The Knowledge Engineering Review, Vol. 31:5, 498–507.
doi:10.1017/S0269888916000266
First published online 18 January 2017

A cooperative search for berth scheduling

EDUARDO LALLA-RUIZ, BELÉN MELIÁN-BATISTA and
JOSÉ MARCOS MORENO-VEGA

Department of Computer Engineering and Systems, University of La Laguna, 38271-La Laguna, Spain;
e-mail: elalla@ull.es, mbmelia@ull.es, jmmoreno@ull.es

Abstract

With the growing demand of freight transport by means of container vessels as well as the important competition among terminals, managers and stakeholders seek to improve the exploitation of the container terminal resources efficiently. In this context, arises the Berth Allocation Problem, which aims to allocate and schedule incoming vessels along the quay. Its appropriate solution plays a relevant role in enhancing the terminal productivity. Thus, for addressing this problem, we propose a cooperative search, where the individuals are organized into groups and each member 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. The computational experiments for this problem reveal that our approach reports high-quality solutions and identifies promising regions within the search space in short computational times.

1 Introduction

In seaborne transportation, maritime container terminals arise as crucial infrastructures aimed at facing an increasing volume of containers within the international freight network. In this context, according to the UNCTAD[1], the international maritime container trade has greatly grown over the last decades. To cope with this trend, terminal managers have to provide on-time and fast services to container vessels. The above, at the same time, allows to enhance the terminal competitiveness based on the fact that one of the most widespread indicators for assessing the competitiveness of a maritime container terminal is the time required to serve those container vessels arriving to the terminal (Yeo, 2010). For this reason, an inefficient utilization of some key and scarce resources, like berths, could produce a loss of competitiveness and have a negative impact on other operations taking place at the container yard and land-side, thus giving rise to a poor overall productivity of the container terminal.

The previous discussion 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 BAP (DBAP) introduced by Cordeau *et al.* (2005), which considers berth and vessel time windows as well as heterogeneous vessel service times stemming from the assigned berth.

In order to solve the DBAP, this work proposes a cooperative search termed as 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, whereas the search is performed. As indicated by Gutiérrez-Castro *et al.* (2008), the 'all to all' communication in working

[1] United Nations Conference on Trade and Development, `http://unctad.org`

systems is not appropriate because it demands too many computational resources. Therefore, the way the information is shared in DCM pursuits a decentralized grouping strategy. Namely, during the search, the individuals only share information with their group partners.

The goal of this work is two-fold, on the one hand, we aim to assess the performance of DCM as well as provide high-quality solutions by means of short computational times for supporting berth scheduling operations 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 by Christensen and Holst (2008) 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 that it requires less 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 concise literature review of the BAP is presented in the following section. Then, the mathematical formulation of the DBAP used in this work is described. In the next section, the algorithm proposed for addressing the BAP is described. Later, the computational experience carried out and a comparative summary are presented. Finally, some conclusions and several lines for further research are drawn in the last section.

2 Literature review

The 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 for this problem. Depending on how the quay is modeled, 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 (Cordeau *et al*., 2005; Umang *et al*., 2013) 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 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 Christiansen *et al*. (2007) and Bierwirth and Meisel (2010).

One of the most relevant approaches is the DBAP. It was first formulated by Imai *et al*. (2001) as an extension of the model proposed in Imai *et al*. (1997) for the Static BAP. Alternative formulations for the dynamic problem have been proposed and studied by Cordeau *et al*. (2005), Monaco and Sammarra (2007) and Christensen and Holst (2008). These models are described and compared in Buhrkal *et al*. (2011). The main conclusion extracted from the latter work is that the model presented by Christensen and Holst is superior to the other models in terms of the time performance. Namely, it is able to provide the optimal solutions within shorter computational times for the largest instances proposed in the work of Cordeau *et al*. (2005).

