Inteligencia Artificial 17(54) (2014), 35-47
doi: 10.4114/ia.v17i54.1096

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

A Study of Rescheduling Strategies for the Quay Crane Scheduling Problem under Random Disruptions

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

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

Abstract Providing a suitable answer to different types of unforeseen changes in optimization problems is one challenging goal. This paper addresses the Quay Crane Scheduling Problem under random disruptions, whose goal is to determine the sequences of transshipment operations performed by a set of quay cranes in order to load and unload containers onto/from a berthed container vessel. An evolutionary algorithm is used to find an initial solution of the problem with completely deterministic data, whereas several rescheduling strategies are integrated into a dynamism management system aimed at keeping a proper quality level after a random disruption. Computational experiments indicate that using knowledge about previous static problems can largely improve the performance of the implemented schedule.

Keywords: Quay Crane Scheduling Problem, Rescheduling Strategy, Estimation of Distribution Algorithm.

1 Introduction

In recent years, an increasing research effort has been devoted to analyse the progress of proposed solutions for many real-world applications. Indeed, in real environments, one of the most challenging objectives is to provide a suitable answer to unforeseen changes in the problem or the knowledge about it. A change in a problem represents a modification in its constraints, input data, or optimization criteria. This type of problems, that change over time, are termed *dynamic problems* [13]. Some examples of dynamic problems are Vehicle Routing Problems with changing requests [15], Job Shop Scheduling Problems with random job arrivals [1], and Knapsack Problems with changing parameters [24]. After a change in the environment of the problem at hand occurs, the global optimal solution can change accordingly. The most widely extended goal in these scenarios is to track the global optimal solution over the time horizon [28]. However, only those situations for which the problems before and after a change are somehow related are really interesting from the researching point of view. Otherwise, the problem after a change can be considered as an entirely new problem and, therefore, tackled by means of the optimization techniques already proposed for it.

The maritime container terminals are highlighted sources of dynamic problems due to the heterogeneous nature of their logistic issues and the way they are related [5]. A maritime container terminal is a dedicated node to keep a steady flow of freights within an intermodal transportation network. Its main goal is to serve the container vessels arriving to port [26]. Once a container vessel arrives to the port,

ISSN: 1988-3064(on-line)

it is given a suitable position along the quay on the basis of its technical characteristics (length, draft, cargo, etc.) [17]. A subset of the available quay cranes at the terminal is allocated to the container vessel [6]. The assignment of quay cranes to vessels is usually subject to contractual obligations between the shipping company and the terminal managers as well as principles of cost/benefit analysis.

The quay cranes are aimed at performing the transshipment operations associated with each container vessel, that is, unloading the import containers and loading the corresponding export containers according to the stowage plan of the container vessel at hand [27]. The stowage plan determines the position of each container into the vessel through a coordinate system that states its bay, row, and tier (bay-row-tier system). In this regard, the service time of a container vessel is determined by the time used by the allocated quay cranes when loading/unloading containers onto/from it. This problem is known as Quay Crane Scheduling Problem (QCSP) [16]. Unfortunately, scheduling quay cranes is exposed to unforeseen disruptions such as breakdowns, unavailability of transport vehicles, etc. that impair the overall performance of the terminal.

The main contributions of this work are providing and analysing the performance of different rescheduling strategies for the QCSP in a dynamic environment, where different unforeseen disruptions can happen.

The remainder of this paper is structured as follows. Section 2 introduces the QCSP, whereas Section 3 reviews the most outstanding works from the related literature. Section 4 describes an evolutionary algorithm used to find an initial schedule of the QCSP from a deterministic viewpoint. Afterwards, Section 5 proposes several rescheduling strategies in order to find a new solution after an unforeseen disruption in the environment happens. Section 6 describes the computational experiments carried out. Finally, Section 7 presents some conclusions and a few future research topics.

2 Quay Crane Scheduling Problem

The goal of the Quay Crane Scheduling Problem (QCSP) is to determine the working sequences of a set of quay cranes in order to perform the transshipment operations associated with a container vessel berthed at the quay of a maritime container terminal according to some optimization criteria.

The input data for the QCSP consist of a set of tasks, $\Omega = \{1, 2, ..., n\}$, and a set of quay cranes, $Q = \{1, 2, ..., m\}$, with similar technical characteristics (dimensions, speed, etc.). It is assumed that both tasks and quay cranes have correlative natural numbers from left to right along the container vessel and quay, respectively. A task represents the transshipment operations (loading or unloading) of a container set belonging to the same group, that is, containers with the same destination port, weight, dimensions, etc. and located adjacent to each other within the same bay of the container vessel. Each task $t \in \Omega$ requires an individual processing time, p_t, and is located on a specific bay of the container vessel, l_t. For the sake of convenience, two dummy tasks 0 and $T = n + 1$, where $p_0 = p_T = 0$, are introduced to represent the beginning and the end of the container vessel service, respectively. Furthermore, there are precedence relationships between pairs of tasks located within the same bay:

- Unloading tasks have to be performed before loading tasks.
- Unloading tasks on the deck have to be performed before unloading tasks in the hold.
- Loading tasks in the hold have to be performed before loading operations on the deck.

