

Simulation optimization based ant colony algorithm for the uncertain quay crane scheduling problem

Naoufal Rouky^{a*}, Mohamed Nezar Abourraja^a, Jaouad Boukachour^a, Dalila Boudebous^a, Ahmed El Hilali Alaoui^b and Fatima El Khoukhi^c

^aNormandie Univ, UNIHAVRE, 76600 Le Havre, France

^bFaculty of Science and Technology, Sidi Mohamed Ben Abdallah University, 2202 Fez, Morocco

^cFaculty of Arts and Humanities, Moulay Ismail University, B.P 11202 Meknes, Morocco

CHRONICLE

Article history:

Received September 20 2017

Received in Revised Format

December 25 2017

Accepted February 14 2018

Available online

February 14 2018

Keywords:

Container terminal

Simulation Optimization

Quay crane

Uncertainty

ABSTRACT

This work is devoted to the study of the Uncertain Quay Crane Scheduling Problem (QCSP), where the loading /unloading times of containers and travel time of quay cranes are considered uncertain. The problem is solved with a Simulation Optimization approach which takes advantage of the great possibilities offered by the simulation to model the real details of the problem and the capacity of the optimization to find solutions with good quality. An Ant Colony Optimization (ACO) meta-heuristic hybridized with a Variable Neighborhood Descent (VND) local search is proposed to determine the assignments of tasks to quay cranes and the sequences of executions of tasks on each crane. Simulation is used inside the optimization algorithm to generate scenarios in agreement with the probabilities of the distributions of the uncertain parameters, thus, we carry out stochastic evaluations of the solutions found by each ant. The proposed optimization algorithm is tested first for the deterministic case on several well-known benchmark instances. Then, in the stochastic case, since no other work studied exactly the same problem with the same assumptions, the Simulation Optimization approach is compared with the deterministic version. The experimental results show that the optimization algorithm is competitive as compared to the existing methods and that the solutions found by the Simulation Optimization approach are more robust than those found by the optimization algorithm.

© 2019 by the authors; licensee Growing Science, Canada

1. Introduction

Maritime transport, by its possibilities of consolidation, plays a crucial role in the international trade, and has become one of the essential actors of globalization. Today, the global volume of freight carried by sea is estimated at over 90%. A growth that is explained by several factors such as the modernization of logistics, the harmonization of equipment and the large capacity of the world fleet of container ships that has exceeded 17.5 million TEUs in 2014 (UNCTAD, 2014). This staggering growth, place the shipping market in the second global position after the agro-food market with a turnover of more than 1.5 trillion Euros (UNCTAD, 2016). This brings new investments but also creates strong competition between container terminals that belong to the same geographical area. One of the key points for improving the competitiveness of a container terminal is the minimization of the time spent by the vessels at the port,

* Corresponding author

E-mail: naoufal.rouky@univ-lehavre.fr, naoufal.rouky@gmail.com (N. Rouky)

denoted by the vessel turnaround time, which depends mainly on the efficiency of planning methods used by the port operators in the handling, storage and transport operations (Tongzon and Heng, 2005). Thus, port operators are on a continuous search of adequate solutions that can address this challenge. Especially for the seaside operations, where arises three major problems: the Quay Crane Assignment Problem (QCAP), the Berth Allocation Problem (BAP) and the Quay Crane Scheduling Problem (QCSP) (Meisel, 2009).

This work is devoted to the study of the Quay Crane Scheduling Problem (QCSP), which consists of determining the best schedule of the unloading/loading operations of containers by quay cranes assigned to a vessel such as the overall handling time is minimized. In the QCSP, vessels are partitioned longitudinally into bay areas, each of them is carrying a certain number of containers. Containers with the same characteristics; namely: weight, origin, destination and type of operation (loading or unloading), are usually located adjacent to each other and they are considered as a single task to facilitate their handling operations (Expósito-Izquierdo et al., 2013). Precedence constraints are defined on tasks in order to respect the stacking plane of the containers, and quay cranes move on the same track therefore, they cannot pass each other and a safe distance must be kept between them to avoid congestion. According to the level of aggregation considered on the definition of a task, we can distinguish in the existing literature between three classes of QCSP: QCSP with container groups where a task involves the handling of a group of containers within a bay, QCSP with complete bays where a task refers to all containers within a bay and the QCSP with bay areas where a task represents all containers within a connected area of bays (See Fig. 4 in Bierwirth & Meisel, 2010 for more details). The QCSP has been the subject of several studies. However, as shown in the next section, most of previous works related to the QCSP study the deterministic version. Even if, in the real-life situations the port operators have to deal with uncertainties in a range of factors such as handling time of tasks and travel time of quay cranes, those perturbations reduce the quality of deterministic solutions and have a significant impact on the overall vessel turnaround time. Therefore, this work is devoted to the study of the uncertain QCSP with groups of containers, in which the time of unloading/loading of containers and the travel time of quay cranes between adjacent bays are supposed uncertain. The problem is solved with a Simulation Optimization approach which takes advantage of the great possibilities offered by the simulation to model the real details of the problem and the capacity of the optimization to find solutions with good quality. An Ant Colony Optimization (ACO) meta-heuristic hybridized with a Variable Neighborhood Descent (VND), used as a local search, is proposed to determine the assignments of the tasks to the quay cranes and the sequences of executions of tasks on each crane. Simulation is used inside the optimization algorithm to generate scenarios in agreement with the probabilities of distributions of the uncertain parameters, thus, we carry out stochastic evaluations of the solutions found by each ant.

The remainder of this paper is structured as follows. The next section presents an overview of existing literature on the QCSP problem. We give more insights about the problem studied in section 3. Then, a Simulation Optimization based Ant Colony approach is proposed to solve the uncertain QCSP in section 4, followed by numerical experiments in section 5. Finally, section 6 summarizes the work.

2. Related works

Container terminal operations are considered as the most challenging topics in the area of operation research, because of their complexity and their applicability. In particular, the QCSP has received great attention during the last decades. Recent compressive overviews of the problem are presented by Bierwirth and Meisel (2010, 2015), Carlo et al. (2015) and Boysen et al. (2017).

QCSP with complete bays was first studied by Daganzo (1989), with the aim of minimizing the total cost of delays. Container ships were assumed to be partitioned into ships-bays, a task was defined as the handling operations of all containers on a bay and only one crane was allowed to work on a bay at a time. Non-interference constraints between quay cranes were not considered and tasks were supposed preemptive. The static and dynamic version of the problem were considered and solved by an exact and a heuristic approach. Peterkofsky and Daganzo (1990), considered the QCSP with the same previous

assumptions and proposed a branch and bound method for its resolution. Later on, Lim et al. (2004) studied a more realistic version of the QCSP with complete bays by considering the non-interference constraints. Three approaches were proposed to determine the best schedule of quay cranes; a dynamic programming algorithm was addressed to solve simpler instances, where a probabilistic Tabu Search and a Squeaky Wheel Optimization heuristic were used to tackle the hardest instances. Zhu and Lim (2006), addressed the QCSP to minimize the latest completion time of all tasks and considered that tasks are non-preemptive. They showed that the problem is NP-complete and provided a branch and bound algorithm and a simulated annealing approach to solve small and large instances of the problem. Lim et al. (2007) proposed a new formulation for the QCSP with complete bays and showed that there is always an optimal solution of the problem among all possible unidirectional schedules of cranes. A simple approximation heuristic and simulated annealing heuristic was designed to solve the problem. Lee et al. (2008) provided another proof of NP-completeness of the QCSP with complete bays, and proposed an efficient genetic algorithm to address the problem. The efficiency of the genetic algorithm was tested on forty random instances with large sizes and the experimental results showed that the Genetic Algorithm is very efficient since deviation to the lower bound was less than 0.9% on all instances.

Few papers in the literature were devoted to the study of the QCSP with bay areas. Steenken et al. (2001) addressed the problem with the aim of minimizing the difference in the use of any two cranes. They showed that for instances of practical size the problem leads to a partitioning problem that can be easily solved by straight-forward enumeration. Lu et al. (2012) proposed an efficient heuristic based on unidirectional movements of cranes for the problem, the heuristic achieved a good trade-off between solution quality and computational time.

The QCSP with group of containers represents the most complex and realistic variant of the QCSP in literature. This variant was introduced by Kim and Park (2004). In this work a task was defined to be a collection of containers located adjacent to each other and that share the same characteristics. Namely, the same port of origin or of destination and the same size. The authors proposed a MIP formulation of the problem, with the aim to minimize the weighted sum of makespan and quay cranes finishing times. This formulation takes into account non-interference constraints between quay cranes, precedence among the handling operations of tasks and the availability date of quay cranes. To avoid collisions between cranes, the non-interference constraints were enforced by a non-simultaneity constraint between tasks located in adjacent bays. A Branch and Bound method and a Greedy Randomized Adaptive Search Procedure (GRASP) were proposed to solve the problem. Results showed that the Branch and Bound method outperforms GRASP in terms of solution quality but fails for largest instances. Moccia et al. (2006) revised the formulation proposed in (Kim & Park, 2004), to avoid some cases where interference cannot be detected. They significantly improve the results found by Kim and Park (2004) using a Branch and Cut algorithm. In (Sammara et al., 2007), the QCSP with group of containers was seen as a combination of a routing and scheduling problems. The routing problem was solved by a Tabu Search heuristic, and a Local Search technique was used for the resolution of the scheduling problem. Experiments showed that the proposed algorithm reduced significantly computation time for the largest instances, compared to the Branch and Cut of Moccia et al. (2006) with a slightly weaker quality of solutions. The MIP formulation developed in Kim and Park (2004) was also improved by Bierwirth and Meisel (2009), the authors introduced a new set of interference constraints and a fast Unidirectional Scheduling (UDS) heuristic based on the Branch and Bound algorithm was proposed to solve the resulting problem. Numerical tests revealed that UDS heuristic outperforms all previous existing algorithms in the literature, in terms of computational time and solution quality. In later work of Chung and Choy (2012) a Genetic Algorithm was proposed to deal with the QCSP. Experiments were executed using Kim and Park benchmarks, and results proved that the Genetic Algorithm is competitive and efficient as compared to the existing algorithms. Monaco and Sammarra (2011) studied the QCSP with group of containers under the hypotheses that cranes can only make unidirectional moves and that their availability is given within a predefined times windows. A Tabu Search heuristic was developed to solve the problem, and its efficiency was tested on known Kim and Park benchmarks and on a real-world

application on the Italian Gioia Tauro terminal. In (Wang & Kim, 2011), the QCSP was combined with yard management problem with the aim of minimizing the time spent by vessels in the port and minimizing the difference of workloads between yard blocks. The problem was solved by a GRASP meta-heuristic. New benchmarks for the QCSP were proposed by Meisel and Bierwirth (2011), these instances tests provide 400 sets ranging from 10 to 100 tasks and 2 to 6 cranes. They were widely used in some recent works by (Kaveshgar et al., 2012), (Unsal and Oguz, 2011), (Chen et al., 2014) and (Rouky et al., 2015). In the work of Nguyen et al. (2013), hybrid evolutionary computation approaches based on Genetic Programming and Genetic Algorithm were proposed to address the QCSP. Computational results demonstrated that the proposed methods perform as well as the existing methods, and that they were able to obtain better solutions than the best known ones in many instances.

