

Chapter 13

Estimation of Distribution Algorithm for the Quay Crane Scheduling Problem

Christopher Expósito Izquierdo, José Luis González Velarde,
Belén Melián Batista, and J. Marcos Moreno-Vega

Abstract. Estimation of Distribution Algorithms (EDA) are a type of optimization techniques that belong to evolutionary computation. Its operation is based on the use of a probabilistic model, which tries to reach promising regions through statistical information concerning to the individuals that belong to the population. In this work, several solution approaches based on the EDA field are presented in order to solve the Quay Crane Scheduling Problem (QCSP). QCSP consists of obtaining a schedule that minimizes the service time of a container vessel given a set of tasks (loading and unloading operations to/from) by means of the available quay cranes at a container terminal. The experimental results confirm that such algorithms are suitable for solving the QCSP and perform a wide exploration of the solution space using reduced computational times.

Christopher Expósito Izquierdo
Dpto. de Estadística, IO y Computación, ETS de Ingeniería Informática,
Universidad de La Laguna, Spain
e-mail: cexposit@ull.es

José Luis González Velarde
Centro de Manufactura y Calidad, Tecnológico de Monterrey, México
e-mail: gonzalez.velarde@itesm.mx

Belén Melián Batista
Dpto. de Estadística, IO y Computación, ETS de Ingeniería Informática,
Universidad de La Laguna, Spain
e-mail: mbmelian@ull.es

J. Marcos Moreno-Vega
Dpto. de Estadística, IO y Computación, ETS de Ingeniería Informática,
Universidad de La Laguna, Spain
e-mail: jmmoreno@ull.es

13.1 Introduction

The current state of globalization would have been impossible without the integration of containers in global supply chains. Occurrence of containerization has promoted the homogenization of facilities and means of transport involved within an intermodal freight transportation system. Containers allow end-to-end transport without the direct access to the freight, ensuring their integrity and speeding up their handling at intermediate nodes of the transportation system. One of the highlighted advantages of a container is its ability to transport goods of several types. There are containers with standard features for the transportation of hazardous materials, liquids, perishable goods, etc.

Transportation by ship is the predominant way of exchange of freight between different places around the world. Maritime transport is attractive due to its large cargo capacity, which makes the cost per unit smaller than in other means of transport of goods, and its suitable level of service. This hegemony of shipping has led to the development of container vessels with increasing capacities (in excess of 18.000 containers) to service the main sources of demand for goods. At the same time, it has required the creation of major container terminals in seaports capable of handling the large number of containers exchanged worldwide and provide an effective service to shipping companies.

Maritime container terminals are infrastructures to ease an effective transfer of containers within an intermodal network, usually composed by sea transport (container vessels) and hinterland transport (trucks and trains). Typically, a container terminal is divided into several sections: seaside, yard and landside, where major container flows are interrelated. The large number of processes, heterogeneity of the means of transport and the dynamic nature of the decisions to take at a container terminal make it particularly complex to manage. The main objective of container terminals is to perform the process of loading and unloading the proper containers to/from vessels; that is, unload the import containers and load the export containers.

In general terms, after the arrival of a container vessel at a port, it is necessary to establish its location within the facility; that is, providing a berth according to their technical characteristics (length, beam, draft, etc.) and its cargo. In order to carry out the container loading and unloading processes a sufficient set of the available quay cranes are assigned. Loading and unloading of containers should be performed in accordance with the intrinsic restrictions marked by the location of the containers along the vessel and the possible interference among the quay cranes. Unloaded containers are stored temporarily on the container yard until their subsequent collection by other container vessel or some hinterland transport.

Current container terminals cope with a very hard competition with nearby ports. High remote seaports can compete for the same customers because of the expansion of container trade. This fact has led to maritime terminals to bet decidedly on innovation, automation and the improvement of their infrastructures. In order to increase the productivity of a container terminal a careful analysis about the operational activities must be carried out. The improvement process leads to container terminals to be more attractive to shipping companies and allows to increase their level of

competitiveness, based on the operating cost reduction and the offered quality of service enhancing. Optimization methods take a prominent role in this regard due to the fact that their use encourages better use of available resources.