A comprehensive description of precedence relationships among tasks is presented in [16]. These precedences are defined by means of the set Φ, where,

$$\Phi = \{(i, j) \mid i, j \in \Omega \wedge i \text{ has to be completed before the starting of } j\}. \quad (1)$$

The quay cranes perform the loading and unloading operations of containers onto/from the container vessel at hand. Each quay crane $q \in Q$ is available only after its earliest ready time, $r^q \geq 0$, and is initially located on a position, l_0^q, which is expressed as a bay of the container vessel. All quay cranes can move between two adjacent bays with $\hat{t} > 0$ time units. Therefore, the time required by quay crane q to move from the bay of task i toward the bay of task j is $t_{ij}^q = \hat{t} \cdot |l_i - l_j|$, whereas the time required to move from its initial position to the bay of task i is defined as $t_{0i}^q = \hat{t} \cdot |l_0^q - l_i|$. Quay cranes are mounted on a rail-track system that allows them to move along the quay. Consequently, since they cannot cross

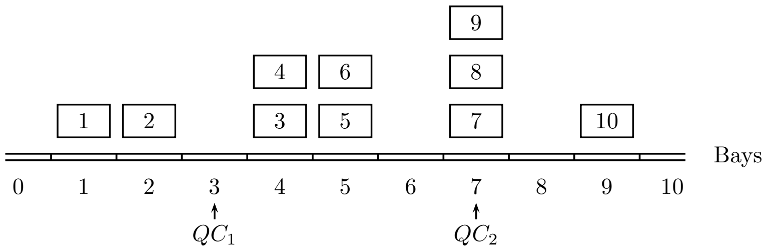

Figure 1: Example of QCSP instance with $n = 10$ tasks and $m = 2$ quay cranes

each other (*i.e.*, non-crossing constraint), the relative positioning over the service time has to be kept. Moreover, the quay cranes have to keep a safety distance, $\delta > 0$, (measured in bay units) between them in order to prevent collisions. Spatial constraints lead to the fact that some tasks cannot be performed at the same time. These constraints are defined by means of the set Ψ, where

$$\Psi = \{(i, j) \mid i, j \in \Omega \wedge i \text{ and } j \text{ cannot be performed at once}\}. \quad (2)$$

The QCSP is already known to be NP-hard [25]. A mathematical formulation for the QCSP can be found in [4].

A solution of the QCSP is a schedule where the starting and the completion time of each task are completely determined. In most of the cases, the main optimization criterion is the minimization of a weighted combination of the makespan, ω, (*i.e.*, the completion time of task T) and the finishing times of the quay cranes, y^q [16]. That is,

$$min\ \alpha_1 \omega + \alpha_2 \sum_{q \in Q} y^q. \quad (3)$$

In practical contexts, terminal managers usually prefer unidirectional schedules (where quay cranes follow the same direction of movement after their initial positioning) to perform the transshipment operations due to the fact that they can ease the stability of the vessel and minimize the potential interferences between quay cranes [4] [18]. Therefore, in this paper only unidirectional schedules are considered. The schedule evaluation is carried out by means of the disjunctive graph scheme proposed in [4].

Figure 1 depicts an example of the QCSP with $n = 10$ tasks and $m = 2$ quay cranes. The location and processing time of each task are presented in Table 1. The quay cranes are initially located on bays $l_0^1 = 3$ and $l_0^2 = 7$, respectively. The safety distance is set to $\delta = 1$ bay between quay cranes and they can move between two adjacent bays in $\hat{t} = 1$ time unit. Additionally,

$$\begin{aligned} \Phi &= \{(3,4), (5,6), (7,8), (8,9), (7,9)\} \\ \Psi &= \{(1,2), (3,5), (3,6), (4,5), (4,6)\} \cup \Phi. \end{aligned}$$

Figure 2 depicts a unidirectional schedule for this example, where quay cranes move from left to right after their initial positioning. In this case, quay crane 1 performs tasks 1, 2, 4, 6, and 9, whereas quay crane 2 performs tasks 3, 5, 7, 8, and 10. If $\alpha_1 = \alpha_2 = 1$ in Equation (3), the values $\omega = 48$, $y^1 = 48$, and $y^2 = 44$ are obtained. Thus, according to Equation (3), the objective function value is 140.

Task i	1	2	3	4	5	6	7	8	9	10
Bay position, l_i	1	2	4	4	5	5	7	7	7	9
Processing, p_i	6	12	8	4	14	7	5	5	7	4

Table 1: Input data for the example depicted in Figure 1

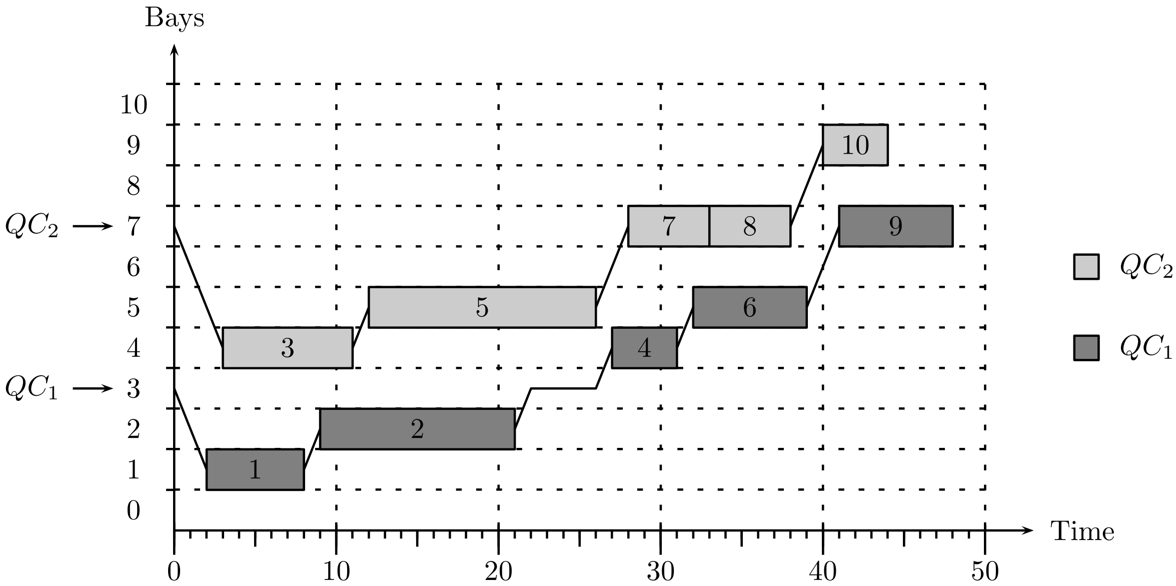

Figure 2: A unidirectional schedule for the example depicted in Figure 1 where quay cranes move from left to right after their initial positioning. The makespan of this schedule is 48 time units, whereas $y^1 = 48$ and $y^2 = 44$ time units

3 Related Works

In the following we summarize several previous works from the literature that address the QCSP with completely deterministic data.