Although a considerable attention has been paid in literature to the different variants of the QCSP, to our best knowledge, very few publications studied the QCSP with uncertainties. Legato et al. (2010) were the first that addressed the QCSP while taking uncertainties in consideration. They considered uncertainties that arise in handling process and proposed a Simulated Annealing algorithm to solve the QCSP and a Discrete Event Simulation to compute the expected cost of the solutions. However, the authors provided only one instance to evaluate the efficiency of their results. In a recent work, AL-Dhaheri et al. (2016) studied a problem that combine Quay Crane Scheduling with complete bays and Straddle Carriers Routing with the aim of increasing the container terminal throughput. While considering the randomness and dynamics related to containers discharging process. A simulation based Genetic Algorithm was proposed for the resolution. The numerical tests demonstrated the significance of using simulation to obtain more realistic solutions.

Our contribution in this paper is different from previous works of Legato et al. (2010) and AL-Dhaheri et al. (2016); Since, on the one hand, the simulation procedure is used in this paper inside the optimization approach to evaluate every possible solution obtained whitening the optimization algorithm rather than only evaluating the best solution as it was proposed in (Legato et al., 2010). On the other hand, this work is devoted to the study of the uncertain QCSP with group of containers, which is known to be more complex than the QCSP with complete bays studied in (AL-Dhaheri et al., 2016). Moreover, to the best of our knowledge, this is the first time in literature when the integration of an Ant Colony Optimization (ACO) with a Simulation procedure is considered to solve an uncertain QCSP problem.

3. Problem description

As explained in section 1, the QCSP consists of determining the best schedule of containers handling operations by a set of quay cranes assigned to a vessel. By extension the uncertain QCSP studied in this paper can be defined as a QCSP in which the handling time of tasks and travel time of quay cranes between location of tasks are supposed uncertain and they are given by probability distributions. Formally, in the uncertain QCSP a vessel is divided to a set of locations, $B = \{1, \dots, |B|\}$ denoted by bays. Bays are used for the storage of a set of tasks $T = \{T_1, \dots, T_N\}$, that represents the loading and unloading operations that must be executed by a set of quay cranes $Q = \{Q_1, \dots, Q_C\}$ assigned to the vessel. Each task T_i has a position l_i , expressed by a bay number. Quay Cranes (QCs) and bays are both supposed indexed in ascending order from left to right. Processing time of tasks and travel time of quay cranes between adjacent bays are assumed to be independent random variables with known distribution; The uncertain process time \tilde{P}_i of task T_i follows a 32-Erlang distribution with an expected value P_i , and the uncertain travel time \tilde{t} of a quay crane between two adjacent bays follows a Triangular distribution with a lower bound of 1 minute, mode of 1.5 minutes, and upper bound of 2.5 minutes. Precedence constraints are defined between tasks to respect the stacking plane of the containers. Since in the QCSP with group of containers, a bay is carrying several tasks, whitening the same bay; tasks that represent unloading operations must be performed before those of loading operations, the unloading operations on the deck of the vessel must precede the unloading on its hold and the loading on the deck can only start after the loading on the hold. We denote by ϕ the set of all pairs of tasks linked by precedence relationship, and

by e_i the set of all tasks that must be accomplished before executing task T_i . Moreover, quay cranes are mounted on the same track therefore they are not allowed to cross each other and at most one quay crane can operate on a bay at a time. Furthermore, to avoid collisions between cranes some tasks that are located in adjacent bays cannot be processed simultaneously. Let denote by ψ the set of all task pairs that have non-simultaneous relationship and by E_i the set of tasks that cannot be performed simultaneously with task T_i . Finally, for each quay crane Q_C , an initial position l_C^0 and an initial ready time r_C^0 are given.

Table 1

An example of input data for the uncertain QCSP with group of containers

Task number	1	2	3	4	5	6	7	8	9	10
Bay position (l_i)	2	10	3	2	6	2	7	7	3	5
Type of operation	L	L	U	U	U	L	U	L	U	U
Type of task	H	D	D	H	D	D	D	H	H	H
Expected Values of Processing time (P_i)	41	19	6	12	37	34	48	10	56	3
Precedence relationships	$\phi = \{(T_4, T_1), (T_4, T_6), (T_1, T_6), (T_3, T_9), (T_7, T_8)\}$									
Non-simultaneous pairs	$\psi = \{(T_4, T_3), (T_4, T_9), (T_1, T_3), (T_1, T_9), (T_6, T_3), (T_6, T_9), (T_{10}, T_5), (T_5, T_7), (T_5, T_8)\} \cup \phi$									
Quay cranes	1	2								
Initial bay position of QCs	1	6								
Initial ready time of QCs	0									
Processing time distribution	32-Erlang distribution									
Travel time distribution	Triangular distribution									

L: Loading operation; U: Unloading operation; H: Hold; D: Deck

Table 1 presents the input data for an instance of the uncertain QCSP. In this instance, 2 quay cranes are used to perform the handling of 10 tasks located in a vessel that is divided to 10 bays. Rows from 2 to 5 present the attributes of tasks, and they show respectively for each task; its location in the vessel given by the bay number, the nature of the operations (i.e. Loading (L) or Unloading (U)), if the task is positioned on the Deck (D) of the vessel or in its Hold (H) and the expected processing time P_i . The sets of precedence and non-simultaneously relationships are reported in rows 6 and 7. Quay cranes are supposed to be available from the starting of the planning horizon and they are located in bay1 and bay6, respectively. Fig 1. (a) gives an illustration of this instance, and Fig 1.(b) provides a simple representation of precedence relationships. Three tasks are located in bay2; an unloading operation T4 and two loading operations T6 and T1. Thus, T4 has to be accomplished before the starting of tasks T6 and T1, and task T1 must precede T6 since loading on hold precedes the loading on deck.

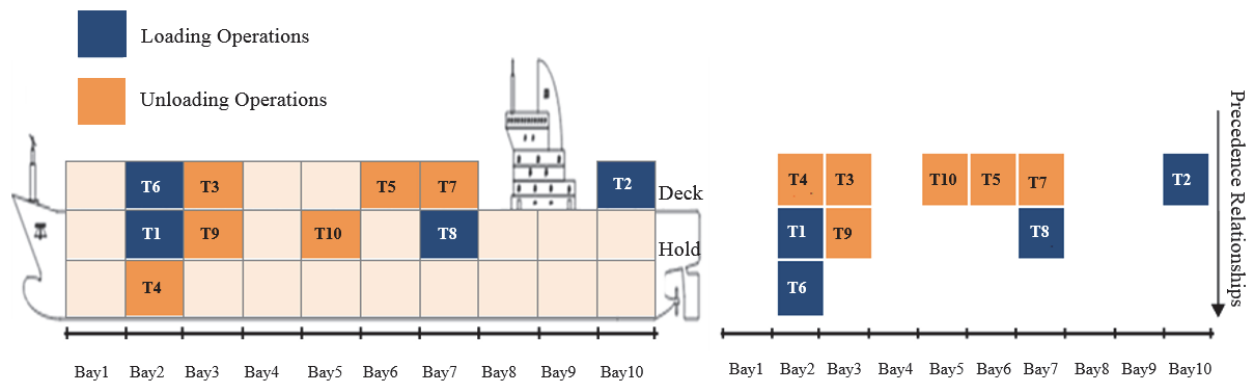


Fig. 1. Illustration of an instance of the Uncertain QCSP (a) and a simple representation of precedence relationships (b)

The QCSP can be viewed as a parallel identical machines scheduling problem, which is known to be NP-Hard (Michael, 1995). Therefore, we proposed in the next section a Simulation-Optimization based Ant Colony algorithm to solve the uncertain QCSP.

4. Solution methodology

A variety of approaches, such as fuzzy programming, robust optimization and stochastic optimization, has been used in the literature to deal with uncertainty in logistics and production systems, good reviews of these approaches are given in (Sahinidis, 2004) and (Gabrel et al., 2014). However, the level of detail and the accuracy described by those approaches are most of the time very insufficient since perturbations on real-world systems are too complex to be modeled analytically (Figueira & Almada-Lobo, 2014). Simulation provides a good way to model details of the problem, nevertheless, when it is used alone, it can only evaluate the performance of some possible alternative organizations of the system, which is not sufficient from the optimization point of view (He et al., 2013). Thanks to the tremendous development in computer performance in recent decades, combined Simulation Optimization (SO) approaches have received great attention, since they provide an intelligent way to explore simultaneously the great possibilities offered by the simulation to model the real details of the problem and the capacity of the optimization to find solutions with good quality.

SO approaches have been successfully applied to a wide range of problems arising in port logistics (Abourraja et al., 2017; Benghalia et al., 2016), rail transportation (Tréfond et al., 2017), risk management (Better et al., 2008) and production systems (Lim et al., 2006), among others. In particular, they have shown their greatest advantage when applied to highly dynamic and uncertain problems. Where only local information is available, such as the supply chain management under demand uncertainty (Jung et al., 2004), the stochastic location-routing problem (Herazo-Padilla et al., 2015) and the container yard design problem under uncertainty (Zhou et al., 2016). In this section we propose a new Simulation Optimization based ant colony heuristic to solve the uncertain QCSP with the aim to minimize the expected value of the compilation date of the last task in the vessel (makespan). Section 4.1 presents the general framework of the proposed Simulation Optimization approach, while the detailed steps are completely described in sections from 4.2 to 4.5.

