. In this regard, a Clustering Search with Simulated Annealing (CS-SA) that is able to provide the optimal solutions in all the cases and outperforms the GSPP time behaviour is presented in [6]. The results shown in the table related to this algorithm correspond to the best objective function values obtained and the average computational time required for 5 tests. Recently, [13] have proposed a Particle Swarm Optimization (PSO), which finds the optimal solutions with less computational effort. The results shown in the table correspond to the best objective function values provided by PSO and the computational time is the average time required for the 30 executions.

The comparison of DCM with the two different population-based solution approaches, CS-SA and PSO, reported in Table 2, shows that DCM presents a similar behaviour regarding the quality of the solutions within less computational time. In this regard, the comparison with PSO, which is a metaheuristic that follows a decentralized strategy inspired by the social behaviour of individuals inside swarms, would highlight the benefits of applying a cooperative structure within a decentralized scheme. Moreover, the comparison with CS-SA can give us an idea of the capability of DCM for identifying high promising regions since CS-SA locates promising regions through framing them by clusters. This could likely indicate that the way the regions are pointed out by DCM could be appropriate. However, this detail cannot be clearly claimed since the CS is used jointly with a Simulated Annealing and a Local Search process. Therefore, a more in-depth analysis of the individual contribution of those components would be required.

Moreover, we have also studied our algorithm with and without a local search applied to each leader solution once their search process is over. The aim of applying a local search after the algorithm has been executed seeks to analyse if it is able to point out high promising regions in the search space. As can be seen in Table 2, the bold numbers indicate those solutions where DCM without local search is able to point out 21 regions, where the optimal solution obtained after applying the local search to each leader

²<http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>

Table 2: Computational results for the instances provided in [5]. Bold numbers indicate those solutions that after applying a local search it is possible to reach the optimal solution

Instance	GSPP		CS			PSO			DCM						
	Opt.	t(s.)	Best	Gap (%)	t(s.)	Best	Gap (%)	t(s.)	w/LS			w/o LS			
	Best	Gap (%)	t(s.)	Best	Gap (%)	t(s.)	Best	Gap (%)	t(s.)	Best	Gap (%)	t(s.)	Best	Gap (%)	t(s.)
i01	1409	17.92	1409	0.00	12.47	1409	0.00	11.11	1409	0.00	5.95	1420	0.78	3.25	
i02	1261	15.77	1261	0.00	12.59	1261	0.00	7.89	1261	0.00	4.15	1261	0.00	3.29	
i03	1129	13.54	1129	0.00	12.64	1129	0.00	7.48	1129	0.00	4.18	1130	0.09	3.20	
i04	1302	14.48	1302	0.00	12.59	1302	0.00	6.03	1302	0.00	4.25	1302	0.00	3.07	
i05	1207	17.21	1207	0.00	12.68	1207	0.00	5.84	1207	0.00	3.21	1207	0.00	2.86	
i06	1261	13.85	1261	0.00	12.56	1261	0.00	7.67	1261	0.00	4.04	1262	0.08	2.90	
i07	1279	14.60	1279	0.00	12.63	1279	0.00	7.5	1279	0.00	3.36	1280	0.08	2.97	
i08	1299	14.21	1299	0.00	12.57	1299	0.00	9.94	1299	0.00	4.96	1304	0.38	3.10	
i09	1444	16.51	1444	0.00	12.58	1444	0.00	4.25	1444	0.00	5.25	1446	0.14	3.31	
i10	1213	14.16	1213	0.00	12.61	1213	0.00	5.2	1213	0.00	3.46	1213	0.00	3.20	
i11	1368	14.13	1368	0.00	12.58	1368	0.00	10.52	1368	0.00	5.21	1374	0.44	3.39	
i12	1325	15.60	1325	0.00	12.61	1325	0.00	12.92	1325	0.00	4.62	1330	0.38	3.38	
i13	1360	13.87	1360	0.00	12.58	1360	0.00	11.97	1360	0.00	3.76	1362	0.15	3.47	
i14	1233	15.60	1233	0.00	12.56	1233	0.00	7.11	1233	0.00	4.14	1233	0.00	3.04	
i15	1295	13.52	1295	0.00	12.61	1295	0.00	8.3	1295	0.00	4.31	1295	0.00	3.40	
i16	1364	13.68	1364	0.00	12.67	1364	0.00	8.48	1364	0.00	4.89	1368	0.29	3.94	
i17	1283	13.37	1283	0.00	13.80	1283	0.00	5.66	1283	0.00	3.09	1283	0.00	2.68	
i18	1345	13.51	1345	0.00	14.46	1345	0.00	8.02	1345	0.00	4.14	1347	0.15	3.36	
i19	1367	14.59	1367	0.00	13.73	1367	0.00	11.42	1367	0.00	5.93	1374	0.51	4.03	
i20	1328	16.64	1328	0.00	12.82	1328	0.00	12.28	1328	0.00	5.60	1334	0.45	3.97	
i21	1341	13.37	1341	0.00	12.68	1341	0.00	7.11	1341	0.00	5.54	1346	0.37	3.51	
i22	1326	15.24	1326	0.00	12.62	1326	0.00	7.94	1326	0.00	4.97	1333	0.53	3.13	
i23	1266	13.65	1266	0.00	12.62	1266	0.00	7.25	1266	0.00	4.01	1266	0.00	3.75	
i24	1260	15.58	1260	0.00	12.64	1260	0.00	5.67	1260	0.00	4.90	1261	0.08	3.61	
i25	1376	15.80	1376	0.00	12.62	1376	0.00	7.13	1376	0.00	5.54	1381	0.36	3.39	
i26	1318	15.38	1318	0.00	12.62	1318	0.00	7.44	1318	0.00	4.92	1325	0.53	3.52	
i27	1261	15.52	1261	0.00	12.64	1261	0.00	6.16	1261	0.00	4.00	1261	0.00	3.15	
i28	1359	16.22	1359	0.00	12.71	1359	0.00	11.52	1359	0.00	5.56	1363	0.29	3.40	
i29	1280	15.30	1280	0.00	12.62	1280	0.00	8.11	1280	0.00	5.82	1282	0.16	3.25	
i30	1344	16.52	1344	0.00	12.58	1344	0.00	7.13	1344	0.00	5.76	1350	0.45	3.52	
	14.98		1306.77		12.76	1306.77		8.17	1306.77		4.65	1309.77		3.33	