[9] is an early outstanding work in which the first definition of the QCSP is proposed. In this paper, the objective function pursues that the available quay cranes are distributed among the bays in such a way that the working sequences are balanced.

[16] presents a Branch and Bound method with a lower bound in order to solve the QCSP to optimality. However, the computational burden of this optimization technique is very large for medium-size instances so that, the authors also develop a Greedy Randomized Adaptive Search Procedure (GRASP) to overcome this limitation. The quality of the solutions reported by the GRASP does not exceed that of the solutions obtained by the Branch and Bound by more than 10% in the worst case.

[20] develops a new mathematical model for the QCSP based upon the aforementioned formulation proposed in [16] in which several interference scenarios are overcome. Afterwards, this paper discusses the application of a Branch and Cut to real-world environments. The computational results indicate that its performance is better than that of the Branch and Bound analysed in [16].

[25] decomposes the QCSP into a routing problem, which determines the working sequences of the quay cranes, and a scheduling problem, which defines the starting time of each task. A Tabu Search algorithm that allows to find high-quality schedules by means of reasonable computational times is developed.

[4] proposes an optimization method called Unidirectional Scheduling (UDS) based upon the Branch and Cut framework to find the optimal unidirectional schedule of the QCSP. The computational experiments show its superiority over previous proposals.

[21] proposes a Tabu Search based on the ideas discussed in [25] for the QCSP under spatial constraints for the quay crane movements. In this case, the objective function value is to minimize the makespan of the obtained schedule and the total delay of quay cranes.

[8] presents a Genetic Algorithm based upon chromosomes that consider the precedence relationships among tasks. From an initial population generated at random the algorithm applies several crossover and mutation operations in order to find high-promising regions of the search space. The computational experiments indicate the good performance of this approach when solving medium-size instances.

Algorithm 1 Pseudocode of the proposed Estimation of Distribution Algorithm

1: $g \leftarrow 0$
2: $p(g) \leftarrow$ Initialize probabilistic learning model
3: $Pop(g) \leftarrow$ Generate initial population from $p(g)$
4: $\Theta \leftarrow$ Obtain subset of best schedules from $Pop(g)$
5: Update probability learning model $p(g)$ with Θ
6: **while** stopping criteria are not meet **do**
7: $\quad g \leftarrow g + 1$
8: $\quad Pop(g) \leftarrow$ Create new population
9: $\quad \Theta \leftarrow$ Obtain subset of best schedules from $Pop(g - 1)$
10: $\quad$ Copy Θ into $Pop(g)$
11: $\quad$ Fill population $Pop(g)$ with schedules from $p(g)$
12: $\quad$ Update probability learning model $p(g)$ with Θ
13: **end while**

[7] discusses the QCSP at indented berths. In this type of berths, the container vessels are served from both sides in such a way that, additional idle times arise from lifting up the arms of quay cranes when they cross each other. The authors consider that the handling time of the tasks depends on the quay crane that performs it. This version of the QCSP is solved by means of a heuristic framework.

[10] presents an Estimation of Distribution Algorithm in order to find near-optimal unidirectional schedules that require very short computational times. As discussed in Section 4, this optimization technique is used in the present paper with the goal of determining the predictive schedule to be conducted by the quay cranes under deterministic data.

[11] develops a hybrid approach based on an Estimation of Distribution Algorithm with Local Search in order to tackle large instances of the QCSP by means of reasonable computational times.

Unfortunately, as far as the authors of this paper know, there are not released works about the QCSP in dynamic environments. Therefore, this one can be seen as a first approach for this type of real-world environments.

4 Predictive Schedule

The initial schedule of a scheduling problem under completely deterministic data is usually termed in the related literature as *predictive schedule* [2]. This schedule is implemented in the real environment until a change on it happens. In this regard and as summarized in Section 3, several optimization techniques have been already proposed in the literature in order to find an appropriate schedule of the QCSP.

Developing a new approach for the QCSP is out of the scope of this work. However, it is necessary to use some optimization technique to generate a suitable predictive schedule of the QCSP on which the proposed rescheduling strategies can be experimented (see Section 5). As indicated in the literature review presented in Section 3, there are already several exact optimization techniques for the QCSP. The most competitive one is the Unidirectional Scheduling (UDS) developed in [4]. In spite of the fact that this technique reports the optimal solutions of the problem, its use has been rejected due to it requires a great computational burden (larger than 1 hour) in some cases, which makes it unsuitable for dynamic environments. As an alternative, the Estimation of Distribution Algorithm (EDA) proposed in [10] is used in this work in order to obtain the predictive schedule due to the fact that it presents a competitive performance on real-world scenarios and follows evolutionary principles, so that it is particularly appropriate for dynamic environments [13].

In order to provide a self-contained paper, in the following a description of the EDA used in this work is presented. The pseudocode of the proposed EDA is depicted in Algorithm 1. This optimization technique keeps a probabilistic learning model during the search with the aim of sampling the search space and finding high-promising regions on it (line 2). At each generation g of the search, a new population $Pop(g)$ is generated (line 8). The subset of schedules with the lowest objective function values (elitism criterion), Θ, is obtained from the previous population (line 9). This subset is obtained as a percentage β of the previous population, whose value is set by the user. These schedules are included into the new population $Pop(g)$ (line 10), whereas the remaining schedules are sampled randomly from the probabilistic learning model (line 11). It is worth mentioning that the schedules included into the first population are completely sampled from the initial probabilistic learning model (line 3). Finally, the probabilistic learning model is updated through information extracted from the schedules belonging to Θ (lines 5 and 12). These steps are repeated until the optimization criteria are met (line 6).