4.1 General Framework of the Simulation Optimization Approach

As it is shown in Fig. 2, the SO-based ACO approach starts by loading input data from the instance file. Input data consist of the number of tasks, the number of QCs assigned to the vessel, the location of tasks, type of tasks and their position in the Deck or in the Hold of the vessel. An iteration I of the SO approach begins by setting the position l_c of each QC and its ready time r_c to their initial values. An ant procedure is then executed to determine a set of feasible solutions of the QCSP problem, in this procedure we use a set of m ants $\{f^1, f^2, \dots, f^m\}$, each ant f^k has to find a feasible schedule $X_0(I, f^k)$ which determines an initial assignment of tasks to QCs and the sequence of the execution of tasks by the QCs with respect to the precedence, the non-interference and non-simultaneous constraints, and by using only the nominal values of processing times of tasks and nominal travel times of QCs between adjacent bays. Next, a Variable Neighborhood Decent (VND) algorithm is executed on each initial schedule $X_0(I, f^k)$ obtained by the Ant procedure, and the resulting improved schedule is denoted by $X_1(I, f^k)$.

Whereas, the simulation procedure is then executed to evaluate each improved schedule on realistic scenarios generated on concordance with the distribution probability of travel times and processing times. At the end of the simulation the expected value of makespan of each improved solution is computed and compared, and a global update of pheromones is executed to increase pheromones over the edges of the best found solution at the iteration. The approach stops when a maximal number of iterations is reached and the best current solution X_{best} is returned.

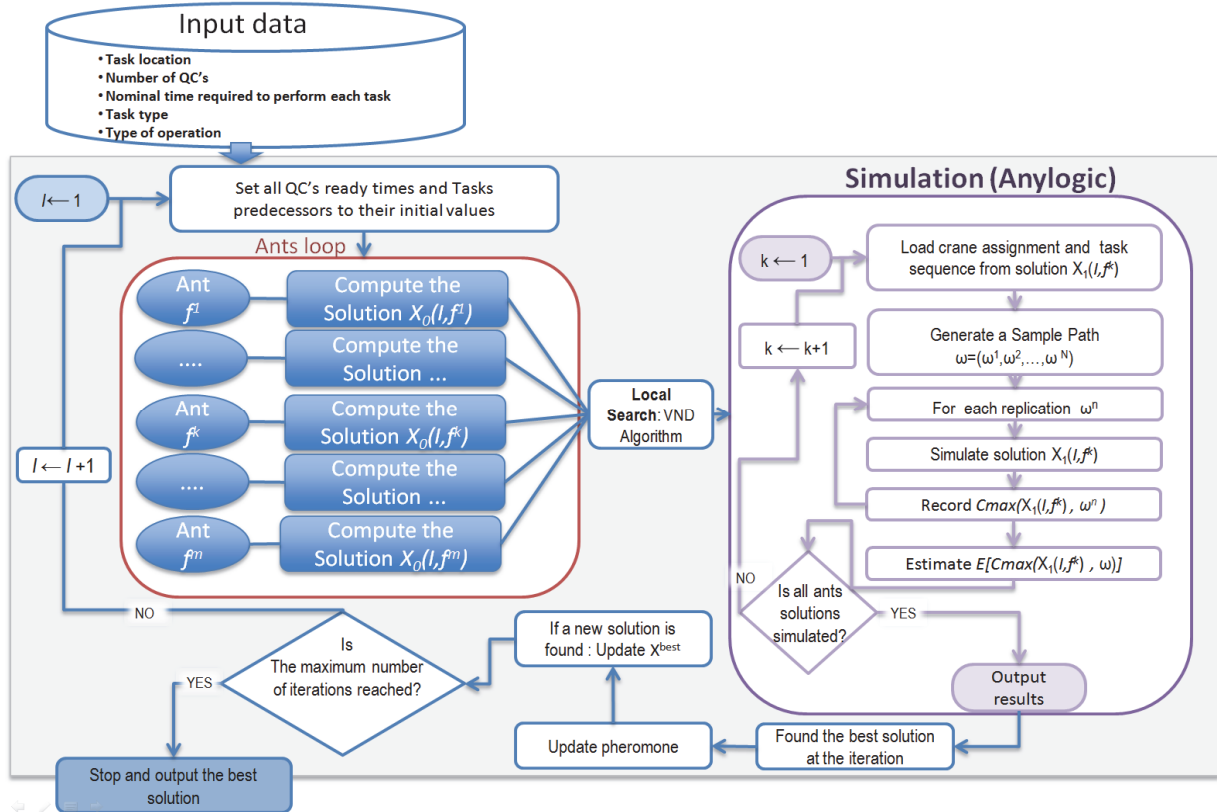


Fig. 2. General structure of the Simulation Optimization based Ant colony algorithm

4.2 Ant Colony procedure

Ant Colony Optimization (ACO) is an algorithm that was initiated by Dorigo and Caro (1999). ACO was inspired by the behavior of ants in real life. Ants start by exploring the area surrounding their nest randomly; then, on their way back to the nest, they lay a substance called pheromone. Pheromones guide other ants towards the target point, since pheromones on the paths used by several ants will be reinforced and these paths will be more interesting to the next ants. ACO algorithm is considered as one of the best choice to solve scheduling problems since the literature about these problems has demonstrated its effectiveness to give good solutions (Rajendran et al., 2004; Hirsch et al., 2012; Thiruvady et al., 2016; Bencheikh et al., 2016; El Khoukhi et al. 2017).

4.2.1 Graphical representation

In each iteration I of our proposed SO approach a ACO Structure that include a set of m ants $\{f^1, f^2, \dots, f^m\}$ is used. Each ant f^k build a initial schedule $X_0(I, f^k)$ well considering tasks and QCs attributes and the initial values of travel time of QCs and of processing times of tasks. The ants run through a bi-level graph (Fig. 3), vertices on the first level of this graph represents the QCs where vertices on its second level represents tasks. Two dummy vertices S and F are added to this graph to represents the beginning and the end of one ant move.

4.2.2 Choice of a quay crane

Each ant f^k starts building its own schedule from the initial position S . Selects the first quay crane Q_c to be used according to the first transition rule defined in Eq. (1), where q a real number chosen randomly in the interval $[0, 1]$, and q_0 is a parameter of the algorithm. Thus, depending on the value of q , we favorite either the choice of the quay crane with the lowest available date r_c , or we select a random quay crane Q_0 .

$$P_{Sc}^k = \begin{cases} \operatorname{argmin}_{c=1..C} r_c & \text{if } q < q_0 \\ Q_0 & \text{otherwise} \end{cases} \quad (1)$$

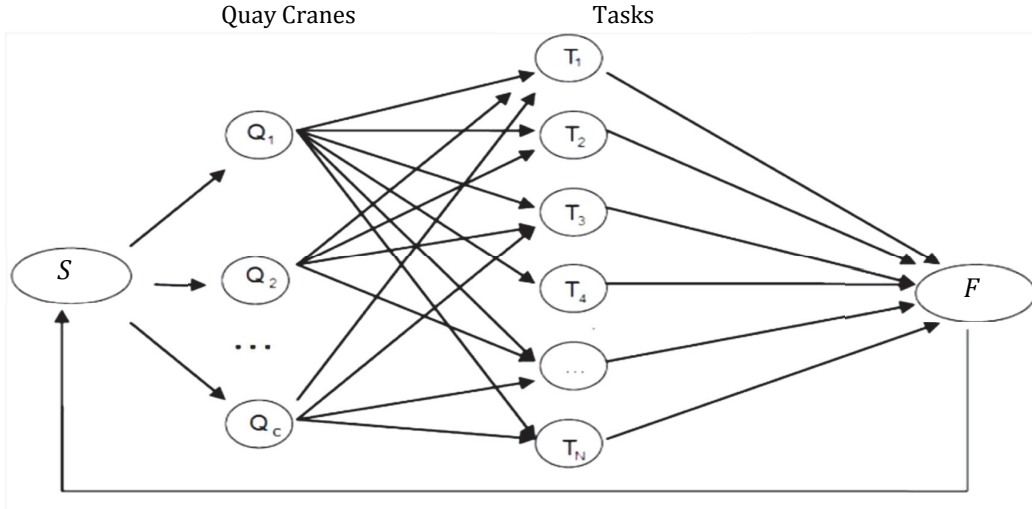


Fig. 3. Proposed Bi-level graph for the displacement of ant

4.2.3 Candidate set

Once a quay crane Q_c is selected, the ant f^k creates a set of candidate tasks that can be executed by the current crane. The set of candidate tasks of an ant is denoted by J^k and contains all tasks T_i that are not yet assigned to a quay crane (i.e. that are not in the set O^k of tasks already selected by the ant f^k) and that their list of predecessor e_i is empty.

$$T_i \in J^k \text{ if } \begin{cases} e_i = \emptyset \\ T_i \notin O^k \end{cases} \quad (2)$$

4.2.4 Choice of a task

Then, the ant selects from J^k the next task T_i to be affected to the current crane Q_c according to the second transition rule defined in Eq. (3).

$$P_{ci}^k = \begin{cases} \frac{(\tau_{ci})^\alpha (\eta_{ci})^\beta}{\sum_{j \in J^k} (\tau_{cj})^\alpha (\eta_{cj})^\beta} & \text{if } i \in J^k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where, τ_{ci} shows the amount of pheromone concentration in edge (c, i) . The pheromone represents the memory of the ACO and it is used to promote displacements on edges that have been selected by a large number of ants. It is initialized at the beginning of the SO approach by a small value τ_0 , and is locally updated in the ACO at each time when a task is selected as shown in the section 4.2.6. A global update of pheromone trail is also performed at the end of each iteration of the SO approach according to the mechanism described in section 4.4. η_{ci} represents the heuristic information associated with assigning task T_i to the current crane Q_c , this information provides worthy information about the problem for guiding the search procedure. Two different strategies are proposed in the section 4.2.5 for the heuristic information. Parameters α and β are introduced to control the search direction, which determine the relative intensity of the pheromone trace and the heuristic information.

4.2.5 Heuristic Information

In order to converge to good schedules in the ACO procedure, we proposed two different strategies for the heuristic information.