Recently, Lalla-Ruiz *et al*. (2012) presented an efficient Tabu Search metaheuristic with Path-Relinking for solving the DBAP. They also proposed a benchmark suite of instances for which the model by Christensen and Holst (2008) does not provide feasible solutions within a time limit. de Oliveira *et al*. (2012) 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 by Cordeau *et al*. (2005). In this regard, Ting *et al*. (2014) propose a Particle Swarm Optimization (PSO) 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 by Cordeau *et al*. (2005) which is modeled as a Multi-Depot Vehicle Routing Problem with Time-Windows. Thus, based on that formulation, 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 by Cordeau *et al*. (2005) 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, departure time of vessel $i \in N$.
- l^k, e^k, start, 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 represents 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, that is, the time when the vessel berths.
- $T_{o(k)}^k,\ \forall\ k \in M$, starting operation time of berth k, that is, the time when the first vessel berths at the berth.
- $T_{d(k)}^k,\ \forall\ k \in M$, ending operation time of berth k, that is, the time when the last vessel departs from its assigned berth.

Furthermore, in the DBAP the following assumptions are considered:

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 (TWs) of the vessels and berths are defined by (c)–(f). The objective function (1) aims to minimize the total (weighted) service time for all the vessels, defined as the time elapsed between their arrival to the port and the completion of their handling. When 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 Cordeau *et al*. (2005).

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

In order to facilitate the understanding of the DBAP, we provide in Figure 1 an example of solution. In the figure, a schedule and an assignment plan for six vessels within three berths are shown. The rectangles indicate the vessels and inside each rectangle we display its corresponding service priority (p_i). The time windows of the vessels are represented by the lines at the bottom of the figure. In this case, for example, vessel 1 arrives at time step 4 and it should be served before time step 14. Moreover, the available time window of each berth is limited by the unshaded areas. Table 1 reports the different handling times for each vessel depending on the assigned berth. For example, if vessel 1 would be assigned to berth 1, its handling time would be equal to 8, which is larger than the handling time of 6 that it has at berth 2. Finally, the objective function value of the given solution example is 93.

4 A cooperative search for the DBAP

In this work, we propose a cooperative search termed as DCM, which is a population-based approach inspired by Migrating Birds Optimization (MBO, Duman *et al*., 2012). In DCM, the population of