Shipping companies base their business on the economies of scale, so that they pursue to increase the commercial activities in order to maximize profits. One of the main problems faced by shipping companies is that their profits are heavily influenced by the overall cargo transit time of their fleets in port. Most of the time spent by a container vessel in a container terminal is used to perform the loading and unloading tasks, so that its minimization is particularly relevant for the industry. Quay Crane Scheduling Problem pursues to achieve a proper scheduling of the allocated quay cranes for loading and unloading of containers to/from a vessel in the shortest possible service time.

The main aim of this work is to define some basic schemes of Estimation of Distribution Algorithm for the resolution of the Quay Crane Scheduling Problem. The rest of the paper is organized as follows. Section 13.2 introduces and defines the constraints considered in the Quay Crane Scheduling Problem. Section 13.3 describes the approaches proposed to solve the Quay Crane Scheduling Problem. Section 13.4 illustrates the computational experiments carried out in this paper. Finally, Section 13.5 presents the concluding remarks extracted from the work.

13.2 Quay Crane Scheduling Problem

Each one of the berthed container vessels at a container terminal must have a proper stowage plan that allows to effectively carry out the loading and unloading of containers. A stowage plan is composed by several cross-sectional layouts of the vessel bays. The stowage plan aims to establish the specific location of each container onto the vessel (these can be placed in the hold or on the deck), usually according to their weights, types or destination ports. Thus, containers with the same characteristics are included in the same container group. Usually, containers belonging to the same container group are stacked into adjacent slots (stack and tier) within a vessel bay.

Quay cranes of a container terminal have the objective to perform the loading and unloading tasks defined by the corresponding stowage plan. In this work, a *task* refers to the loading or unloading of a set of containers belonging to the same group. In the literature this problem is referred to as Quay Crane Scheduling Problem (QCSP), whose aim is to carry out the tasks associated with a container vessel using the shortest possible service time (makespan).

There are several outstanding works that address the QCSP. In the first place, Kim and Park [2] introduced its definition and constraints. They proposed a Branch & Bound method and a GRASP technique for its resolution. Moccia et al. [4] formulated the QCSP as a vehicle routing problem with side constraints, including precedence relationships between vertices. They also proposed a Branch & Cut method to solve large instances. Samarra et al. [5] decomposed the QCSP into a routing problem solved by a Tabu Search and a scheduling problem solved by a Local Search. Finally, Bierwirth and Meisel [1] demonstrated the incorrectness of the

previous mathematical models and presented new interference constraints. In addition, they developed the Unidirectional Scheduling (UDS) heuristic in order to carry out an exhaustive search of unidirectional schedules. UDS employs a tree search to establish the performed set of tasks by every quay crane and then sets the order of their realization to meet the unidirectional movement. UDS is able to find optimal unidirectional schedulings for small instances using short computational times, its performance is decremented when the problem size grows.

In the QCSP, a set of tasks $\Omega = \{1, \dots, n\}$ and a set of quay cranes $Q = \{1, \dots, m\}$ with similar technical characteristics are considered. For convenience, dummy tasks 0 and $T = n + 1$ are incorporated to represent the starting and the finishing of the service of the container vessel. Let us define $\overline{\Omega} = \Omega \cup \{0, T\}$. Every task has a processing time, p_i , indicating the necessary time to load or unload its containers by means of one available quay crane. The processing time of the dummy tasks are $p_0 = p_T = 0$. Let $l_i \in \mathbb{Z}^+$, the position of the task $i \in \Omega$, expressed as a bay position of the container vessel.

Every quay crane q has a ready time, r^q , that specifies its earliest possible activity. The initial and final bay positions of the quay crane q are denoted as $l_0^q, l_T^q \in \mathbb{Z}^+$, respectively. The required time for a quay crane to move itself between two adjacent bays is \hat{t} . The required time for a quay crane to move itself between the container bays i and j is $t_{ij} = \hat{t}|l_i - l_j|$ (so, $t_{0j}^q = \hat{t}|l_0^q - l_j|$ and $t_{iT}^q = \hat{t}|l_i - l_T^q|$ is the time required by the quay crane q to move from its initial position to the bay j and from the bay i to its final position). Quay cranes can be moved along the length of the container vessel by means of a pair of rails, so that they cannot cross each other and must keep a safety distance δ (measured in container bay units).