As described above, the EDA is based upon the use of a probabilistic learning model $p(g)$ (for each generation g) with m rows and n columns, where each element $p_{qt}(g)$ defines the probability of that quay crane q performs task t in a new schedule. That is:

$$p(g) = \begin{pmatrix} p_{11}(g) & p_{12}(g) & \cdots & p_{1n}(g) \\ p_{21}(g) & p_{22}(g) & \cdots & p_{2n}(g) \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1}(g) & p_{m2}(g) & \cdots & p_{mn}(g) \end{pmatrix}$$

The probabilistic learning model is updated with statistical information about promising schedules from the current population in order to represent good features of the search space at hand. Therefore, the sampling of the probabilistic learning model pursues to generate new schedules of high-quality areas of the search space. In this regard, at each generation g, the value $p_{qt}(g)$ is defined as follows:

$$p_{qt}(g) = \frac{c_{qt}(g)}{\sum_{q' \in Q} c_{q't}(g)}, \forall q \in Q, t \in \Omega, \tag{4}$$

where $c_{qt}(g)$ determines the number of times that the task t has been performed by the quay crane q in one high-quality schedule in a previous population.

For a more comprehensive description about this optimization technique and its performance in real-world scenarios, the interested reader is referred to [10].

5 Rescheduling Strategies

The QCSP is a practical application where, in real environments, multitude of disruptions can occur. Some of them are breakdowns of quay cranes, unavailability of transport vehicles, opening of hatches, etc. Unforeseen disruptions can cause that the suitability of the predictive schedule is severely impaired and, therefore, this must be revised according to the new information in order to keep a high performance [3].

The rescheduling strategies are aimed at adjusting the part of the predictive schedule not already implemented in such a way that the disruptions are assimilated. These strategies can be grouped into partial and complete, according to the size of the part of the predictive schedule to be restructured [19]. Furthermore, information about the search space or historical data can provide a useful support for rescheduling strategies. The interested reader is referred to [13] and [2] for obtaining in-depth studies concerning these environments.

A dynamism management system based upon [22] is considered to properly handle the possible disruptions in the environment. Its main goals are the detection of new disruptions, finding suitable schedules and implementing the working sequences of the quay cranes allocated to the container vessel. In this work, a disruption is denoted as $\gamma(q, t, r)$, which defines that the quay crane q at time t is disrupted for r time units. Therefore, a complete knowledge about the disruption at the time it happens is available [14]. The architecture of this system is depicted in Figure 3.

The Tracking System allows to follow the implementation of the proposed schedule over time. That is, it checks the performance of the transshipment operations initially defined as the predictive schedule (Subsection 4) at each time period. Whenever a disruption in the environment is detected, the Event Manager creates a new static problem whose set of tasks is composed of those that are not previously performed, whereas the quay cranes are initially located on the bays in which they are at the time the disruption happens. The new problem is solved by means of a specific solver for it. It is worth mentioning that, the dynamism management system proposed in this work is totally independent of the optimization technique aimed at addressing the QCSP. In this regard, different solution techniques can be effectively used in future research. Thus, each rescheduling strategy proposed along this work follows

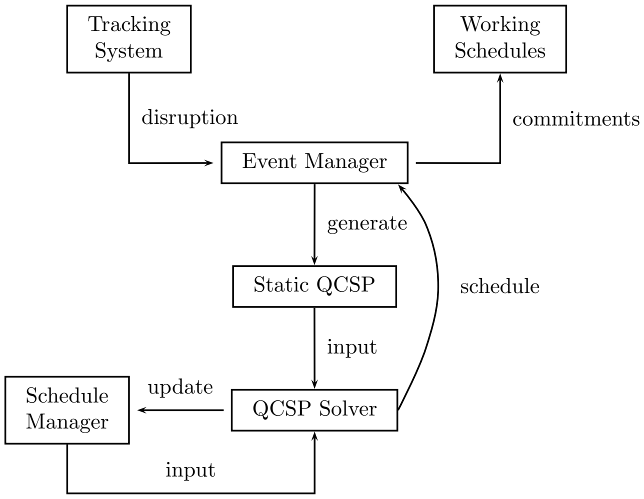

Figure 3: Architecture of the dynamism management system aimed at handling the unforeseen changes in the environment

an event-driven policy because the predictive schedule is always revised after a new disruption happens in the environment [23]. Solving the static problems is supported by the Schedule Manager, which stores historical and promising information about the static problems to solve. This information is updated every time a new static problem is solved. Finally, the schedules found for the new static problems are committed to the available quay cranes.

In the following subsections an individual description about the six rescheduling strategies (organized into three groups: right-shifting, N_1-based, and EDA-based rescheduling strategies) implemented in QCSP Solver (Figure 3) is presented.

5.1 Right-Shifting

The right-shifting strategy uses the predictive schedule (Section 4) as a basis in order to obtain a new schedule where the unforeseen disruption is accommodated. In this strategy, the working sequences of the quay cranes are kept over the time. However, the tasks affected by the disruption are delayed according to their lengths. Additionally, other quay cranes might be blocked due to the interferences between them.

Figure 4 shows the implemented schedule obtained by means of the right-shifting strategy for the predictive schedule previously depicted in Figure 2 after the disruption $\gamma(2, 20, 10)$. In this case, the quay crane 2 is stopped at time 20 for 10 time units. This fact gives rise to that the task 5 is performed in two working periods, named 5_1 and 5_2, respectively.

5.2 N_1-based Rescheduling Strategies

The N_1-based rescheduling strategy is proposed in [12] for the Job Shop Scheduling Problem in order to improve the robustness, that is, the quality of the implemented schedule after an unforeseen change in the environment. This strategy is based on minimizing the implementation costs of a set of schedules located around the predictive schedule. Whenever a new disruption happens, a set of candidate neighbour solutions $N_1(\sigma)$ is generated from the predictive schedule σ for the new static problem. The fittest neighbour solution from $N_1(\sigma)$ is selected as new predictive schedule.