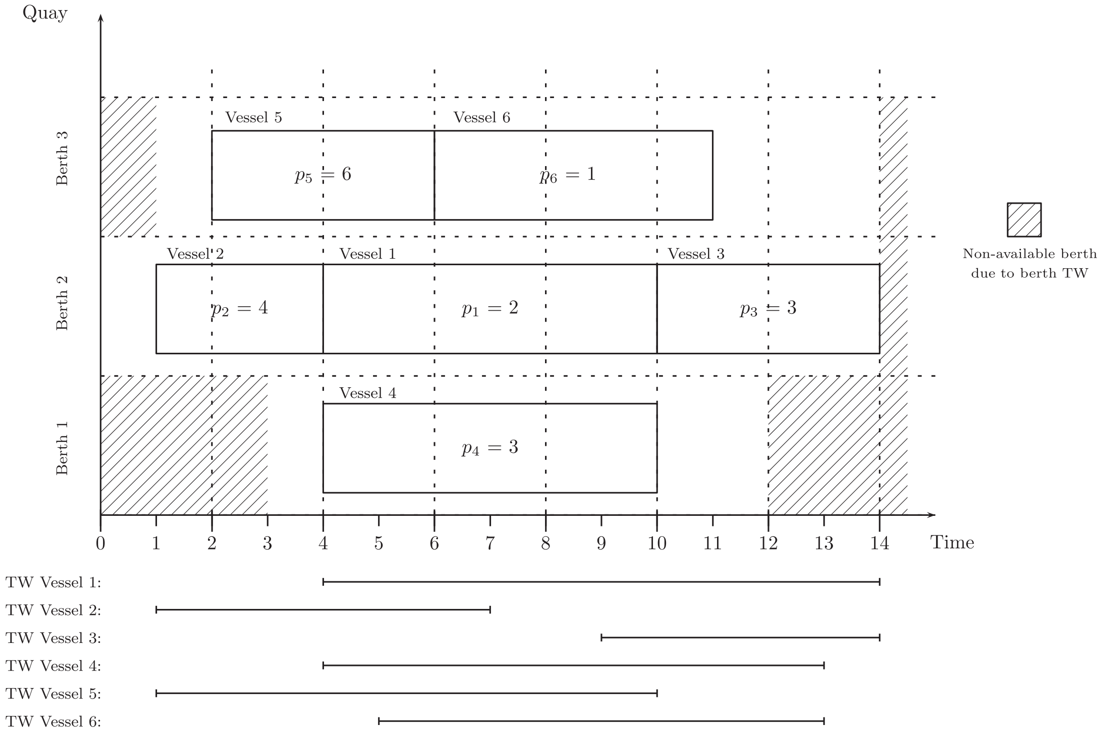

Figure 1 Example of solution for the DBAP with six vessels and three berths

Table 1 Vessels handling time depending on the allocated berth

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

individuals *S* is organized into groups and each individual performs a search procedure. Moreover, while the search is being performed, the individuals exchange information with other individuals belonging to their same group. The way the information is shared is almost constrained to the members of the same group in the sense that only at most two members of a group can share information with other groups. Nevertheless, some individuals may belong to more than one group and, therefore, they share information with more than one group, subsequently, DCM consists on the following basically of three main components that are summarized in the following points:

1. In order to organize the individuals into groups a given criterion is used. That criterion can be established according to different measurements or strategies such as the objective function value, solution structure, similarity, etc. In the context of the DBAP, we have established the objective function value provided by Equation (1) as the criterion to be used. Moreover, the adjacency of each individual is determined by their generation order. Once the individuals are created and their objective function values are calculated, a comparison among the individuals and their adjacent ones based on the objective function value is performed. This way, when an individual presents a worse objective function value than its adjacent individual, then it will directly form part of its adjacent group.

However, if both of them have the same objective function value, there will not exist any communication.

2. The individuals share information with their adjacent partners. There is no direct exchange between individuals of different groups, but indirect exchanges may arise due to individuals belonging to more than one group. The information shared in DCM consists of the best discarded solutions and the way the individuals exchange information depends on their objective function value. That is, if an individual presents a better objective function value than its adjacent solution, it will directly share information with it by providing its best discarded solutions. However, if both of them 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 solution.
3. The division of the population into groups and the way the information is exchanged among them is not always the same, it can be re-determined if a given reorganization condition is met. That is, every time that condition is met, the distribution of the population is re-designed.

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 distribution of the population in groups is determined by considering and comparing the objective function value of each individual and its adjacent ones (line 4). Once the individuals are organized into groups, the search process is performed (lines 5–15) until the reorganization condition is met. In this case, the reorganization condition used for the DBAP is set until the best solution known, s_{best}, is improved or any best solution of a group, $s \in L$, is able to improve. In the search process, n_{on} random neighbor solutions are generated (line 7) for each group solution belonging to the set containing the best solutions of each group, L, or belonging to the independent individuals set I (line 6). If the best neighbor random solution leads to an improvement, the current solution is replaced by that one (line 8). Then, each individual not being a leader or independent $s \in F$ generates $n_{\text{on}} - \delta$ neighbors and adds the δ best discarded neighbors received from its adjacent individual (lines 11–12). In the special case that an individual belongs to two groups, it will receive $2 \cdot \delta$ solutions. If the best solution (generated by the individual or received from a previous individual) 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 neighbors 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.

Algorithm 1: Decentralized Cooperative Metaheuristic

```
1  Generate a set S of n_s individuals at random
2  s_best ← best solution ∈ S
3  while (stopping criterion is not met) do
4      Determine the groups according to Eq. 1
5      while (reorganization condition is not met) do
6          for (∀s ∈ L ∪ I) 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 (∀s ∈ F) do
11             Generate n_on − δ neighbour solutions for each s
12             Each s obtains δ unused best neighbours from the previous solution
13             Move each individual to its best solution if leads to an improvement
14         end
15     end
16 end
17 return s_best
```

4	0	2	1	3	0	5	6

Figure 2 Solution structure for the Berth Allocation Problem

4.1 Solution structure for the DBAP

In the context of the DBAP, the DCM implementation for this problem considers a solution *s* as a sequence composed by 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 depicted 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 4 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 1, and so on.