There are tasks that need to be done before other ones because they are placed within the same container bay. For example, unloading tasks on the deck must be carried out before unloading tasks in the hold. Let Φ be the set of task pairs within the same container bay for which exists a precedence relationship. On the other hand, Ψ is the set of task pairs that cannot be processed simultaneously. That is,

$$\begin{aligned}\Phi &= \{(i, j) : i \text{ has to be completed before the starting of } j\} \\ \Psi &= \{(i, j) : i \text{ and } j \text{ cannot be done simultaneously}\}\end{aligned}$$

Note that $\Phi \subseteq \Psi$.

The objective of the QCSP is to determine the completion times c_i of all tasks $i \in \overline{\Omega}$ so that the completion time of the last task T (c_T) is minimized; that is, minimizing the makespan.

Figure 13.1 shows an instance of the QCSP with $n = 8$ tasks, $m = 2$ quay cranes, $l_0^1 = 2, l_0^2 = 5, r^1 = r^2 = 0, \hat{t} = 1$ and $\delta = 1$. Processing times and the location of the tasks on the bays are shown in Table 13.1. Additionally,

$$\begin{aligned}\Phi &= \{(1, 2), (5, 6), (5, 7), (6, 7)\} \\ \Psi &= \{(1, 2), (3, 4), (5, 6), (5, 7), (6, 7)\}\end{aligned}$$

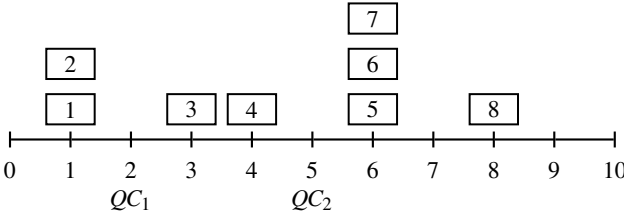


Fig. 13.1 Example of an instance of the Quay Crane Scheduling Problem

Table 13.1 Input data of the example instance

Task i	1	2	3	4	5	6	7	8
Bay position, l_i	1	1	3	4	6	6	6	8
Processing time, p_i	10	8	10	15	7	6	5	10

In this work, the search space is limited to unidirectional schedules. A scheduling is denominated *unidirectional* if the quay cranes have the same sense of moving and this is not changed after the initial positioning (see [1]). The best unidirectional scheduling may be not the optimal scheduling but, usually, it has high quality. On the other hand, without loss of generality, a lexicographic order of the tasks is assumed. That is, tasks are ordered according to their bay position along the container vessel. Figure 13.2 depicts a unidirectional scheduling for the previous instance, where tasks 1, 2, 3, 5, 6 and 7 are done by the quay crane 1 and tasks 4 and 8 are done by the quay crane 2. After the initial positioning of the quay crane 1 on the bay 1 and the quay crane 2 on the bay 4, both quay cranes move unidirectionally from left to right. Note that tasks within the container vessel are sequenced according to the lexicographic order. The lexicographic order indicates that tasks are ordered from left to right and, within each bay, in order of precedence.

The set of feasible solutions for the QCSP is composed by schedules σ that satisfy the precedence and non-simultaneity relations among tasks and safety and interference restrictions among quay cranes. In this sense, the objective is to determine the feasible schedule σ^* , which minimizes the completion time of the last task. The evaluation of every schedule is performed following the scheme based on the disjunctive graph model proposed in [1] so that $f(\sigma)$ denotes the makespan of the schedule σ .

13.3 Estimation of Distribution Algorithm

Estimation of Distribution Algorithms (EDA) [3] are a type of optimization techniques that belong to evolutionary computation. Its operation is based on the use of a probabilistic model that generates new solutions for the population. The probabilistic model is updated through statistical information concerning to the individuals that belong to the population and with the intention of reaching the most promising

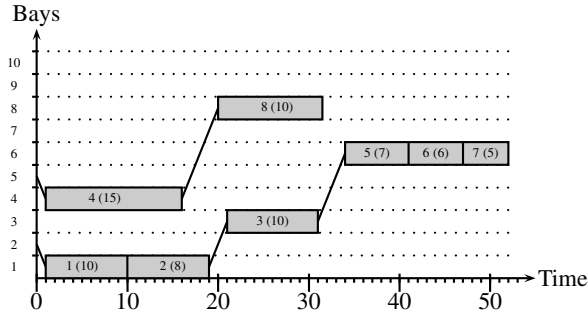


Fig. 13.2 An unidirectional scheduling

Algorithm 13.1. Template of the EDA structure

```

Initialize model
 $p \leftarrow$  Generate initial population
Update probability model
while Stopping criteria is not meet do
     $p' \leftarrow$  Create new population
    Copy best individuals from  $p$  to  $p'$ 
    Fill population  $p'$ 
    Update probability model
     $p \leftarrow p'$ 
end while

```

regions of the solution space. Unlike other classical evolutionary solution schemes, EDA dismiss the use of mutation or recombination operators by the sampling of the probabilistic model.