- *Earliest Start Time strategy (EST)*: This strategy is based on the Earliest Start Time (EST) of candidate tasks, this time is defined in Equation (4) as the first date on which the current quay crane Q_c can start the handling of a candidate task $T_i \in J^k$, without violating the non-interference and non-simultaneous constraints.

$$EST_i = \max(r_c, \max_{j \in O^k \cap E_i} C_j, \max_{\substack{s < c \\ l_s > l_i}} r_s, \max_{\substack{s > c \\ l_s < l_i}} r_s) \tag{4}$$

The heuristic information associated to the EST strategy is defined by Eq. (5) as:

$$\eta_{ci} = \frac{1}{1 + EST_i} \tag{5}$$

- *Local Work Load strategy (LWL)*: This strategy is based on the Local Work Load (LWL) of a candidate task $T_i \in J^k$, which is defined in Equation (6) as the sum of initial processing time P_j of remaining tasks T_j in the radius of 2 bays from the location l_i of task T_i .

$$LWL_i = \sum_{j \in L} P_j, L = \{T_j, |l_i - l_j| \leq 2\} \tag{6}$$

The heuristic information associated to the LWL strategy is defined by Eq.n (7) as:

$$\eta_{ci} = \frac{P_i + \hat{t}|l_c - l_i|}{1 + LWL_i}, \tag{7}$$

where P_i represents the initial processing time of candidate task T_i , \hat{t} is the initial travel time of a quay crane between two adjacent bays and thus $\hat{t}|l_c - l_i|$ gives the required travel time of quay crane Q_c from its current position l_c to the bay location l_i of candidate task T_i .

4.2.6 Compilation time, ready time and Local update of pheromone

Once a task T_i is assigned to the current crane Q_c , the ant f^k updates its state and moves to the vertices F . The ant adds the selected task to the set O^k of tasks already selected and deletes the task from the predecessor sets e_j of all other tasks such as $T_i \neq T_j$. Then we assign, using Eq. (8) a completion time C_i to the selected task T_i and we set the ready time r_c of the current quay crane Q_c to C_i .

$$C_i = \max(r_c, \max_{j \in O^k \cap E_i} C_j, \max_{\substack{s < c \\ l_s > l_i}} r_s, \max_{\substack{s > c \\ l_s < l_i}} r_s) + \hat{t}|l_c - l_i| + P_i \tag{8}$$

When the ant moves to vertices F , a local update of pheromone is performed on the edge (c, i) of the graph according to the following equation:

$$\tau_{ci} = \tau_{ci} + (1 - \rho)\tau_0 \tag{9}$$

Steps from 4.2.2 to 4.2.6 are repeated until that all tasks were be selected by the current ant f^k . Then we start the construction of the schedule $X_0(I, f^{k+1})$ of the next ant f^{k+1} . The ACO procedure is summarized in *Algorithm 1*.

Algorithm 1: Ant Procedure

```

1: Input: Iteration I; Heuristic Strategy (HS): EST or LWL; Pheromone trail  $\tau$ 
2: for  $k \leftarrow 1$  to  $m$  do
3:   Initialize all predecessor sets  $e_i$  of tasks
4:   For each quay crane  $Q_c$ : Set the ready time  $r_c$  to  $r_c^0$  and the current position time  $l_c$  to  $l_c^0$ 
5:    $X_0(I, f^k) \leftarrow \emptyset$  and  $O^k \leftarrow \emptyset$ 
6:   repeat
7:     Assign to parameter  $q$  a random value from the interval  $[0,1]$ 
8:     if  $q \leq q_0$  do
9:        $Q_c \leftarrow \underset{c=1..C}{\operatorname{argmin}} r_c$ 
10:    else
11:      Select a random crane  $Q_c \leftarrow Q_0 \in Q$ 
12:    end if
13:    for  $i \leftarrow 1$  to  $N$  do
14:      if  $((T_i \notin O^k) \wedge (e_i = \emptyset))$  do
15:        Insert  $T_i$  in the set of candidate tasks  $J^k$ 
16:      end if
17:    end for
18:    for each task  $T_i \in J^k$  do
19:      if HS=EST do
20:         $EST_i \leftarrow \max(r_c, \max_{j \in O^k \cap E_i} C_j, \max_{\substack{s < c \\ l_s > l_i}} r_s, \max_{\substack{s > c \\ l_s < l_i}} r_s)$ 
21:         $\eta_{ci} \leftarrow \frac{1}{1+EST_i}$ 
22:      else if HS=LWL do
23:         $LWL_i = \sum_{j \in L} P_j, L = \{T_j, |l_i - l_j| \leq 2\}$ 
24:         $\eta_{ci} \leftarrow \frac{P_i + \hat{t}|l_c - l_i|}{1+LWL_i}$ 
25:      end if
26:    end for
27:    Select the Task  $T_i$  to assign to the current crane  $Q_c$  with probability  $\frac{(\tau_{ci})^\alpha (\eta_{ci})^\beta}{\sum_{j \in J^k} (\tau_{cj})^\alpha (\eta_{cj})^\beta}$ 
28:    Add  $T_i$  to  $O^k$  and remove  $T_i$  from all predecessor sets  $e_j$  of tasks  $T_j$  such as  $T_i \neq T_j$ 
29:     $C_i \leftarrow \max(r_c, \max_{j \in O^k \cap E_i} C_j, \max_{\substack{s < c \\ l_s > l_i}} r_s, \max_{\substack{s > c \\ l_s < l_i}} r_s) + \hat{t}|l_c - l_i| + P_i$  and set  $r_c$  to  $C_i$ 
30:    Execute a local update of pheromone:  $\tau_{ci} = \tau_{ci} + (1 - \rho)\tau_0$ 
31:     $X_0(I, f^k) \leftarrow X_0(I, f^k) \cup \{< Q_c, T_i, C_i >\}$ 
32:  until (all task are selected)
33:  return ( $X_0 = \{X_0(I, f^1), \dots, X_0(I, f^m)\}$ )

```

4.3 Variable Neighborhood Descent

We use a Variable Neighborhood Decent algorithm (VND) as a general local search to improve schedules obtained by the ACO algorithm. VND is considered as a single solution based meta-heuristic that is known for its ability to escape from local optimum (Hansen et al., 2010). The proposed VND algorithm is executed on each ant schedule, it begins with an initial schedule $X_0(I, f^k)$ and attempts iteratively to improve it by employing a set of three neighborhood structures. The sequence order of exploring the neighborhood structures is randomly generated, at each call of the algorithm. The VND executes the first neighborhood in the generated order as long as an improvement is obtained, and moves to the next one in the sequence when the previous fails to lead to improvement. VND stop when all neighborhood structures were applied. Steps of the VND algorithm are given in *Algorithm2* and the details about the neighborhood structures are described bellows.

- *Swap1*: Randomly select a quay crane and examine all possible swaps between each pair of tasks, only possible swaps are considered.

- *Swap2*: Randomly select two different quay cranes and examine all possible swaps of tasks between the selected cranes, only possible swaps are considered.

- *Relocate*: Randomly select a quay crane and examine all possible moves of tasks that are currently assigned to this quay crane to a different position on the same quay crane, only possible moves are considered.

Algorithm2: *Variable Neighborhood Decent*

```

1: Input: Iteration I; a initial schedule  $X_0(I, f^k)$ 
2: Let  $\Gamma$  be the number of Neighbourhoud Structure (NS)
3: Genarate the neighbourhouds sequence execution randomly
4:  $it \leftarrow 1$ 
5:  $X_1(I, f^k) \leftarrow X_0(I, f^k)$ 
6: do
8:   Generate a neighborhood  $X'(I, f^k)$  of  $X_1(I, f^k)$  using  $NS_{it}$ 
9:   if  $Cmax(X'(I, f^k)) < Cmax(X_1(I, f^k))$  do
10:      $X_1(I, f^k) \leftarrow X'(I, f^k)$ 
11:   else
12:      $it \leftarrow it + 1$ 
13:   end if
14: while ( $it < \Gamma$ )
15: return  $X_1(I, f^k)$ 

```

4.4 The Simulation procedure

Simulation experiments are performed using Compare-run tool of Anylogic Software (Multimethod Software, 2015). AnyLogic is a multi-approach simulator, equipped with a rich and easy-to-master toolbox, that allows to effectively combine the various simulation techniques (agent based simulation, discrete-event simulation and system dynamics). Anylogic is used in this paper for the analysis, the collection and the comparison of results as well as for extension, visualization and experimentation of simulation, which allows us to considerably reduces the time of developing the simulation model.

The simulation procedure starts by generating a simple path $w = \{w^1, w^2, \dots, w^N\}$, each element of this path represents one replication of the simulation experiment. Each replication w^n defines one possible scenario generated on concordance with the distribution probability of the uncertain parameters. Travel time of QCs between adjacent bays are generated according to a Triangular distribution with a lower bound of 1 minute, mode of 1.5 minutes and an upper bound of 2.5 minutes. Process time of tasks follows a 32-Erlang distribution with an expected value P_i . A Discrete Event Simulation (DES) is then used to evaluate the schedules obtained in the optimization procedure and to correct the starting and completing time of tasks according to each generated replication. The DES starts by loading the tasks assignment and operation sequences of QCs from schedule $x_1(I, f^k)$ given by the optimization model, and the values of the uncertain parameters from replication w^n . We use the status of each QC, i.e. either "idle" "busy" or "cannot be selected", and the number of tasks waiting to be executed by each QC as two variables to describe the state of the simulated system. QCs are marked as "busy" until their ready times is reached. Then, the first available QC is selected and moves to the bay position of the first task in its sequence and the handling operations of containers in this task start. After that, the selected QC changes its statue from "idle" to "busy" and we move to select another available crane. In the case where several QCs show equivalent ready times, priority is given to the QC with the lowest index. Before performing any moves of a QC, we check if this move will cause interference, if it is the case, we change the statue of the QC to "cannot be selected now" and we select another idle QC. The statue of a QC change also to "cannot be selected now" when the first task in its sequence cannot be executed simultaneously with one of tasks that are on execution by other QCs. When the handling operations of a task finish, its assigned QC changes its statue to "idle" and the number of tasks waiting to be executed by this QC decrease by 1. The simulation process finishes when the number of tasks waiting in all QCs is equal to 0, and the DES return the completing time of the last task executed. This experiment is performed for all improved schedules $X_1(I, f^1), \dots, X_1(I, f^m)$ obtained at the end of the optimization procedure, and the expected value of the makespan of each schedule is computed by:

$$E(Cmax(X_1(I, f^k), w)) = \frac{1}{N} \sum_{n=1}^N Cmax(X_1(I, f^k), w^n) \quad (10)$$

Fig. 4 shows the general flowchart of the Discrete Event Simulation:

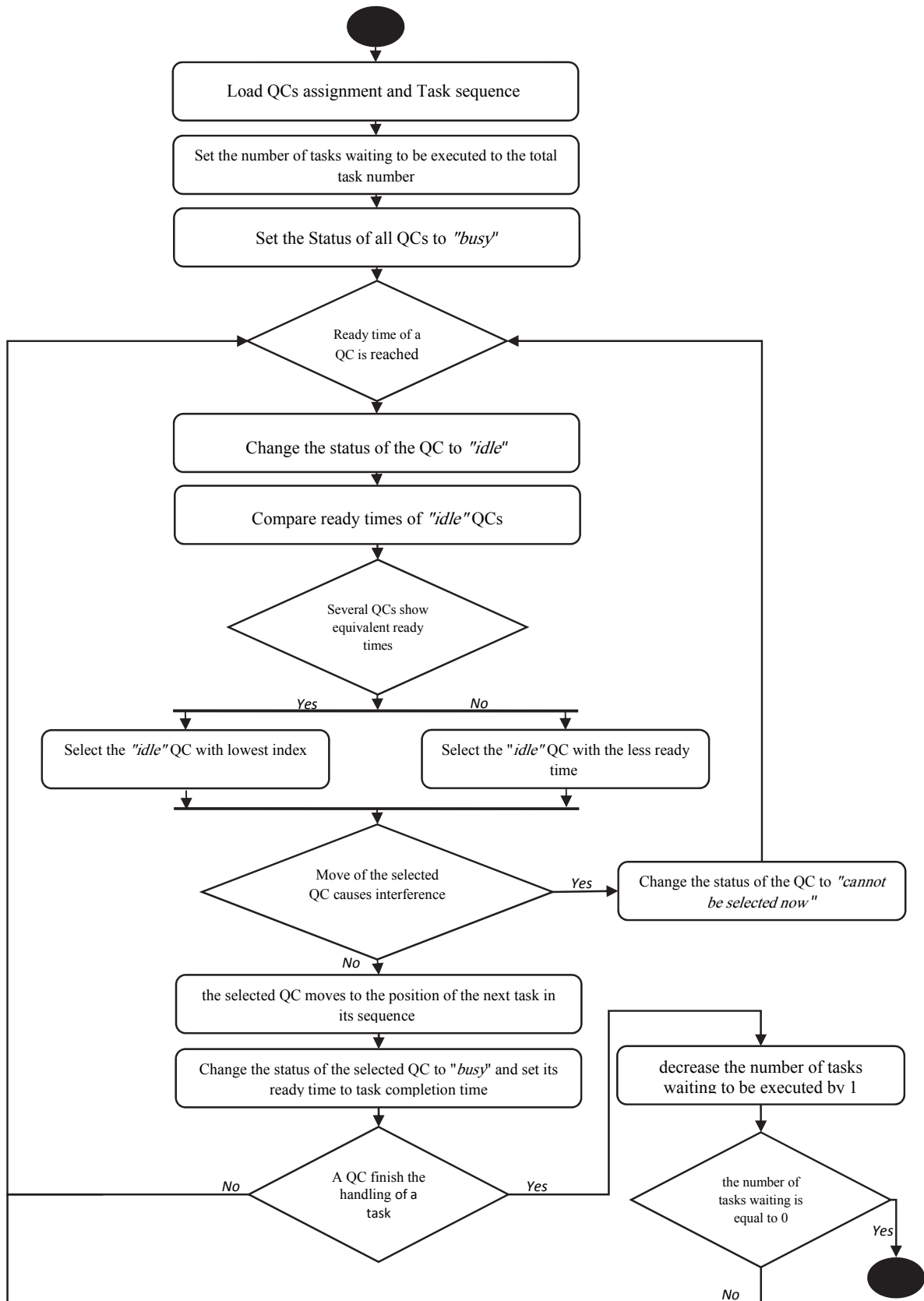


Fig. 4. Flowchart of the Discrete Event Simulation

4.5 Global Update of Pheromone

At the end of each iteration of the Simulation-Optimization approach, the different schedules are compared and a global update of pheromone trail is performed on arcs of the schedule with the less expected makespan, according to the following formulation:

$$\tau_{ci} = (1 - \rho)\tau_{ci} + \rho\Delta\tau_{ci} \tag{11}$$

where $\Delta\tau_{ci} = 1/E(Cmax(X_{Iterbest}))$ is the amount of pheromone added to the arc (c, i) of the best schedule at the iteration, and ρ is the coefficient of pheromone evaporation.

5. Numerical experiments

In this section several numerical experiments are executed to evaluate the performance of our developed SO-based ACO approach. First, the selection of the best parameters for both the ACO algorithm and the simulation procedure is investigated. Then, the performance of the proposed Hybrid Ant Colony Optimization (HACO) procedure (i.e. the ACO with the VND algorithm), is tested on the well-known data of Kim and Park under deterministic environment. In the stochastic case, two Gaps are recorded to compare the performance of the SO approach using both the EST and LWL strategies .

All experiments are carried out on an Intel(R) Core (TM) i5-3337U, 1.80 GHz PC with 6.00 GB of RAM.

5.1. Parameters Setting

5.1.1 Ant Colony parameters

In general, efficiency of any meta-heuristic depend on its parameters setting since a good tuning can allow a fast convergence to solutions with high quality well a random choice can cause bad performances.

The IRACE package (López-Ibáñez et al., 2016) is used in this section to select good combination of parameters used in the ACO algorithm. IRACE package performs an automatic algorithm tuning, which starts by generating a finite set of possible configurations. Then, compare their performance on a set of training instances. Elite configurations are then used to generate more good potential configurations of the parameters in next iterations. The procedure is repeated until the tuning budget, which is given by the maximal number of configurations to be examined, is reached. The current best combination of parameters is returned at the end of the procedure. Table 2 presents the results obtained for each parameter used in the ACO algorithm, The tuning budget in the IRACE package was set to 3000 experiments.

Table 2
Best combination of parameters

Parameter	Range	Best Value for ACO with EST strategy	Best Value for ACO with LWL strategy
Number of ants m	{5,10,25,30,50}	10	10
Maximal number of iterations I_{max}	{300,500,1000,1500,2000}	1000	1000
Initial value of pheromone trails τ_0	[0,1]	0.01	0.01
Probability of exploitation strategy q_0	[0,1]	0.8	0.7
Relative importance of pheromone trails α	{1,2,3,4,5}	1	2
Relative importance of the heuristic information β	{1,2,3,4,5}	2	2
Coefficient of evaporation ρ	{0.01, 0.02, 0.2, 0.5}	0.2	0.2

5.1.2 Number of replications in the Simulation

Compare-run tool of Anylogic Software provides a Parameter Variation Experiment that stops the Simulation procedure after a minimum number of replications, when the confidence level is reached. If the confidence level is not met, the Parameter Variation Experiment ends when the maximum number of

replications has been exceeded. The confidence level was fixed at 95% and the error percentage was set as 0.5. The minimum and maximum number of replications were set as 10 and 500, respectively.

5.2. Deterministic Results

In order to test the performance of the proposed optimization problem, we conduct a series of computational tests over the well-known benchmarking data of Kim and Park (2004). This benchmark is commonly used to test the efficiency of algorithms for the Quay Crane Scheduling Problems (QCSP). The data consists of 43 instances, from k13 to k49, divided on 4 sets. The first set represents the small size problem composed of 2 QCs and 10 tasks, set 2 contains medium size problem with 2 QCs and 15 tasks, where sets 3 and 4 are regarded as large size problem with 3 QCs, 20 tasks and 3 QCs, 25 tasks, respectively.

5.2.1 Performance Analysis

The results of the Hybrid ACO algorithm using both the Earliest Start Time (EST) strategy and the Local Work Load (LWL) strategy are compared to those from Branch-and-Bound (B&B) and Greedy Randomized Adaptive Search Procedure (GRASP) algorithms by Kim and Park (2004), Tabu Search (TS) meta-heuristic by Sammarra et al. (2007), Unidirectional Scheduling (UDS) heuristic by Bierwirth and Meisel (2009), Genetic Algorithm (GA) by Chung and Choy (2012), Hybrid Genetic Algorithm (HGA) and Hybrid Genetic Programming (HGP) by Nguyen et al. (2013). The relative percent deviation (Eq. 12) of each algorithm to the Lower Bound (LB) is used to evaluate the performance of the methods. Relative Percent Deviation (RPD) on an instance i is given by:

$$RPD_i = \frac{Best_i(H) - LB_i}{LB_i} \times 100 \quad (12)$$

where $Best_i(H)$ is the makespan obtained by heuristic H on instance i and LB_i is the Lower Bound obtained by CPLEX solver and reported in (Bierwirth & Meisel, 2009).

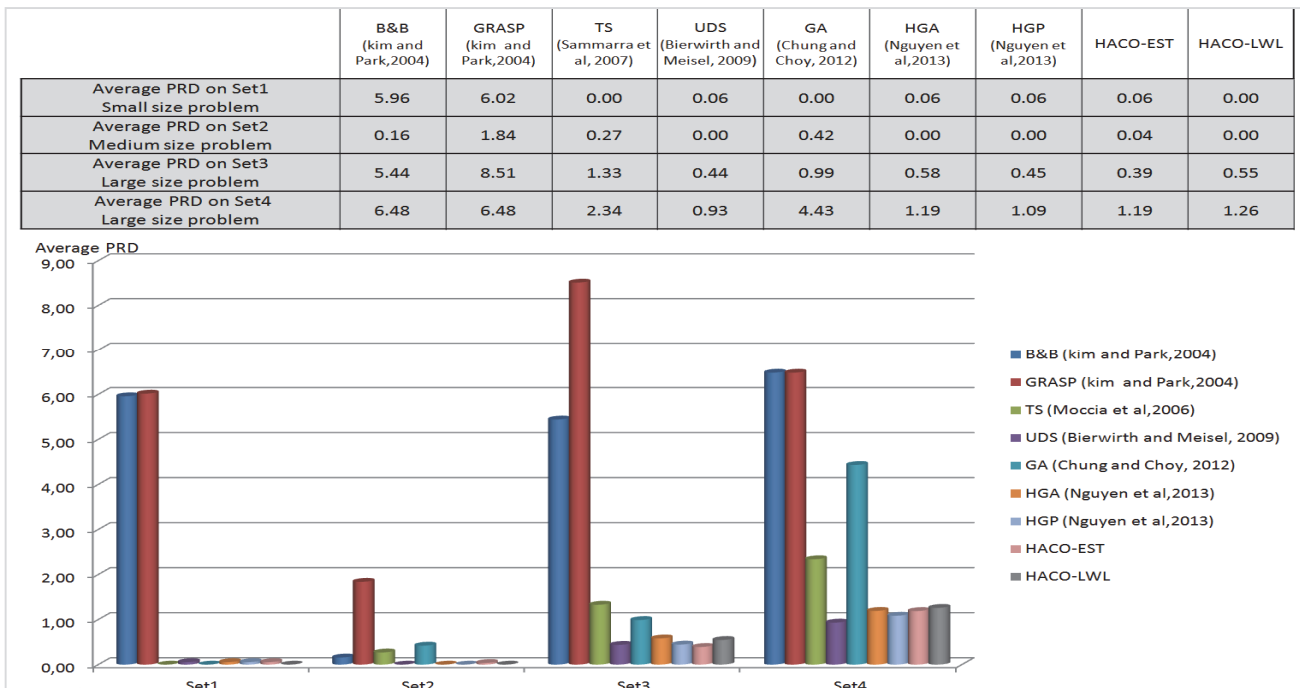


Fig 5. Performance evaluation of optimization procedure versus existing algorithms

Table 5 (Appendix A) shows the detailed comparison results, well Fig. 5 summarises these results and reports the average Relative Percent Deviation obtained on each set of instances. The results demonstrate

that both proposed strategies, i.e. *EST* and *LWL*, used in the HACO are very competitive and effective to solve the QCSP as compared to the existing algorithms in the literature. The results reveal that the Hybrid ACO with the *LWL* strategy is the only algorithm in literature that is able to achieve 0% relative deviation in both small and medium size problem with a good performance in large instances of set 3 and a slightly weak deviation in the instances of set 4. On the other hand, the Hybrid ACO with the *EST* strategy also gives good results when it is applied to small and medium instances of set 1 and set 2, since the average RPD is less than 0.06% in set 1 and equals to 0.04% in set 2. In addition, the HACO-*EST* outperforms all other methods on the large instances of set 3 with an average RPD equals to 0.39%, and shows quite similar performance to the UDS heuristic and HGP in the large instances of set 4.