4.2 Neighborhood structures for the DBAP

The neighborhoods used in this approach are the following:

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 best individuals of each group as well as the independent individuals produce n_{on} random neighbor 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 neighbors 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 our proposed approach. 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: number of generated neighbour solutions per individual (n_{on}) = 20, number of shared solutions (δ) = 3, number of individuals $n_s = 30$, and stopping criteria of $max_N = |N|^3$ as the number of generated neighbor solutions by the individuals or $n_{\text{imp}} = 20$ consecutive iterations without improvement from any individual of the population.

The problem instances used for evaluating our proposed algorithm are the largest ones provided in Cordeau *et al.* (2005) consisting of 60 vessels and 13 berths. As described in Cordeau *et al.* (2005), those instances were generated taking into account a statistical analysis of the traffic and berth allocation data at the maritime container terminal of Gioia Tauro (Italy). Furthermore, in order to compare our algorithmic proposal, the following approaches have been considered: (i) the mathematical model proposed by Christensen and Holst (2008) implemented in CPLEX, (ii) PSO proposed by Ting *et al.* (2014), and (iii) an adaptation MBO algorithm for this problem.

Table 2 shows the computational results obtained by applying these solution approaches. The mathematical formulation implemented in CPLEX[2] by Buhrkal *et al.* (2011) provides the optimal solution in 17.92 seconds in the worst case. However, as discussed in the work of Lalla-Ruiz *et al.* (2012), CPLEX may require large amounts of memory and computational time, depending on the complexity of the

[2] `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

Table 2 Computational results with the best solution approach provided in the literature, Particle Swarm Optimization ((PSO, Ting *et al.* (2014)) for the largest instances (60 vessels and 13 berths) provided by Cordeau *et al.* (2005)

	CPLEX		PSO			DCM					
						w/LS			w/o LS		
	Optimization	Time	Best	Gap (%)	Time	Best	Gap (%)	Time	Best	Gap (%)	Time
i01	1409	17.92	1409	0.00	11.11	1409	0.00	5.95	**1420**	0.78	3.25
i02	1261	15.77	1261	0.00	7.89	1261	0.00	4.15	1261	0.00	3.29
i03	1129	13.54	1129	0.00	7.48	1129	0.00	4.18	**1130**	0.09	3.20
i04	1302	14.48	1302	0.00	6.03	1302	0.00	4.25	1302	0.00	3.07
i05	1207	17.21	1207	0.00	5.84	1207	0.00	3.21	1207	0.00	2.86
i06	1261	13.85	1261	0.00	7.67	1261	0.00	4.04	**1262**	0.08	2.90
i07	1279	14.60	1279	0.00	7.50	1279	0.00	3.36	**1280**	0.08	2.97
i08	1299	14.21	1299	0.00	9.94	1299	0.00	4.96	**1304**	0.38	3.10
i09	1444	16.51	1444	0.00	4.25	1444	0.00	5.25	**1446**	0.14	3.31
i10	1213	14.16	1213	0.00	5.20	1213	0.00	3.46	1213	0.00	3.20
i11	1368	14.13	1368	0.00	10.52	1368	0.00	5.21	**1374**	0.44	3.39
i12	1325	15.60	1325	0.00	12.92	1325	0.00	4.62	**1330**	0.38	3.38
i13	1360	13.87	1360	0.00	11.97	1360	0.00	3.76	**1362**	0.15	3.47
i14	1233	15.60	1233	0.00	7.11	1233	0.00	4.14	1233	0.00	3.04
i15	1295	13.52	1295	0.00	8.30	1295	0.00	4.31	1295	0.00	3.40
i16	1364	13.68	1364	0.00	8.48	1364	0.00	4.89	**1368**	0.29	3.94
i17	1283	13.37	1283	0.00	5.66	1283	0.00	3.09	1283	0.00	2.68
i18	1345	13.51	1345	0.00	8.02	1345	0.00	4.14	**1347**	0.15	3.36
i19	1367	14.59	1367	0.00	11.42	1367	0.00	5.93	**1374**	0.51	4.03
i20	1328	16.64	1328	0.00	12.28	1328	0.00	5.60	**1334**	0.45	3.97
i21	1341	13.37	1341	0.00	7.11	1341	0.00	5.54	**1346**	0.37	3.51
i22	1326	15.24	1326	0.00	7.94	1326	0.00	4.97	**1333**	0.53	3.13
i23	1266	13.65	1266	0.00	7.25	1266	0.00	4.01	1266	0.00	3.75
i24	1260	15.58	1260	0.00	5.67	1260	0.00	4.90	**1261**	0.08	3.61
i25	1376	15.80	1376	0.00	7.13	1376	0.00	5.54	**1381**	0.36	3.39
i26	1318	15.38	1318	0.00	7.44	1318	0.00	4.92	**1325**	0.53	3.52
i27	1261	15.52	1261	0.00	6.16	1261	0.00	4.00	1261	0.00	3.15
i28	1359	16.22	1359	0.00	11.52	1359	0.00	5.56	**1363**	0.29	3.40
i29	1280	15.30	1280	0.00	8.11	1280	0.00	5.82	**1282**	0.16	3.25
i30	1344	16.52	1344	0.00	7.13	1344	0.00	5.76	**1350**	0.45	3.52
Avg.	1306.77	14.98	1306.77		8.17	1306.77		4.65	1309.77		3.33