Broadly speaking, the EDA scheme used in this work for the solution of the QCSP starts with the initialization of the probabilistic model; that is, for each pair of quay crane-task, an initial probability is assigned. From the initial probabilistic model, a population with a defined number of individuals is generated. The population size, N , is considered as fixed during the life cycle of the algorithm. For each generation of the process a new population is created. This new population consists of a defined percentage, α , of the individuals belonging to the previous population and the rest is completed with random-generated solutions from the probabilistic model. The set of solutions defined by the parameter α is denoted as *top solutions* because they are those with the best objective function value in the population. After the generation of each population the probabilistic model update is performed. These steps are repeated while a stopping criteria is not met. The whole pseudocode of the EDA is sketched in Algorithm 13.1.

Since the solution space is limited to unidirectional schedules, the EDA is applied twice, once to find schedules where the sense of movement of the quay cranes is

from left to right along the length of container vessel and another in the reverse sense. The solution provided by the algorithm is the best found in both searches.

In the following several, solution proposals based on the previous scheme are analyzed.

13.3.1 Probabilistic Model

In order to identify promising regions in the search space, EDA keeps a probabilistic model. In this case, at each generation $g \in G$ (G is the maximum number of generations) a probability matrix $p(g)$ with $|Q|$ rows and $|\Omega|$ columns is used, where each value $p_{qt}(g)$ represents the probability of assigning the task t to the quay crane q . That is,

$$p(g) = \begin{pmatrix} p_{11}(g) & p_{12}(g) & \cdots & p_{1\Omega}(g) \\ p_{21}(g) & p_{22}(g) & \cdots & p_{2\Omega}(g) \\ \vdots & \vdots & \ddots & \vdots \\ p_{Q1}(g) & p_{Q2}(g) & \cdots & p_{Q\Omega}(g) \end{pmatrix}$$

The following constraint must be satisfied during the execution of the algorithm:

$$\sum_{t \in \Omega} p_{qt}(g) = 1, \forall q \in Q, \forall g \in G \quad (13.1)$$

The model initialization is performed setting probabilities to each possible task-quay crane assignment. In this sense, there are many options that can be considered. A first approach is to establish similar probabilities to each value, $p_{qt}(0)$. The specific characteristics of the problem suggest that the most promising solutions are those in which tasks are performed by a quay crane that is near to their bay positions along the container vessel in order to reduce its moving distance. In this case, the probabilistic model initialization aims to reflect this fact. Specifically, the initial probability of each $p_{qt}(0)$ is set to the value of the gaussian function with $\mu = l_0^q$, $\sigma^2 = b_{max} - b_{min}$ (where b_{max} and b_{min} are the right-most and the left-most bay position with at least a task, respectively). This is

$$p_{qt}(0) = \frac{1}{\sqrt{b_{max} - b_{min}} \sqrt{2\pi}} \exp \left\{ \frac{-(x - l_0^q)^2}{2(b_{max} - b_{min})} \right\}$$

with

$$x = |l_0^q - l_t|$$

Initial values of the probabilities model need to be normalized to satisfy (13.1).

13.3.2 EDA Based on Counting

One possible assumption to explore the solution space in an intensive way is that, if a task has been previously assigned to a specific quay crane in a large number

Table 13.2 Characteristics of the benchmark instances

	Categories								
	A	B	C	D	E	F	G	H	I
n	10	15	20	25	30	35	40	45	50
m	2	2	3	3	4	4	5	5	6

of high-quality solutions from the population, this assignment should take place in new generated solutions with a high probability.

At each generation $g \in G$ this scheme keeps a matrix $c(g)$ with $|Q|$ rows and $|\Omega|$ columns. Each value $c_{qt}(g)$ is the number of times that task t has been assigned to the quay crane q in one of the *top solutions* from the population. From the values of this matrix the probabilities of the model for the next generation are calculated.