individual. Concerning the required computational effort, DCM is able to improve the computational time if compared with the best solution approaches presented in the literature.

Moreover, as indicated in [10] the GSPP mathematical formulation implemented in CPLEX is not able to provide even a feasible solution for some complex instances where other characteristics are considered. In this regard, we are interested in assessing the behaviour of DCM in such kind of instances. In doing so, a representative set of some of the largest instances proposed in [11] has been tackled. The dimensions of the set of instances are 60 vessels and 5 berths. For evaluating the performance of DCM, a comparison among the best algorithmic methods used for those instances is provided:

- GSPP mathematical model [11]
- Tabu Search (T^2S^*+PR) [11]
- Decentralized Cooperative Metaheuristic (DCM)

Table 3 shows the computational results for the algorithms listed above. A column, *MIN*, with the best solution known for those instances is also included. As can be seen, GSPP is not able even to provide a feasible solution because it runs out of memory. Regarding the approximate solution approaches, DCM presents a better performance on average within an almost similar time requirement than the best approach ($T^2S^* + PR$) reported in the literature. Moreover, after analysing the use of the local search method, DCM points out 7/10 regions where the best solution known can be found after applying a local search. In this regard, DCM provides two new best objective function values that have not been reached before, namely, instances *i04* and *i10*.

Table 3: Computational results for the instances provided in [11]. Bold numbers indicate the best objective function value

Instance	GSPP	MIN	T2S*+PR			DCM					
						w/LS			w/o LS		
			Best	Gap (%)	t (s.)	Best	Gap (%)	t (s.)	Best	Gap (%)	t (s.)
i01	—	5753	5753	0.00	3.12	5759	0.10	3.71	5765	0.21	1.37
i02	—	6884	6884	0.00	3.20	6884	0.00	3.96	6886	0.03	1.45
i03	—	6780	6780	0.00	4.25	6780	0.00	4.17	6796	0.24	1.49
i04	—	5092	5105	0.26	2.30	5092	0.00	3.18	5102	0.20	1.48
i05	—	6715	6715	0.00	3.18	6715	0.00	3.68	6721	0.09	1.24
i06	—	6616	6616	0.00	3.53	6618	0.03	3.41	6632	0.24	1.29
i07	—	6011	6011	0.00	4.75	6011	0.00	3.58	6031	0.33	1.54
i08	—	4385	4385	0.00	3.77	4385	0.00	3.24	4396	0.25	1.52
i09	—	5235	5235	0.00	3.99	5237	0.04	3.15	5251	0.31	1.47
i10	—	7255	7281	0.36	3.62	7255	0.00	3.91	7266	0.15	1.56
		6072.6	6076.5	0.06	3.57	6073.6	0.02	3.60	6084.6	0.20	1.44

6 Conclusions and Further Research

The Dynamic Berth Allocation Problem (DBAP) has been addressed in this work. In order to efficiently solve it, we propose Decentralized Cooperative Metaheuristic (DCM). It is based on a decentralized grouping strategy to divide a population of individuals into groups. The individuals in the same group cooperate by interchanging information. This grouping strategy improves the diversification of the search as well as the intensification in some regions of the search space through the sum of efforts among the individuals of the same group. Furthermore, the constrained relation for sharing information among individuals through the division of groups allows to reduce resources in comparison to ‘all to all’ communication.