DCM = decentralized cooperative metaheuristic, MBO = migrating birds optimization, LS = local search.
Bold numbers indicate those cases where, after applying a local search, the optimal solution was provided.

instances. Recently, Ting *et al.* (2014) have proposed a 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 30 executions.

The comparison of DCM with the population-based approach based on PSO reported in Table 2 shows that DCM presents a similar performance 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 behavior of individuals inside swarms, would highlight the benefits of applying a cooperative structure within a decentralized scheme.

Table 3 shows the comparison of the MBO approach for the DBAP with DCM. The rationale behind including this algorithm is to compare the behavior of DCM with MBO. Moreover, both algorithms are studied with and without a local search (LS) applied to each best solution once their search process is over. The aim of applying a local search after the algorithms have been executed seeks to analyze if they are able

Table 3 Computational results with the Migrating Birds Optimization (MBO) algorithm for the largest instances (60 vessels and 13 berths) provided by Cordeau *et al*. (2005)

			MBO						DCM					
	CPLEX		w/LS			w/o LS			w/LS			w/o LS		
	Optimization	Time	Best	Gap (%)	Time	Best	Gap (%)	Time	Best	Gap (%)	Time	Best	Gap (%)	Time
i01	1409	17.92	1411	0.14	3.42	1441	2.27	2.72	1409	0.00	5.95	**1420**	0.78	3.25
i02	1261	15.77	1261	0.00	3.52	**1265**	0.32	2.43	1261	0.00	4.15	1261	0.00	3.29
i03	1129	13.54	1129	0.00	3.63	**1144**	1.33	2.51	1129	0.00	4.18	**1130**	0.09	3.20
i04	1302	14.48	1302	0.00	3.81	**1304**	0.15	2.43	1302	0.00	4.25	1302	0.00	3.07
i05	1207	17.21	1207	0.00	3.13	**1212**	0.41	2.12	1207	0.00	3.21	1207	0.00	2.86
i06	1261	13.85	1261	0.00	3.46	**1272**	0.87	2.42	1261	0.00	4.04	**1262**	0.08	2.90
i07	1279	14.60	1279	0.00	3.05	**1291**	0.94	2.09	1279	0.00	3.36	**1280**	0.08	2.97
i08	1299	14.21	1299	0.00	3.30	**1313**	1.08	2.21	1299	0.00	4.96	**1304**	0.38	3.10
i09	1444	16.51	1444	0.00	3.48	**1457**	0.90	2.29	1444	0.00	5.25	**1446**	0.14	3.31
i10	1213	14.16	1213	0.00	3.40	**1219**	0.49	2.44	1213	0.00	3.46	1213	0.00	3.20
i11	1368	14.13	1370	0.15	3.41	1380	0.88	2.16	1368	0.00	5.21	**1374**	0.44	3.39
i12	1325	15.60	1330	0.38	3.54	1344	1.43	2.54	1325	0.00	4.62	**1330**	0.38	3.38
i13	1360	13.87	1360	0.00	3.59	**1372**	0.88	2.45	1360	0.00	3.76	**1362**	0.15	3.47