13.3.3 EDA Based on Fitness

Unlike the previous scheme, the quality of the obtained solutions can take a more prominent role in the updating of the probabilistic model. In this case, changes in the probabilities associated to possible assignments are determined by the value of the objective function of the solutions where they are present.

At each generation $g \in G$ this scheme maintains a matrix $f(g)$ with $|Q|$ rows and $|\Omega|$ columns. Each value $f_{qt}(g)$ corresponds with the sum of the relative objective function values of the *top solutions* from the population in which the task t is assigned to the quay crane q . That is, for each solution σ in *top solutions* the value of the proper indexes in $f_{qt}(g)$ are incremented in an amount of $f(\sigma)^{-1}$ with the intention of encouraging the assignment of tasks to quay cranes that have led to solutions with better objective function value.

13.4 Numerical Results

The aim of this section is to compare the performance of the different schemes proposed in the work with respect to the results of the UDS presented in the previous study [1] to illustrate its adequate use. The set of benchmark problems proposed in [2] is considered. This set is composed by 9 groups with 9 instances each one. The characteristics of the instances are shown in Table 13.2. Additionally, $r^q = 0, \forall q$, $\delta = 1$ and $\hat{t} = 1$ is assumed.

All experiments presented in this section with the EDA approaches have been programmed using the Java language and the executions have been carried out on a PC equipped with an Intel Core 2 Duo E8500 3.16 GHz. On the other hand, UDS was executed on a PC P4 2.8 GHz.

For each instance, the EDA approaches have been executed considering a population where $N = 100$ and $\alpha = 20\%$. Execution has been completed when there have been 50 generations without improvement in the objective function value of the best

Table 13.3 Comparison between EDA approaches and UDS. Small instances

Set	instance	UDS		EDA Counting			EDA Fitness		
		obj.	time	obj.	time	dev.	obj.	time	dev.
A	k13	453		453	4,95E-04	0,00	453	6,48E-04	0,00
	k14	546		546	3,42E-04	0,00	546	3,18E-04	0,00
	k15	513		513	2,43E-04	0,00	513	2,58E-04	0,00
	k16	312		312	3,07E-04	0,00	312	2,93E-04	0,00
	k17	453		453	3,17E-04	0,00	453	2,53E-04	0,00
	k18	375		375	2,45E-04	0,00	375	2,20E-04	0,00
	k19	543		543	2,83E-04	0,00	543	1,88E-04	0,00
	k20	399		399	3,12E-04	0,00	399	2,33E-04	0,00
	k21	465		465	3,15E-04	0,00	465	2,47E-04	0,00
	k22	540		540	4,45E-04	0,00	540	2,63E-04	0,00
				1,12E-05	3,30E-04	0,00	2,92E-04	0,00	
B	k23	576		576	6,50E-04	0,00	585	5,82E-04	1,56
	k24	666		666	7,62E-04	0,00	714	5,53E-04	7,21
	k25	738		747	8,60E-04	1,22	753	6,72E-04	2,03
	k26	639		639	8,22E-04	0,00	684	5,20E-04	7,04
	k27	657		657	6,45E-04	0,00	657	4,57E-04	0,00
	k28	531		531	1,09E-02	0,00	531	4,82E-04	0,00
	k29	807		807	7,62E-04	0,00	807	4,58E-04	0,00
	k30	891		891	4,80E-04	0,00	984	4,52E-04	10,44
	k31	570		570	1,09E-03	0,00	570	6,00E-04	0,00
	k32	591		591	2,26E-03	0,00	597	6,35E-04	1,02
				3,68E-05	1,92E-03	0,12	5,41E-04	2,93	
C	k33	603		603	0,03	0,00	603	0,03	0,00
	k34	717		717	0,02	0,00	717	0,03	0,00
	k35	684		684	0,03	0,00	684	0,03	0,00
	k36	678		678	0,03	0,00	678	0,03	0,00
	k37	510		510	0,02	0,00	510	0,02	0,00
	k38	618		618	0,02	0,00	618	0,02	0,00
	k39	513		513	0,03	0,00	519	0,03	1,17
	k40	564		564	0,02	0,00	567	0,02	0,53
	k41	588		588	0,02	0,00	588	0,02	0,00
	k42	573		573	0,03	0,00	573	0,03	0,00
				6,26E-04	0,03	0,00	0,03	0,17	
D	k43	876		876	0,04	0,00	876	0,06	0,00
	k44	822		822	0,04	0,00	822	0,05	0,00
	k45	834		834	0,04	0,00	840	0,04	0,72
	k46	690		690	0,05	0,00	690	0,05	0,00
	k47	792		792	0,04	0,00	792	0,04	0,00
	k48	639		639	0,04	0,00	639	0,04	0,00
	k49	894		894	0,05	0,00	900	0,05	0,67
				3,43E-03	0,04	0,00	0,04	0,20	