5.2.2 Computational Time

It is generally very difficult to compare computational times since each method is implemented in different computer configuration. To make a fair comparison, we adopt an approach based on Million Floating Point Calculation per second (Mflops) as reported in (Dongarra, 2014). Table 4 shows the different Mflops values relevant to our study. Column 2 presents the configurations of the computers where each tested algorithm is implemented. In column 3, since some of the computers configurations are not listed in the report of Dongarra (2014), we give the equivalent machine configuration found in the report, and we report the corresponding approximate Mflops value for the particular machine in Column 4. Finally in the last column, We report the conversion factor *r* that will be used to scale the computational times of the algorithms. The conversion factor *r* presents the ratio of the Mflops value of a given computer configuration to the Mflops value of our computer configuration.

Table 3
Mflops values of the different computers configurations relevant to our study.

Approaches	Kim and Park (2004)		Sammarra et al. (2007)	Bierwirth and Meisel (2009)	Chung and Choy (2012)	Nguyen et al. (2013)		Proposed	
	B&B	GRASP	TS	UDS	GA	HGA	HGP	HACO-EST	HACO-LWL
Actual Computer Configurations used	P2, 466 MHz		P4, 2.5 GHz	P4, 2.8 GHz	i2, 2 GHz	i5, 3.10 GHz		Intel Xeon 2.4 GHz	
Approximate Equivalent Computer Configurations reported in (Dangarra, 2014).	P2, 450 MHz		P4, 2.53 GHz	P4, 2.8 GHz	P4, 2.8 GHz	P4, 3.06 GHz		Intel Xeon 2.4 GHz	
Million Floating Point Calculation per second <i>Mflops</i>	98		1190	1317	1317	1414		1055	
Conversion Factor <i>r</i>	0.09289		1.12796	1.24834	1.24834	1.49099		1	

Table 4
Computational times in minutes of B&B, GRASP, TS, UDS, GA, HGA, HGP, HACO-EST and HACO-LWL algorithms

Set of Instances	CPU	Kim and Park (2004)		Sammarra et al. (2007)	Bierwirth and Meisel (2009)	Chung and Choy (2012)	Nguyen et al. (2013)		Proposed	
		B&B	GRASP	TS	UDS	GA	HGA	HGP	HACO-EST	HACO-LWL
Set1 Small size problem	Average Time	0.44	0.35	1.52	1.12×10^{-5}	0.52	0.01	0.01	0.02×10^{-3}	0.3×10^{-3}
	Scaled average Time	0.41	0.33	1.71	1.39×10^{-5}	0.65	0.01	0.01	0.02×10^{-3}	0.3×10^{-3}
Set2 Medium size problem	Average Time	17.53	1.46	5.86	3.86×10^{-5}	0.75	0.04	0.03	0.01	0.01
	Scaled average Time	1.63	0.13	6.61	4.81×10^{-5}	0.93	0.06	0.04	0.01	0.01
Set3 Large size problem	Average Time	564.47	3.16	21.75	6.26×10^{-4}	1.18	0.18	0.20	0.04	0.03
	Scaled Average Time	52.43	0.29	24.53	7.81×10^{-4}	1.47	0.27	0.29	0.04	0.03
Set4 Large size problem	Average Time	809.73	7.56	48.68	3.43×10^{-3}	1.58	0.57	0.39	0.07	0.09
	Scaled Average Time	75.21	0.70	54.91	4.28×10^{-3}	1.97	0.85	0.58	0.07	0.09

Table 4 presents the results of computational times comparison. The average computational time corresponds to the original computing time of each algorithm in the literature, where the scaled average time is equal to the original computational time multiplied by the conversation rate r . Thus, the scaled time shows the computing time of the different algorithms if they had been executed on our computer. We can clearly see from results of Table 4 that our proposed algorithm is very efficient at solving the QCSP, because all instances are solved quickly in less than 0.1 minutes. Our proposed algorithms are advantaged, in terms of computational time, with respect to most existing algorithms. The UDS is the only heuristic that is faster than our proposed algorithm.

5.3. Stochastic Results

Previous deterministic results have shown that the hybrid ACO algorithm, with both proposed strategies EST and LWL defined for the selection of task, is very effective and efficient to solve the deterministic QCSP. This performance is very recommended in Simulation Optimization (SO) practice, because the choice of an optimization algorithm always depends on its computational efficiency (Kelly, 2002). In this section we will compare the results of both strategies when they are applied in the Simulation Optimization approach to deal with uncertainties. Then we demonstrate the significance of using Simulation Optimization to obtain more realistic solutions under stochastic assumption rather than employing deterministic approaches.

Table 6 (Appendix A) presents the detailed results for simulation tests based on the EST and LWL proposed strategies. To compare these strategies and selects the most adequate one for the SO approach, we recorded two gaps:

- $Gap1 = \frac{MaxValue[Cmax(LWL,w)] - MaxValue[Cmax(EST,w)]}{MaxValue[Cmax(EST,w)]} \times 100$: the deviation rate between the maximum makespan value obtained by the LWL strategy and the maximum makespan value obtained by the EST strategy, across all replications of the SO approach (each component of the sample path w).

- $Gap2 = \frac{E[Cmax(LWL,w)] - E[Cmax(EST,w)]}{E[Cmax(EST,w)]} \times 100$: the deviation rate between the expected makespan obtained by the LWL strategy and the expected makespan obtained by the EST strategy, among all replications of the SO approach (each component of the sample path w).

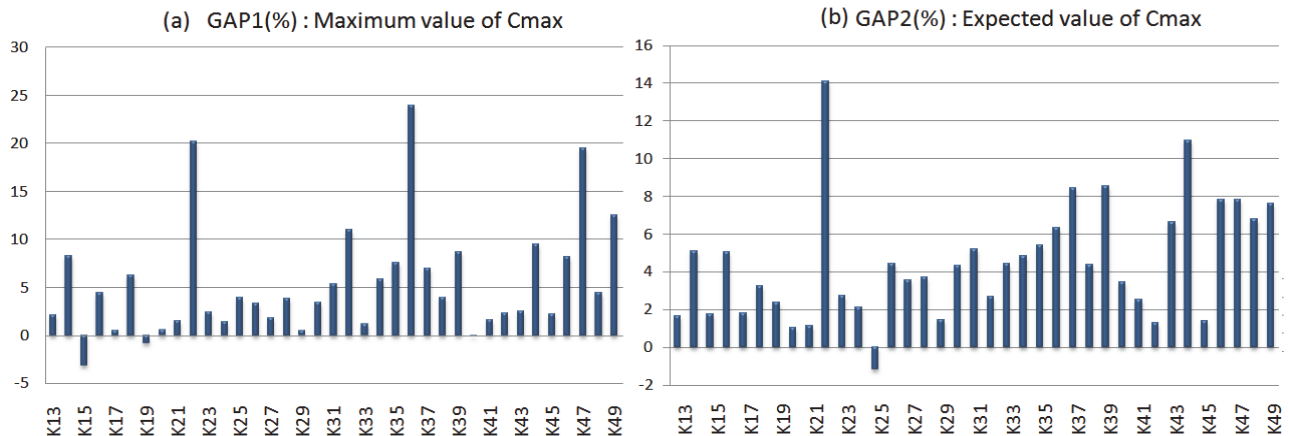


Fig 6. Results of maximum deviation rate (a) and expected deviation rate (b)

Fig. 6.(a) summarizes Gap1 values and shows that EST strategy outperforms the LWL strategy in regards to worst case scenarios, since the maximum makespan value (i.e. worst possible performance) of the former strategy is higher than these of the latter in most instances (positive gap), except in instances k15 and k19 where negative gaps of -3.12% and -0.86% are respectively observed. Moreover, in regards to the objective function that aims to minimize the expected makespan, the results of Gap2 reported on Fig. 6.(b) show clearly that the use of the EST strategy within the optimization procedure provides good

results for the SO approach on all tested instances since the only negative gap was observed in the instance k25 with a weak value of -1.19%.

Therefore, it is obvious from results presented in Fig. 7 that the use of EST strategy within the hybrid ACO procedure is more useful to deal with uncertainties than the LWL strategy, even if both strategies had shown similar performance in the deterministic case.

To demonstrate the significance of using the SO approach to deal with uncertainties we compare its results with those of deterministic solutions when they are put into operation with identical scenarios. For the SO solutions we use those obtained by applying the EST strategy, and for the deterministic solutions we consider on each instance the best solution over the EST and LWL strategies. Solutions are then simulated on 1000 scenarios which have stochastic quay crane travel time and tasks processing times. The relative improvement that SO solutions yield compared to deterministic solutions is recorded in Fig. 7.