i14	1233	15.60	1233	0.00	3.27	**1242**	0.73	2.28	1233	0.00	4.14	1233	0.00	3.04
i15	1295	13.52	1295	0.00	3.43	**1306**	0.85	2.28	1295	0.00	4.31	1295	0.00	3.40
i16	1364	13.68	1367	0.22	4.14	1394	2.20	2.51	1364	0.00	4.89	**1368**	0.29	3.94
i17	1283	13.37	1283	0.00	2.63	1283	0.00	1.94	1283	0.00	3.09	1283	0.00	2.68
i18	1345	13.51	1345	0.00	3.38	**1350**	0.37	2.18	1345	0.00	4.14	**1347**	0.15	3.36
i19	1367	14.59	1372	0.37	3.81	1390	1.68	2.57	1367	0.00	5.93	**1374**	0.51	4.03
i20	1328	16.64	1329	0.08	3.55	1352	1.81	2.39	1328	0.00	5.60	**1334**	0.45	3.97
i21	1341	13.37	1343	0.15	3.93	1359	1.34	2.65	1341	0.00	5.54	**1346**	0.37	3.51
i22	1326	15.24	1326	0.00	3.38	**1348**	1.66	2.25	1326	0.00	4.97	**1333**	0.53	3.13
i23	1266	13.65	1266	0.00	3.47	**1283**	1.34	2.28	1266	0.00	4.01	1266	0.00	3.75
i24	1260	15.58	1260	0.00	3.51	**1264**	0.32	2.37	1260	0.00	4.90	**1261**	0.08	3.61
i25	1376	15.80	1377	0.07	3.30	1392	1.16	2.00	1376	0.00	5.54	**1381**	0.36	3.39
i26	1318	15.38	1319	0.08	3.45	1333	1.14	2.20	1318	0.00	4.92	**1325**	0.53	3.52
i27	1261	15.52	1261	0.00	3.16	**1273**	0.95	2.27	1261	0.00	4.00	1261	0.00	3.15
i28	1359	16.22	1361	0.15	3.42	1372	0.96	2.50	1359	0.00	5.56	**1363**	0.29	3.40
i29	1280	15.30	1281	0.08	3.77	1289	0.70	2.60	1280	0.00	5.82	**1282**	0.16	3.25
i30	1344	16.52	1349	0.37	3.78	1380	2.68	2.48	1344	0.00	5.76	**1350**	0.45	3.52
Avg.	1306.77	14.98	1307.77		3.47	1320.80		2.35	1306.77		4.65	1309.77		3.33

MBO = migrating birds optimization; DCM = decentralized cooperative metaheuristic; LS = local search.
Bold numbers indicate those cases where, after applying a local search, the optimal solution was provided.

to point out high promising regions in the search space. As can be seen in the table, the bold numbers indicate those cases where the solution provided by DCM without local search is able to point out 21 regions that contained the optimal solution after a local search to each best individual in each group is applied. In this context, MBO is able to provide 17 regions where the optimal solution is included.

Concerning the required computational effort, DCM is able to improve the computational time when it is compared with the best solution approaches presented in the literature. Nevertheless, from the comparison with MBO, it can be pointed out that the latter requires less computational time. Since both algorithms are executed under the same stopping criteria, it could likely indicate a premature convergence of MBO.