It is concluded from the computational experimentation that the proposed algorithm is able to provide the optimal solutions within reasonable computational time for the instances proposed in [5]. In

this regard, the time advantage makes DCM suitable as a resolution method for being applied either individually or included into integrated schemes where the berth allocation is required. DCM is also appropriate for pointing out high promising regions in the search space.

Furthermore, the computational results show that DCM exhibits a better performance than other optimization algorithms presented in the literature for the DBAP. In this sense, the comparison with PSO and CS-SA remarks the benefits of applying a decentralized cooperative scheme for improving the processing times and detecting promising regions in the search space. Moreover, the experimentation over a representative set of instances, where the GSPP formulation implemented in CPLEX is not able to provide any feasible solution, shows that DCM is able to provide feasible solutions within small computational effort. In this regard, the comparison with the best approaches used for those instances indicates that DCM presents a better performance on average and provides two new best known solutions.

Multitude of lines are open for further research. In the future we are going to test the performance of DCM in other heterogeneous transportation problems, such as Vehicle Routing Problem due to its generalist standpoint. In this regard, we are also going to study different ways to exchange information among individuals and establish groups.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness (project TIN2012-32608). Eduardo Lalla-Ruiz and Christopher Expósito-Izquierdo thank the Canary Government the financial support they receive through their post-graduate grants.

References

- [1] C. Bierwirth and F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3):615 – 627, 2010.
- [2] K. Buhrkal, S. Zuglian, S. Ropke, J. Larsen, and R. Lusby. Models for the discrete berth allocation problem: a computational comparison. *Transportation Research Part E*, 47(4):461–473, 2011.
- [3] C.G. Christensen and C.T. Holst. Berth allocation in container terminals. Master’s thesis, Department of Informatics and Mathematical Modelling, Technical university of Denmark, 2008. In Danish.
- [4] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Chapter 4 maritime transportation. In Cynthia Barnhart and Gilbert Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 189 – 284. Elsevier, 2007.
- [5] 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.
- [6] R. Martins de Oliveira, G. Regis-Mauri, and L.A. Nogueira-Lorena. Clustering search for the berth allocation problem. *Expert Systems with Applications*, 39(5):5499 – 5505, 2012.
- [7] J. P. Gutiérrez-Castro, B. Melián Batista, J. A. Moreno-Pérez, J. M. Moreno-Vega, and J. Ramos-Bonilla. Codea: An architecture for designing nature-inspired cooperative decentralized heuristics. In Natalio Krasnogor, Giuseppe Nicosia, Mario Pavone, and David Pelta, editors, *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, volume 129 of *Studies in Computational Intelligence*, pages 189–198. Springer Berlin Heidelberg, 2008.
- [8] A. Imai, K. Nagaiwa, and W. T. Chan. Efficient planning of berth allocation for container terminals in asia. *Journal of Advanced Transportation*, 31(1):75–94, 1997.
- [9] A. Imai, E. Nishimura, and S. Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4):401–407, 2001.

-
- [10] E. Lalla-Ruiz, C. Expósito-Izquierdo, B. Melián-Batista, and J. M. Moreno-Vega. A metaheuristic approach for the seaside operations in maritime container terminals. In *IWANN (Part II)*, volume 7903 of *Lecture Notes in Computer Science*. Springer, 2013.
 - [11] E. Lalla-Ruiz, B. Melián-Batista, and J. M. Moreno-Vega. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence*, 25(6):1132 – 1141, 2012.
 - [12] M. F. Monaco and M. Sammarra. The berth allocation problem: a strong formulation solved by a lagrangean approach. *Transportation Science*, 41(2):265–280, 2007.
 - [13] C. Ting, K. Wu, and H. Chou. Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications*, (0):–, 2013.
 - [14] N. Umang, M. Bierlaire, and I. Vacca. Exact and heuristic methods to solve the berth allocation problem in bulk ports. *Transportation Research Part E: Logistics and Transportation Review*, 54(0):14 – 31, 2013.
 - [15] H. J. Yeo. Competitiveness of asian container terminals. *The Asian Journal of Shipping and Logistics*, 26(2):225 – 246, 2010.