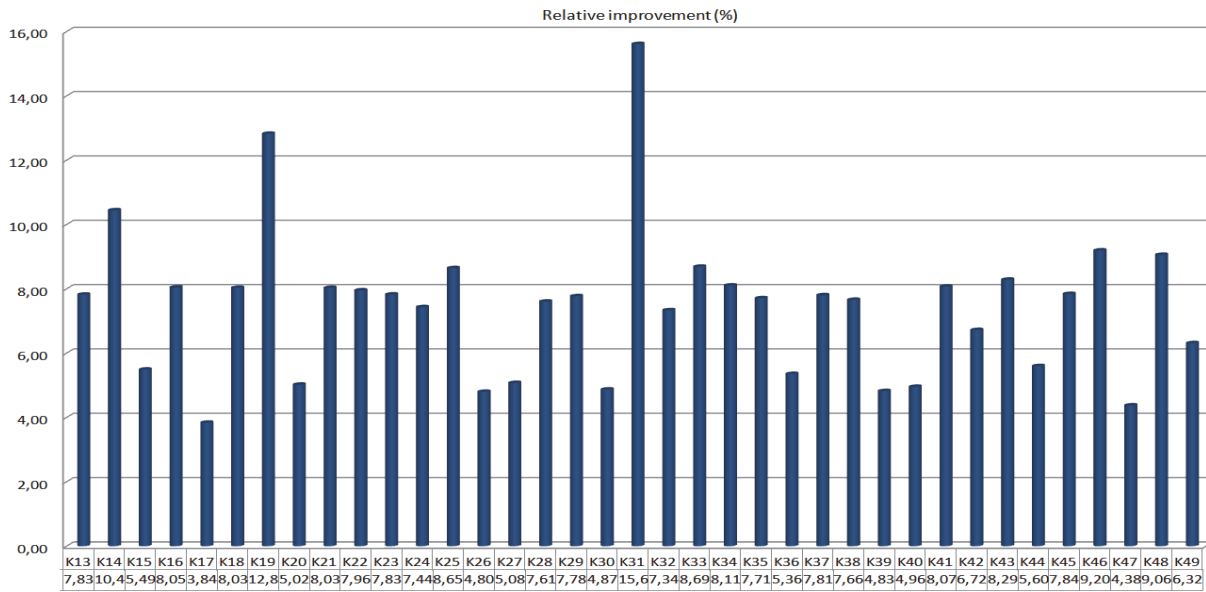


Fig 7. Relative improvement offered by SO solutions compared to deterministic solutions

The average improvement value was equal to 7.43% and the maximum improvement was as high as 15.6%. Therefore, we can conclude that the use of SO solutions are very useful to achieve a good robustness against uncertainties.

6. Conclusion

In this paper, we have considered the Quay Crane Scheduling Problem under stochastic assumptions, we have assumed that there is uncertainty in task processing time and quay crane travel time. To solve the resulting problem, we proposed a Simulation Optimization (SO) approach, which takes advantage of the great possibilities offered by the simulation to model the uncertain parameters of the problem and from the capacity of the optimization to find high quality solutions. In the optimization procedure we implemented a Hybrid Ant Colony Optimization algorithm which includes an Ant Colony and Variable Neighborhood Descend and we proposed two different strategies for the selection of tasks, namely; the Earliest Start Time (EST) strategy and the Local Work Load (LWL) strategy. A Discrete Event Simulation (DES) was used to evaluate the obtained solutions under different scenarios generated on concordance with probability distribution of the uncertain parameters. Since it is known that performance of SO approaches depend on computational efficiency of the selected optimization algorithm, our first numerical tests were devoted to the comparison, in deterministic case, between the HACO algorithm using EST and LWL strategies and other existing algorithms in the literature. The results have shown that the HACO algorithm, with both proposed strategies, is very effective and efficient to solve the deterministic QCSP. Then, a series of computational study was carried out on the stochastic case using

the proposed SO approach to test the efficiency of both strategies under uncertainty. Tests revealed that the use of the EST strategy within the hybrid ACO procedure is more useful to deal with uncertainty in the SO approach than the LWL strategy.

Acknowledgement

This research was co-financed by the European Regional Development Fund (ERDF) and by The Haute Normandie region under project ‘CLASSE – Corridors Logistics: Application to the Seine Valley and Its Environment’.

References

- Abourraja, M. N., Oudani, M., Samiri, M. Y., Boudebous, D., El Fazziki, A., Najib, M., ... & Rouky, N. (2017). A multi-agent based simulation model for rail–rail transshipment: An engineering approach for gantry crane scheduling. *IEEE Access*, 5, 13142-13156.
- Al-Dhaheri, N., Jebali, A., & Diabat, A. (2016). A simulation-based Genetic Algorithm approach for the quay crane scheduling under uncertainty. *Simulation Modelling Practice and Theory*, 66, 122-138.
- Bencheikh, G., Boukachour, J., & Alaoui, A. E. H. (2016). A memetic algorithm to solve the dynamic multiple runway aircraft landing problem. *Journal of King Saud University-Computer and Information Sciences*, 28(1), 98-109.
- Benghalia, A., Boukachour, J., & Boudebous, D. (2016). Gestion du transfert interne de conteneurs: le cas du port du Havre. *Logistique & Management*, 24(1), 57-69.
- Better, M., Glover, F., Kochenberger, G., & Wang, H. (2008). Simulation optimization: Applications in risk management. *International Journal of Information Technology & Decision Making*, 7(04), 571-587.
- Bierwirth, C., & Meisel, F. (2009). A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling*, 12(4), 345-360.
- 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.
- Bierwirth, C., & Meisel, F. (2015). A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 244(3), 675-689.
- Boysen, N., Briskorn, D., & Meisel, F. (2017). A generalized classification scheme for crane scheduling with interference. *European Journal of Operational Research*, 258(1), 343-357.
- Carlo, H. J., Vis, I. F., & Roodbergen, K. J. (2015). Seaside operations in container terminals: literature overview, trends, and research directions. *Flexible Services and Manufacturing Journal*, 27(2-3), 224-262.
- Chen, J. H., Lee, D. H., & Goh, M. (2014). An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem. *European Journal of Operational Research*, 232(1), 198-208.
- Chung, S. H., & Choy, K. L. (2012). A modified genetic algorithm for quay crane scheduling operations. *Expert Systems with Applications*, 39(4), 4213-4221.
- Daganzo, C. F. (1989). The crane scheduling problem. *Transportation Research Part B: Methodological*, 23(3), 159-175.
- Dongarra, J. J. (2014). *Performance of various computers using standard linear equations software*. Electrical Engineering and Computer Science Department University of Tennessee, Knoxville.
- Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (Vol. 2, pp. 1470-1477). IEEE.
- El Khoukhi, F., Boukachour, J., & Alaoui, A. E. H. (2017). The “Dual-Ants Colony”: A novel hybrid approach for the flexible job shop scheduling problem with preventive maintenance. *Computers & Industrial Engineering*, 106, 236-255.

- Expósito-Izquierdo, C., González-Velarde, J. L., Melián-Batista, B., & Moreno-Vega, J. M. (2013). Hybrid estimation of distribution algorithm for the quay crane scheduling problem. *Applied Soft Computing*, 13(10), 4063-4076.
- Figueira, G., & Almada-Lobo, B. (2014). Hybrid simulation–optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory*, 46, 118-134.
- Gabrel, V., Murat, C., & Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3), 471-483.
- Hansen, P., Mladenović, N., & Pérez, J. A. M. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1), 367-407.
- He, J., Zhang, W., Huang, Y., & Yan, W. (2013). A simulation optimization method for internal trucks sharing assignment among multiple container terminals. *Advanced Engineering Informatics*, 27(4), 598-614.
- Herazo-Padilla, N., Montoya-Torres, J. R., Isaza, S. N., & Alvarado-Valencia, J. (2015). Simulation-optimization approach for the stochastic location-routing problem. *Journal of Simulation*, 9(4), 296-311.
- Hirsch, P., Palfi, A., & Gronalt, M. (2012). Solving a time constrained two-crane routing problem for material handling with an ant colony optimisation approach: an application in the roof-tile industry. *International Journal of Production Research*, 50(20), 6005-6021.
- Jung, J. Y., Blau, G., Pekny, J. F., Reklaitis, G. V., & Eversdyk, D. (2004). A simulation based optimization approach to supply chain management under demand uncertainty. *Computers & Chemical Engineering*, 28(10), 2087-2106.
- Kaveshgar, N., Huynh, N., & Rahimian, S. K. (2012). An efficient genetic algorithm for solving the quay crane scheduling problem. *Expert Systems with Applications*, 39(18), 13108-13117.
- Kelly, J. P. (2002). Simulation optimization is evolving. *INFORMS Journal on Computing*, 14(3), 223-225.
- Kim, K. H., & Park, Y. M. (2004). A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156(3), 752-768.
- Legato, P., Mazza, R. M., & Trunfio, R. (2010). Simulation-based optimization for discharge/loading operations at a maritime container terminal. *OR Spectrum*, 32(3), 543-567.
- Lee, D. H., Wang, H. Q., & Miao, L. (2008). Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 44(1), 124-135.
- Lim, A., Rodrigues, B., & Xu, Z. (2007). A m-parallel crane scheduling problem with a non-crossing constraint. *Naval Research Logistics (NRL)*, 54(2), 115-127.
- Lim, A., Rodrigues, B., Xiao, F., & Zhu, Y. (2004). Crane scheduling with spatial constraints. *Naval Research Logistics (NRL)*, 51(3), 386-406.
- Lim, S. J., Jeong, S. J., Kim, K. S., & Park, M. W. (2006). A simulation approach for production-distribution planning with consideration given to replenishment policies. *The International Journal of Advanced Manufacturing Technology*, 27(5), 593-603.
- López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3, 43-58.
- Lu, Z., Han, X., Xi, L., & Erera, A. L. (2012). A heuristic for the quay crane scheduling problem based on contiguous bay crane operations. *Computers & Operations Research*, 39(12), 2915-2928.
- Meisel, F., & Bierwirth, C. (2011). A unified approach for the evaluation of quay crane scheduling models and algorithms. *Computers & Operations Research*, 38(3), 683-693.
- Meisel, F. (2009). Seaside operations planning in container terminals. *Berlin, Heidelberg: Physica-Verlag*.
- Michael, P. (1995). *Scheduling, theory, algorithms, and systems*. Englewood Cliffs, New Jersey.
- Moccia, L., Cordeau, J. F., Gaudioso, M., & Laporte, G. (2006). A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics (NRL)*, 53(1), 45-59.