Several neighbourhood structures can be defined in order to generate the set $N_1(\sigma)$. In this work, two different neighbourhood structures are considered: reassignment and interchange of tasks. The former does the reassignment of a task from a quay crane to another adjacent one, whereas the second does the interchange of two tasks performed by adjacent quay cranes. In a natural way, the definition of these neighbourhood structures results in two rescheduling strategies. Moreover, a combined rescheduling strategy, where the explored neighbourhood is made up by the set of neighbours obtained with the

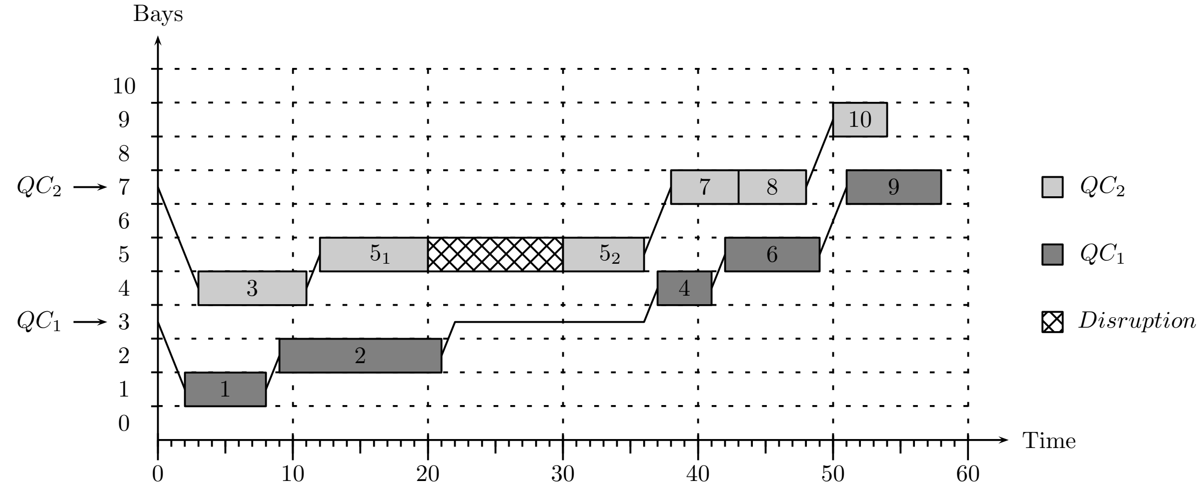

Figure 4: Schedule obtained after disruption $\gamma(2, 20, 10)$ in the predictive schedule depicted in Figure 2

reassignment and interchange structures, is proposed.

5.3 EDA-based Rescheduling Strategies

Undoubtedly, tracking the global optimal solution over time in a dynamic environment is certainly a hard issue. In this regard, exploiting knowledge collected from previous searches can provide a high performance when solving the new static problem after a change in the environment occurs [13]. In real applications such as the QCSP, changes are highly unpredictable in such a way that one of the most important goals for using available knowledge is to provide a high diversity level for the rescheduling strategies in order to find the new optimal solution in the subsequent search space.

The executions of the EDA for different static problems provide useful knowledge to face unforeseen changes in the future. Indeed, the population can be used as a representative set of promising solutions from the search space as long as the changes in the environment are smooth. The Schedule Manager presented in Figure 3 records the population generated in the last execution of the EDA in order to provide it to the rescheduling strategies. However, since a disruption in the QCSP produces that the dimensions of the new problem are different to those of the previous one, the individuals in the population have to be adapted in a careful way. In this case, for each individual, those tasks already performed are removed. Consequently, the adapted population (Pop') can be used as the initial population in another execution of the EDA for solving the new static problem.

On the other hand, the initialization of the probabilistic learning model can be carried out according to information provided by the Schedule Manager. Let Ω' define the set of tasks in the new static problem and $\Omega'_s \subseteq \Omega'$ be a subset of selected tasks. After a change, the probabilities associated with Ω'_s in the new probabilistic learning model are initialized from statistical information of the last adapted population Pop'. That is, each probability value $p_{qt}(0)$ is initialized as follows

$$p_{qt}(0) = \begin{cases} \frac{\sum_{\sigma' \in Pop'} c_{qt}^{\sigma'}}{N} & \text{if } t \in \Omega'_s \\ c_{qt}^{\sigma} & \text{otherwise} \end{cases}, \qquad (5)$$

where σ is the predictive schedule, N is the number of individuals in Pop' and c_{qt}^{σ} is a binary variable set to 1 if, and only if, quay crane q performs task t in schedule σ. This process is based on the larger the number of times a task is performed by a specific quay crane in schedules from Pop', the larger the corresponding probability value. Additionally, only tasks included into Ω'_s are considered in the search, whereas the remaining ones are initialized according to the assignments in the predictive schedule.

Two rescheduling strategies are proposed from the developed scheme. The former (Reduced EDA-based rescheduling strategy) considers only those tasks directly affected by the disruption, whereas the