solution found during the search or when, for a single generation, $2 \cdot N$ random-generated solutions from the probabilistic model have been previously found in the population. These values have been extracted from the experiment carried out during the study.

The comparative analysis between the proposed EDA schemes and the UDS for small and large instances is summarized in Table 13.3 and Table 13.4, respectively. The first two columns indicate the category and the instance considered. Column *UDS* shows the objective function value and the computational time obtained by means of the UDS. UDS was run with a time limit of 1 hour. In Table 13.4 the computational time is not presented in those instances for which this time was insufficient. Column *EDA Counting* represents the objective function value, the computational time and the deviation between the UDS and the EDA based on counting. Similarly, column *EDA Fitness* shows the objective function value, the

Table 13.4 Comparison between EDA approaches and UDS. Large instances

Set	instance	UDS		EDA Counting			EDA Fitness		
		obj.	time	obj.	time	dev.	obj.	time	dev.
D	k50	741	< 0,01	744	0,05	0,40	759	0,04	2,43
	k51	798	< 0,01	798	0,04	0,00	798	0,03	0,00
	k52	960	< 0,01	960	0,04	0,00	969	0,04	0,94
			< 0,01		0,04	0,13		0,04	1,12
E	k53	717	-	717	0,07	0,00	717	0,06	0,00
	k54	774	0,02	774	0,06	0,00	774	0,05	0,00
	k55	684	0,01	690	0,05	0,88	684	0,05	0,00
	k56	690	0,22	693	0,08	0,43	696	0,07	0,87
	k57	705	0,24	708	0,05	0,43	708	0,04	0,43
	k58	786	0,17	789	0,07	0,38	789	0,06	0,38
	k59	687	0,01	687	0,08	0,00	687	0,06	0,00
	k60	783	0,19	789	0,07	0,77	783	0,06	0,00
	k61	639	0,04	639	0,06	0,00	639	0,05	0,00
	k62	837	0,01	855	0,07	2,15	855	0,06	2,15
			0,10		0,07	0,50		0,06	0,38
F	k63	948	1,51	954	0,11	0,63	948	0,08	0,00
	k64	741	1,06	744	0,12	0,40	780	0,08	5,26
	k65	837	1,61	843	0,11	0,72	861	0,09	2,87
	k66	924	0,63	930	0,10	0,65	933	0,10	0,97
	k67	882	0,24	885	0,08	0,34	882	0,07	0,00
	k68	963	0,03	972	0,10	0,93	963	0,09	0,00
	k69	807	1,40	807	0,10	0,00	807	0,08	0,00
	k70	957	0,61	960	0,08	0,31	957	0,08	0,00
	k71	834	3,77	840	0,13	0,72	921	0,08	10,43
	k72	744	0,35	750	0,09	0,81	744	0,07	0,00
			1,08		0,10	0,55		0,08	1,95
G	k73	870	31,71	879	0,12	1,03	879	0,10	1,03
	k74	843	4,71	855	0,14	1,42	858	0,11	1,78
	k75	675	0,37	684	0,13	1,33	675	0,11	0,00
	k76	852	0,90	864	0,10	1,41	855	0,10	0,35
	k77	699	1,27	714	0,12	2,15	708	0,09	1,29
	k78	642	8,96	651	0,16	1,40	654	0,14	1,87
	k79	744	1,52	774	0,13	4,03	759	0,14	2,02
	k80	750	1,28	771	0,11	2,80	759	0,10	1,20
	k81	738	1,28	744	0,12	0,81	744	0,10	0,81
	k82	717	1,03	738	0,12	2,93	717	0,11	0,00
			2,37		0,13	1,93		0,11	1,04
H	k83	948	6,37	957	0,17	0,95	948	0,12	0,00
	k84	897	3,29	909	0,21	1,34	897	0,17	0,00
	k85	972	5,82	990	0,16	1,85	984	0,14	1,23
	k86	816	-	825	0,17	1,10	819	0,14	0,37
	k87	867	-	885	0,20	2,08	885	0,17	2,08
	k88	768	43,73	780	0,15	1,56	771	0,15	0,39
	k89	843	10,96	849	0,17	0,71	843	0,13	0,00
	k90	1053	24,95	1080	0,19	2,56	1068	0,15	1,42
	k91	837	10,74	849	0,19	1,43	849	0,12	1,43
	k92	897	34,61	918	0,15	2,34	912	0,15	1,67
			19,16		0,17	1,59		0,14	0,86
I	k93	816	-	846	0,20	3,68	831	0,20	1,84
	k94	786	-	816	0,24	3,82	834	0,22	6,11
	k95	834	-	867	0,21	3,96	843	0,19	1,08
	k96	819	-	840	0,23	2,56	837	0,21	2,20
	k97	720	-	732	0,25	1,67	738	0,22	2,50
	k98	735	23,79	765	0,22	4,08	747	0,17	1,63
	k99	852	-	867	0,23	1,76	864	0,18	1,41
	k100	900	-	906	0,25	0,67	897	0,22	-0,33
	k101	813	-	831	0,20	2,21	819	0,19	0,74
	k102	903	-	921	0,23	1,99	900	0,21	-0,33
			23,79		0,22	2,64		0,20	1,68