- Monaco, M. F., & Sammarra, M. (2011). Quay crane scheduling with time windows, one-way and spatial constraints. *International Journal of Shipping and Transport Logistics*, 3(4), 454-474.
- Multimethod Simulation Software and Solutions, accessed on, Nov. 22, 2015. [Online]. Available: <http://www.anylogic.com/>
- Nguyen, S., Zhang, M., Johnston, M., & Tan, K. C. (2013). Hybrid evolutionary computation methods for quay crane scheduling problems. *Computers & Operations Research*, 40(8), 2083-2093.
- Peterkofsky, R. I., & Daganzo, C. F. (1990). A branch and bound solution method for the crane scheduling problem. *Transportation Research Part B: Methodological*, 24(3), 159-172.
- Rajendran, C., & Ziegler, H. (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 155(2), 426-438.
- Rouky, N., Boukachour, J., Alaoui, A.E.H, El Khoukhi, F., & Boudebous D. (2015). Un algorithme d'optimisation par colonie de fourmis pour l'ordonnancement des grues de quai. *JD-JN- MACS*, Bourges, France.
- Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6), 971-983.
- Sammarra, M., Cordeau, J. F., Laporte, G., & Monaco, M. F. (2007). A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling*, 10(4), 327-336.
- Steenken, D., Winter, T., & Zimmermann, U. T. (2001). Stowage and transport optimization in ship planning. In Online optimization of large scale systems (pp. 731-745). *Springer Berlin Heidelberg*.
- Thiruvady, D., Ernst, A. T., & Singh, G. (2016). Parallel ant colony optimization for resource constrained job scheduling. *Annals of Operations Research*, 242(2), 355-372.
- Tongzon, J., & Heng, W. (2005). Port privatization, efficiency and competitiveness: Some empirical evidence from container ports (terminals). *Transportation Research Part A: Policy and Practice*, 39(5), 405-424.
- Tréfond, S., Billionnet, A., Elloumi, S., Djellab, H., & Guyon, O. (2017). Optimization and simulation for robust railway rolling-stock planning. *Journal of Rail Transport Planning & Management*, 7(1-2), 33-49.
- UNCTAD, WID. (2014). United nations conference on trade and development. *Review of Maritime Transport*.
- UNCTAD, WID. (2016). United nations conference on trade and development. *Review of Maritime Transport*.
- Unsal, O., & Oguz, C. (2013). Constraint programming approach to quay crane scheduling problem. *Transportation Research Part E: Logistics and Transportation Review*, 59, 108-122.
- Wang, Y., & Kim, K. H. (2011). A quay crane scheduling algorithm considering the workload of yard cranes in a container yard. *Journal of Intelligent Manufacturing*, 22(3), 459-470.
- Zhou, Y., Wang, W., Song, X., & Guo, Z. (2016). Simulation-based optimization for yard design at mega container terminal under uncertainty. *Mathematical Problems in Engineering*, Vol. 2016.
- Zhu, Y., & Lim, A. (2006). Crane scheduling with non-crossing constraint. *Journal of the Operational Research Society*, 57(12), 1464-1471.

Appendix A

Table 5

Detailed deterministic results of B&B, GRASP, TS, UDS, GA, HGA, HGP, HACO-EST and HACO-LWL algorithms

Set	Instance	Kim and Park (2004)		Sammarra et al. (2007)	Bierwirth and Meisel (2009)	Chung and Choy (2012)	Nguyen et al. (2013)		Proposed		
		LB	B&B	GRASP	TS	UDS	GA	HGA	HGP	HACO-EST	HACO-LWL
Set1	K13	453.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K14	546.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K15	513.00	0.00	0.58	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K16	312.00	2.88	2.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K17	453.00	0.66	0.66	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K18	375.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K19	543.00	1.66	1.66	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K20	399.00	20.30	20.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K21	465.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K22	537.00	34.08	34.08	0.00	0.56	0.00	0.56	0.56	0.56	0.00
Average RPD on Set1		5.96	6.02	0.00	0.06	0.00	0.06	0.06	0.06	0.06	0.00
Set2	K23	576.00	0.00	2.60	1.04	0.00	0.00	0.00	0.00	0.00	0.00
	K24	666.00	0.45	1.35	0.45	0.00	0.45	0.00	0.00	0.00	0.00
	K25	738.00	0.00	0.41	0.41	0.00	0.81	0.00	0.00	0.00	0.00
	K26	639.00	0.00	1.88	0.00	0.00	0.94	0.00	0.00	0.00	0.00
	K27	657.00	0.00	4.57	0.46	0.00	0.46	0.00	0.00	0.00	0.00
	K28	531.00	1.13	3.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K29	807.00	0.00	1.49	0.37	0.00	0.37	0.00	0.00	0.37	0.00
	K30	891.00	0.00	1.68	0.00	0.00	0.67	0.00	0.00	0.00	0.00
	K31	570.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K32	591.00	0.00	1.02	0.00	0.00	0.51	0.00	0.00	0.00	0.00
Average RPD on Set2		0.16	1.84	0.27	0.00	0.42	0.00	0.00	0.00	0.04	0.00
Set3	K33	603.00	0.00	10.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	K34	717.00	0.00	6.28	2.51	0.00	0.00	0.84	0.00	0.00	0.00
	K35	684.00	0.88	2.19	0.88	0.00	0.88	0.00	0.00	0.00	0.00
	K36	678.00	6.19	4.42	0.44	0.00	0.00	0.00	0.00	0.00	0.00
	K37	510.00	1.18	5.88	1.76	0.00	2.35	0.00	0.00	0.00	0.00
	K38	613.70	3.14	7.54	0.71	0.70	0.70	0.21	0.21	0.21	0.70
	K39	508.40	8.58	13.89	2.09	0.90	2.08	0.90	0.90	0.90	0.90
	K40	564.00	5.85	5.85	0.53	0.00	0.53	0.53	0.00	0.00	0.53
	K41	585.00	9.74	9.74	1.53	0.51	0.51	0.00	0.00	0.00	0.00
	K42	560.30	18.86	18.86	2.80	2.27	2.80	3.34	3.34	2.80	3.34
Average RPD on Set3		5.44	8.51	1.33	0.44	0.99	0.58	0.45	0.39	0.55	
Set4	K43	859.30	9.62	9.62	2.29	1.94	4.39	1.59	1.94	1.59	1.94
	K44	820.40	4.58	4.58	1.66	0.20	4.22	1.29	0.20	1.29	1.29
	K45	824.90	5.83	5.83	3.29	1.10	4.74	1.83	1.83	1.83	1.83
	K46	690.00	6.52	6.52	0.00	0.00	4.78	0.00	0.00	0.00	0.00
	K47	792.00	1.89	1.89	0.00	0.00	3.41	0.00	0.00	0.00	0.00
	K48	628.90	6.38	6.38	5.43	1.61	5.42	1.61	1.61	1.61	1.61
	K49	879.20	10.56	10.56	3.73	1.68	4.07	2.02	2.02	2.02	2.14
Average RPD on Set4		6.48	6.48	2.34	0.93	4.43	1.19	1.09	1.19	1.26	
Average on all instance		4.51	5.71	0.99	0.36	1.46	0.46	0.40	0.42	0.45	

Table 6
Detailed SO approach results using EST and LWL strategies

Set	Instance	Maximum Value			Expected Value of Cmax			CPU (s)	
		HACO-EST	HACO-LWL	Gap1	HACO-EST	HACO-LWL	Gap2	HACO-EST	HACO-LWL
Set1	K13	188.20	192.09	2.07	172.76	175.63	1.66	8.26	5.11
	K14	211.36	228.79	8.25	199.36	209.53	5.10	3.51	3.66
	K15	213.55	206.89	-3.12	198.05	206.89	4.46	3.27	2.63
	K16	136.25	142.25	4.40	119.67	125.69	5.03	6.24	5.21
	K17	190.50	191.50	0.52	173.79	176.94	1.81	3.92	3.36
	K18	151.71	161.24	6.28	143.63	148.32	3.27	3.02	3.32
	K19	233.88	231.87	-0.86	207.38	212.28	2.36	4.95	3.77
	K20	181.70	182.70	0.55	155.63	157.27	1.05	6.96	5.85
	K21	197.57	200.57	1.52	180.87	182.94	1.14	5.16	3.34
	K22	230.16	276.55	20.16	211.76	241.58	14.08	5.48	3.62
Set2	K23	243.24	249.16	2.43	226.73	232.95	2.74	7.26	7.64
	K24	274.29	278.25	1.44	260.78	266.63	2.14	6.84	6.28
	K25	316.99	351.68	10.94	286.03	282.63	-1.19	9.00	8.21
	K26	269.33	278.31	3.33	250.72	261.78	4.41	7.26	8.30
	K27	269.71	274.62	1.82	254.93	263.92	3.53	4.31	5.14
	K28	225.67	234.37	3.86	206.90	214.58	3.71	8.26	8.69
	K29	333.77	335.47	0.51	313.17	317.76	1.47	4.85	5.56
	K30	350.31	362.22	3.40	340.35	355.10	4.33	6.80	6.31
	K31	240.45	253.39	5.38	221.36	232.84	5.19	9.58	8.04
	K32	241.35	268.00	11.04	229.75	235.95	2.70	7.27	9.23
Set3	K33	259.69	262.71	1.16	237.55	248.05	4.42	18.05	20.46
	K34	298.12	315.42	5.80	285.51	299.31	4.83	16.40	21.79
	K35	290.55	312.45	7.54	269.58	284.14	5.40	15.79	22.42
	K36	282.18	349.79	23.96	267.97	284.89	6.31	14.92	20.58
	K37	220.22	235.56	6.97	201.49	218.54	8.46	18.17	22.01
	K38	250.07	259.81	3.89	237.74	248.17	4.39	12.36	19.65
	K39	213.34	231.94	8.72	202.05	219.34	8.56	16.48	26.60
	K40	240.64	240.47	-0.07	222.23	229.84	3.42	13.02	21.22
	K41	244.00	248.00	1.64	230.04	235.85	2.53	14.73	21.99
	K42	244.00	249.58	2.29	230.04	233.03	1.30	12.94	19.54
Set4	K43	371.45	380.70	2.49	346.29	369.22	6.62	25.23	38.00
	K44	349.47	382.61	9.48	327.69	363.63	10.97	24.16	42.24
	K45	351.43	359.17	2.20	334.23	338.90	1.40	22.48	42.50
	K46	295.49	319.54	8.14	275.35	296.91	7.83	22.50	36.54
	K47	333.76	398.95	19.53	320.32	345.35	7.81	21.66	40.36
	K48	276.84	289.19	4.46	255.13	272.51	6.81	17.34	33.56
	K49	358.08	402.91	12.52	351.02	377.80	7.63	21.04	35.03
Average on all instance				5.53		4.53	11.61	16.55	



© 2019 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).