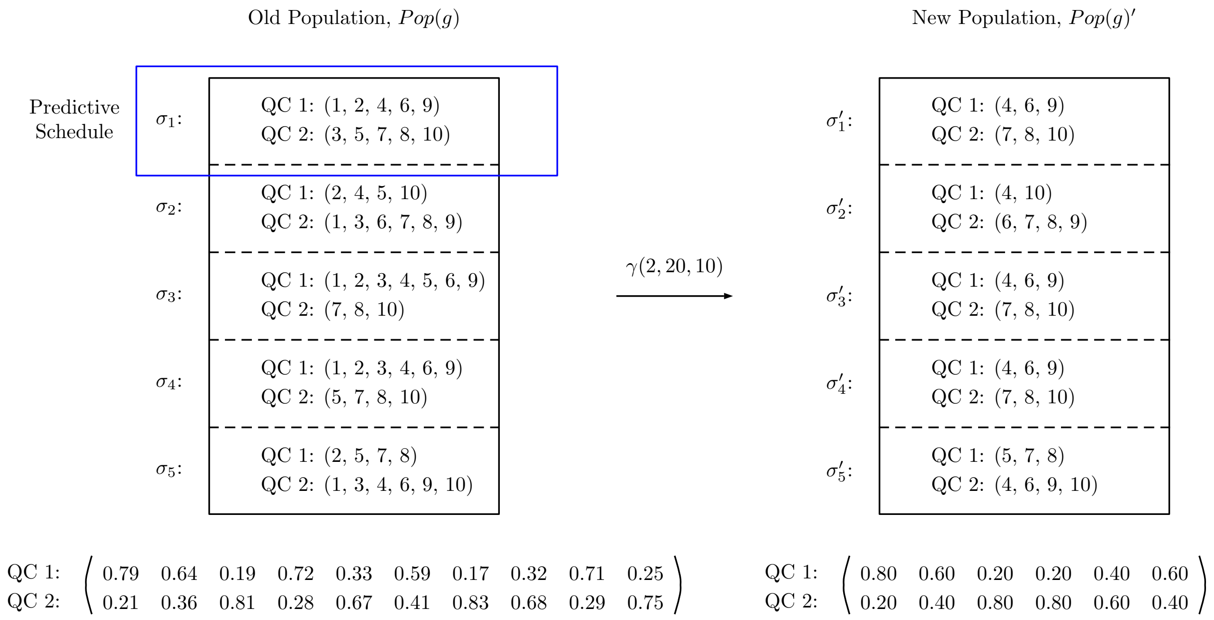

Figure 5: Behaviour of the complete EDA-based rescheduling strategy when the disruption $\gamma(2, 20, 10)$ occurs in the predictive schedule depicted in Figure 2

second one (Complete EDA-based rescheduling strategy) considers all tasks of the new static problem ($\Omega'_s = \Omega'$).

In order to ease the understanding of these rescheduling strategies, Figure 5 shows the behaviour of EDA-based rescheduling strategy in the same scenario as that depicted in Subsection 5.1 when $\Omega'_s = \Omega'$ (Complete EDA-based rescheduling strategy). That is, given the predictive schedule depicted in Figure 2, the disruption $\gamma(2, 20, 10)$ occurs. As can be seen, the EDA provided a population with $N = 5$ schedules when it was applied to the QCSP under completely deterministic data. It is worth mentioning that the predictive schedule is included into this population. The probabilistic learning model at that time is also reported in Figure 5. As shown in Figure 4, the tasks 1 and 3 have already been performed by quay cranes 1 and 2, respectively, whereas the tasks 2 and 5 are being performed at the disruption time. Therefore, the tasks 1, 2, 3, and 5 are not considered in the new problem. The schedules in the population are adapted to the new QCSP instance in such a way that the aforementioned tasks are removed. Once the new population is built, the probabilistic learning model can be derived according to Equation (5).

6 Experimental Study

The main goal of this section is to carry out a performance analysis of the rescheduling strategies proposed in Section 5 under dynamic environments, where unforeseen disruptions can happen.

The suite of benchmark problems proposed in [16] and extended in [4] is used in this paper because it is a consolidated basis for experiments concerning the QCSP. This suite is composed of 90 instances grouped into 9 sets with 10 instances each one. The dimensions of the problem instances are ranged from 10 up to 50 tasks and from 2 up to 6 quay cranes. For each problem instance, all quay cranes are available from the starting of the scheduling horizon, that is, $r^q = 0, \forall q \in Q$. Additionally, the safety distance between quay cranes is set to $\delta = 1$ bay and they move between adjacent bays in $\hat{t} = 1$ time units.

All experiments reported in this section have been carried out by means of a computer equipped with an Intel Core 2 Duo E8500 3.16 GHz and 4 GB of RAM. The programming language chosen has been Java Standard Edition 7.

Several disruption scenarios can be proposed for the QCSP. In this paper, a scenario is composed of a set of random and non-overlapping disruptions. Therefore, there is no available information about the disruptions in advance. Additionally, two quay cranes cannot be disrupted at once. Each scenario is defined according to a group of parameters: the scheduling horizon within which the disruptions can occur at random, τ, the number of quay cranes that can be disrupted during the scheduling horizon, η, and the average length of the disruptions, μ. Without loss of generality, it is assumed that the scheduling horizon starts at time 0. For each set of instances, several sets of disruption scenarios are generated according to the number of available quay cranes. That is, m sets of scenarios are generated for problem instances with m quay cranes, in such a way that, η ranges from 1 up to m, respectively. Furthermore, as η increases, μ is reduced from 300 down to 50. Finally, each set of disruption scenarios is composed of 30 scenarios where $\tau = 350$.

On the other hand, the EDA described in Section 4 has been used in order to find a predictive schedule for the considered problem instances and by several rescheduling strategies in an implicit way (*i.e.*, EDA-based rescheduling strategies, Subsection 5.3). The behaviour of this metaheuristic algorithm is governed by a set of parameters. In this paper, the parameter values used are the same as in the original paper [10]. That is, the population size (N) is set to 100 individuals and the percentage of best schedules to select at each step of the search (β) is set to 20%. Moreover, the optimization method called Unidirectional Scheduling (UDS) and proposed in [4] is considered for comparison reasons. As indicated in Section 3, this method allows to reach the optimal unidirectional schedule of the QCSP.

Table 2 reports the results obtained when solving the QCSP with completely deterministic data as well as under the proposed disruption scenarios. As done in other works from the related literature, values $\alpha_1 = 3$ and $\alpha_2 = 0$ in Equation (3) are considered in the experiments for comparison purposes. The first five columns show the set of instances, the number of tasks, the number of quay cranes, the number of quay cranes to be disrupted, and the average length of the disruptions, respectively. Next, two columns show the average objective function values reached for the deterministic QCSP by means of the UDS and the proposed EDA. In the case of EDA, results correspond to average values of 10 executions. An exhaustive analysis of this method can be found in [10]. Column RS reports the average objective function values reached with the right-shifting rescheduling strategy for the predictive schedule obtained with EDA under the corresponding set of disruption scenarios. This rescheduling strategy is used as reference for comparison. In a similar way, the remaining columns in the table show the average relative deviation in percentage with respect to the results provided by the right-shifting rescheduling strategy and achieved through the N_1-based and EDA-based rescheduling strategies, respectively. Finally, last row reports some average values. In all cases, the computational times required by the rescheduling strategies are quite short, less than 2 seconds.