6 Conclusions and further research

The DBAP has been addressed in this work. In order to efficiently solve it, we propose the application of a cooperative search termed as DCM. It is based on a decentralized grouping strategy for dividing a population of individuals into groups. The individuals within the same group cooperate by exchanging 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 can be concluded from the computational experimentation that the proposed algorithm is able to provide the optimal solutions within reasonable computational time for the instances proposed by Cordeau *et al*. (2005). In this regard, the time advantage exhibited by our proposed approach makes it suitable as a resolution method for being applied either individually or included into integrated schemes where solving the berth allocation is required. In addition, DCM is also appropriate for pointing out high promising regions in the search space. Moreover, the computational results show that DCM exhibits a better performance than other optimization algorithms presented in the literature for the DBAP. Finally, it is worth to mention that the comparison with the best population-based approach based on PSO highlights the benefits of applying a decentralized cooperative scheme for improving the solving times and detecting promising regions in the search space.

On the basis of the contributions presented in this paper, the subsequent stage of our research will be focused on the analysis of the influence of the different components composing our proposed approach as well as its application to other relevant optimization problems at the seaside of maritime container terminals.

Acknowledgments

The preparation of this work has been partially funded by the Spanish Ministry of Economy and Competitiveness (projects TIN2012-32608, TIN2015-70226-R). Eduardo Lalla-Ruiz thanks the Canary Government the financial support he receives through his doctoral grant.

This work was previously submitted to the COPLAS workshop in 2014 and thanks to that included in this special issue.

References

Bierwirth, C. & Meisel, F. 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research* **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. *Transportation Research Part E* **47**(4), 461–473.

Christensen, C. & Holst, C. 2008. *Berth Allocation in Container Terminals*. Master’s thesis, Department of Informatics and Mathematical Modelling, Technical University of Denmark (in Danish).

Christiansen, M., Fagerholt, K., Nygreen, B. & Ronen, D. 2007. Chapter 4 maritime transportation. In *‘Transportation’, Volume 14 of Handbooks in Operations Research and Management Science,* Barnhart, C. & Laporte, G. (eds). Elsevier, 189–284.

Cordeau, J.-F., Laporte, G., Legato, P. & Moccia, L. 2005. Models and tabu search heuristics for the berth-allocation problem. *Transportation Science* **39**(4), 526–538.

de Oliveira, R. M., Mauri, G. R. & Lorena, L. A. N. 2012. Clustering search for the berth allocation problem. *Expert Systems with Applications* **39**(5), 5499–5505.

Duman, E., Uysal, M. & Alkaya, A. F. 2012. Migrating birds optimization: a new metaheuristic approach and its performance on quadratic assignment problem. *Information Sciences* **217**, 65–77.

Gutiérrez-Castro, J. P., Melian Batista, B., Moreno-Perez, J. A., Moreno-Vega, J. M. & Ramos-Bonilla, J. 2008. Codea: an architecture for designing nature-inspired cooperative decentralized heuristics. In *'Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)', Vol. 129 of Studies in Computational Intelligence,* Krasnogor, N., Nicosia, G., Pavone, M. & Pelta, D. (eds). Springer Berlin Heidelberg, 189–198.

Imai, A., Nagaiwa, K. & Chan, W. T. 1997. Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation* **31**(1), 75–94.

Imai, A., Nishimura, E. & Papadimitriou, S. 2001. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological* **35**(4), 401–407.

Lalla-Ruiz, E., Melián-Batista, B. & Moreno-Vega, J. M. 2012. Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence* **25**(6), 1132–1141.

Monaco, M. F. & Sammarra, M. 2007. The berth allocation problem: a strong formulation solved by a lagrangean approach. *Transportation Science* **41**(2), 265–280.

Ting, C.-J., Wu, K.-C. & Chou, H. 2014. Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications* **41**(4), 1543–1550.

Umang, N., Bierlaire, M. & Vacca, I. 2013. 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.

Yeo, H. J. 2010. Competitiveness of Asian container terminals. *The Asian Journal of Shipping and Logistics* **26**(2), 225–246.