computational time and the deviation between the UDS and the EDA based on fitness. Computational times reported in the tables are measured in minutes while the deviation between an EDA scheme and the UDS is defined as follows

$$dev. = 100 \cdot \frac{(\text{EDA objective} - \text{UDS objective})}{\text{UDS objective}}$$

Average computational times and average deviations are reported for each category of instances (depicted in bold).

As can be drawn from an analysis of the presented results, the application of the proposed algorithms can achieve satisfactory results for the instances considered in this study. In both cases, the average deviation for each set of instances is less than 3% compared to the results achieved with UDS. Despite the difference in the characteristics of the computers used in the study, the computational times used by the EDA approaches are significantly lower than those employed by the UDS, except for very small instances. In this sense, the largest computational time spent at solving an instance has been 0.25 minutes.

Making a comparison between the two proposals presented in this work, it is possible to observe that performance of EDA based on fitness is higher than the performance of EDA based on counting as the size of instances increases. The reason can be found in the fact that the direct consideration of the quality of the solutions in the update of the probabilistic model allows a more proper distinction in the assignments of tasks to more promising quay cranes than those made by simple counting. This has improved a 0.33% the objective function value in two of the largest instances between those considered.

13.5 Concluding Remarks and Future Work

EDA are a type of techniques that allow to achieve satisfactory results when solving the Quay Crane Scheduling Problem using simple exploration components. In this study, two resolution schemes have been proposed based on the counting of tasks-to-quay cranes assignments and information about the quality of the solutions from the population during the updating process of the probabilistic model. At the same time, with the intention of achieving high quality solutions inherent characteristics of the problem are exploited in order to establish a probabilistic model that can generate a priori initial solutions with good quality.

The experiments carried out in this paper show that some encouraging preliminary experimental results have been achieved and can be used as a basis for future works. Future studies should consider including restarting systems of the probabilistic model to avoid situations in which the generation of new solutions become ineffective, intelligent stopping criteria and updates of the probabilistic model with more precise information about the characteristics of the solutions reached in previous generations.

Acknowledgements. This work has been partially funded by the European Regional Development Fund, Spanish Ministry of Science and Technology (projects TIN2009-13363 and TIN2008-06872-C04-01), Canary Government (project PI2007/019) and University of La Laguna.

References

- [1] Bierwirth, C., Meisel, F.: A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling* 12(4), 345–360 (2009)
- [2] Kim, K.H., Park, Y.-M.: A crane scheduling method for port container terminals. *European Journal of Operational Research* 156(3), 752–768 (2004)
- [3] Larriaga, P., Lozano, J.A. (eds.): *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer, Boston (2002)
- [4] Moccia, L., Cordeau, J.-F., Gaudioso, M., Laporte, G.: A branch and cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics* 53(1), 45–59 (2006)
- [5] Sammarra, M., Cordeau, J.-F., Laporte, G., Flavia Monaco, M.: A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling* 10(4-5), 327–336 (2007)