As can be seen in Table 2, from a general point of view, there are meaningful differences in the performance of the rescheduling strategies. Firstly, N_1-based rescheduling strategies present limited improvements with respect to the right-shifting one. If a comparison between them is carried out, the reassignment of tasks provides better results than the interchange of tasks. The reason is that, after a disruption, the working sequences are unbalanced in such a way that only interchanges of tasks with highly different processing times are profitable. Moreover, the combined N_1-based rescheduling strategy is superior due to it covers the previous ones.

On the other hand, the use of knowledge provided by the Schedule Manager improves the quality of the implemented schedules under different disruption scenarios. An average improvement greater than 4.5% is obtained when only those tasks directly affected by the corresponding disruptions are considered in the initialization step of the probabilistic learning model for the new static problems, whereas this improvement is greater than 7% if all tasks are considered. This fact reveals that determining a suitable subset of tasks for new probabilistic learning models allows to find high-quality schedules. Considering only tasks directly affected can be seen as an intermediate approach, but can be unable when reaching the new global optimal solution requires modifying the probabilities associated to additional tasks.

Furthermore, if an analysis by set of instances is performed, it can be seen that there are situations where EDA-based rescheduling strategies present improvements greater than 20%, whereas only improvements shorter than 13% are reached by N_1-based rescheduling strategies in all cases. The performance of EDA-based rescheduling strategies is superior than N_1-based rescheduling strategies in most cases.

Finally, an important reduction in the performance is obtained as the number of disruptions increases. The reason is that rescheduling strategies try to reassign the tasks not yet implemented by the disrupted quay crane. Therefore, the resulting working sequences in the new schedule tend to be balanced in order to

Set	n	m	η	μ	f_{UDS}	f_{EDA}	RS	Reassignment	Interchange	Combined	Reduced EDA	Complete EDA
A	10	2	1	300	459.9	459.9	1347.72	-5.41	-0.89	-5.00	-8.11	-8.09
			2	150			1003.99	-0.17	-1.09	1.99	4.57	4.82
B	15	2	1	300	666.6	673.5	1564.47	-7.10	-3.67	-9.53	-14.26	-13.74
			2	150			1179.16	4.30	4.01	5.57	5.01	5.68
C	20	3	1	300	604.8	606.3	1498.30	-7.84	-4.48	-9.78	-16.38	-17.86
			2	150			1096.94	2.71	3.19	3.29	6.61	6.12
			3	100			953.12	6.70	7.48	7.66	11.93	12.94
D	25	3	1	300	804.6	807.6	1699.03	-7.67	-5.43	-10.56	-19.57	-22.86
			2	150			1276.35	0.92	1.94	1.43	2.40	0.78
			3	100			1165.21	4.19	3.79	5.00	5.18	4.97
E	30	4	1	300	730.2	737.4	1626.10	-7.74	-4.99	-9.95	-17.30	-22.45
			2	150			1208.48	-2.54	-0.42	-2.39	0.60	-2.64
			3	100			1088.68	2.86	2.34	3.50	5.50	5.41
			4	75			1024.04	4.48	3.49	5.76	5.44	7.46
F	35	4	1	300	863.7	876.6	1766.68	-8.22	-5.93	-11.66	-20.78	-25.44
			2	150			1343.64	-4.78	-1.74	-4.16	-0.46	-5.98
			3	100			1213.89	1.56	1.80	2.94	4.14	2.39
			4	75			1140.20	4.57	3.97	6.27	6.78	6.61
G	40	5	1	300	753.3	771.9	1661.30	-7.44	-6.10	-10.93	-17.50	-23.64
			2	150			1242.56	-3.07	-1.92	-4.49	-1.22	-5.29
			3	100			1135.65	1.00	0.52	-0.29	1.92	0.57
			4	75			1055.67	3.16	1.59	4.25	3.44	3.39
			5	60			1012.31	2.98	3.04	4.19	3.99	4.87
H	45	5	1	300	890.1	915.3	1794.48	-7.91	-6.39	-12.26	-18.91	-26.63
			2	150			1574.85	-6.61	-4.50	-9.28	-11.57	-18.91
			3	100			1460.27	-4.43	-2.65	-6.48	-7.48	-13.63
			4	75			1385.36	-2.74	-1.29	-4.30	-4.88	-10.10
			5	60			1331.51	-1.41	-0.20	-2.65	-3.07	-7.41
I	50	6	1	300	812.7	845.7	1720.69	-7.21	-5.27	-9.73	-16.32	-24.51
			2	150			1295.76	-4.01	-1.54	-4.30	-1.79	-8.26
			3	100			1183.14	-1.37	-0.17	-0.98	3.61	-0.38
			4	75			1113.67	0.55	1.44	0.91	3.17	0.91
			5	60			1068.60	2.14	2.60	2.78	3.98	3.79
			6	50			1047.68	2.16	1.12	2.96	3.59	4.16
					731.6	743.8	1302.34	-2.35	-1.08	-3.14	-4.57	-7.48

Table 2: Comparison of rescheduling strategies under disruption environments

minimize the optimization criterion (Equation (3)). As a subsequent effect of this process, the remaining quay cranes are more susceptible of being affected by other unforeseen disruptions. Consequently, since in this experimentation disruptions are applied to different quay cranes, as the value η approximates m the resulting effect obtained when the right-shifting rescheduling strategy is applied is that some disruptions are not within the corresponding working sequences and, then, are not actually applied.

7 Conclusions and Future Research Topics

The management of the available quay cranes is a backbone issue in container terminals. An evolutionary algorithm is presented to determine a suitable schedule of the quay cranes in a completely deterministic environment. However, their activities are subjected to unforeseen disruptions such as breakdowns which seriously impair their performance.

This paper presents a dynamism management system to follow the implementation of the schedule for a given container vessel. Its architecture allows to integrate different rescheduling strategies in order to face up changes in the environment. These rescheduling strategies provide a suitable answer to disruptions in the working sequences of quay cranes, in such a way that, the original schedule can be modified to accommodate the disruptions and keep a high performance. Several approaches of rescheduling strategies are discussed. Computational experiments show that using knowledge about disruptions can largely improve the behaviour of rescheduling strategies under different disruption scenarios.

From the presented work, several future research topics can be derived. In the first place, studying the predictability of changes in the future in order to determine if these follow a well-known pattern can allow to design more suitable rescheduling strategies. In the same line, predictive schedules less sensitive to new disruptions can be found from historical data concerning the length and occurrence of disruptions in the past. Finally, some real contexts may encourage that rescheduling strategies follow periodic or hybrid policies, where these are not applied whenever a change in the environment happens.

Acknowledgements

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

References

[1] M.A. Adibi, M. Zandieh, and M. Amiri. Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications*, 37(1):282 – 287, 2010.

[2] N. Al-Hinai and T.Y. ElMekkawy. Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics*, 132(2):279 – 291, 2011.

[3] H. Aytug, M.A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161(1):86 – 110, 2005.

[4] C. Bierwirth and F. Meisel. A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling*, 12(4):345–360, 2009.

[5] H.J. Carlo, I.F.A. Vis, and K.J. Roodbergen. Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research*, 236(1):1 – 13, 2014.

[6] S.L. Chao and Y.J. Lin. Evaluating advanced quay cranes in container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 47(4):432 – 445, 2011.

[7] J.H. Chen, D.H. Lee, and J.X. Cao. Heuristics for quay crane scheduling at indented berth. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1005 – 1020, 2011.

[8] S.H. Chung and K.L. Choy. A modified genetic algorithm for quay crane scheduling operations. *Expert Systems with Applications*, 39(4):4213–4221, 2012.

[9] C.F. Daganzo. The crane scheduling problem. *Transportation Research Part B: Methodological*, 23(3):159 – 175, 1989.

[10] C. Expósito-Izquierdo, J.L. González-Velarde, B. Melián-Batista, and J.M. Moreno-Vega. Estimation of distribution algorithm for the quay crane scheduling problem. In D.A. Pelta, N. Krasnogor, D. Dumitrescu, C. Chira, and R. Lung, editors, *Nature Inspired Cooperative Strategies for Optimization (NICSO 2011)*, volume 387 of *Studies in Computational Intelligence*, pages 183–194. Springer Berlin Heidelberg, 2012.

[11] C. Expósito-Izquierdo, J.L. González-Velarde, B. Melián-Batista, and J.M. Moreno-Vega. Hybrid estimation of distribution algorithm for the quay crane scheduling problem. *Applied Soft Computing*, 13(10):4063 – 4076, 2013.

[12] M.T. Jensen. Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures. *Applied Soft Computing*, 1(1):35 – 52, 2001.

[13] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments -a survey. *IEEE Trans. Evolutionary Computation*, 9(3):303–317, 2005.

[14] S.M. Kamrul-Hasan, R. Sarker, and D. Essam. Genetic algorithm for job-shop scheduling with machine unavailability and breakdowns. *International Journal of Production Research*, 49(16):4999–5015, 2011.

[15] M.R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, and E.G. Talbi. A comparative study between dynamic adapted pso and vns for the vehicle routing problem with dynamic requests. *Applied Soft Computing*, 12(4):1426 – 1439, 2012.

[16] K.H. Kim and Y.M. Park. A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156(3):752–768, 2004.

[17] 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.

[18] P. Legato, R. Trunfio, and F. Meisel. Modeling and solving rich quay crane scheduling problems. *Computers and Operations Research*, 39(9):2063 – 2078, 2012.

[19] L. Liu, H.Y. Gu, and Y.G. Xi. Robust and stable scheduling of a single machine with random machine breakdowns. *The International Journal of Advanced Manufacturing Technology*, 31:645–654, 2007.

[20] L. Moccia, J.F. Cordeau, M. Gaudioso, and G. Laporte. A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics*, 53(1):45–59, 2006.

[21] M.F. Monaco and M. Sammarra. Quay crane scheduling with time windows, one-way and spatial constraints. *International Journal of Shipping and Transport Logistics*, 3(4):454–474, 2011.

[22] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, 10(4):327–343, 2005.

[23] A. Pfeiffer, B. Kádár, and L. Monostori. Stability-oriented evaluation of rescheduling strategies, by using simulation. *Computers in Industry*, 58(7):630 – 643, 2007.

[24] P. Rohlfshagen and X. Yao. The dynamic knapsack problem revisited: A new benchmark problem for dynamic combinatorial optimisation. In *Applications of Evolutionary Computing*, volume 5484 of *Lecture Notes in Computer Science*, pages 745–754. Springer Berlin / Heidelberg, 2009.

[25] M. Sammarra, J.F. Cordeau, G. Laporte, and M.F. Monaco. A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling*, 10(4-5):327–336, 2007.

[26] B.W. Wiegmans, P. Rietveld, and P. Nijkamp. Container terminal services and quality. Serie Research Memoranda 0040, VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics, 2001.

[27] I. D. Wilson and P. A. Roach. Container stowage planning: A methodology for generating computerised solutions. *The Journal of the Operational Research Society*, 51(11):1248–1255, 2000.

[28] Y. Wu, Y. Wang, and X. Liu. Multi-population based univariate marginal distribution algorithm for dynamic optimization problems. *Journal of Intelligent and Robotic Systems*, 59:127–144, 2010.