Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

SIMULATION-BASED OPTIMIZATION FOR THE QUAY CRANE SCHEDULING PROBLEM

Pasquale Legato
Rina Mary Mazza

Dip. di Elettronica Informatica e Sistemistica (DEIS)
Via P. Bucci, 41C, University of Calabria
87036, Rende (CS), ITALY

Roberto Trunfio

CESIC – NEC Italia S.r.l.
Via P. Bucci, 22B, CESIC, NEC Italy
87036, Rende (CS), ITALY

ABSTRACT

Maritime terminals of pure transhipment are emerging logistic realities in long-distance containerized trade. Here, complex activities of resource allocation and scheduling should be optimized in a dynamic, non deterministic environment. The assignment of expensive quay cranes to multiple vessel-holds for container discharging and loading operations is a major problem, whose solution affects the operational performance of the whole terminal container. In OR literature, this problem is known as the quay crane scheduling problem. With the objective of minimizing the vessel's overall completion time, we first give our IP formulation and then, under the more realistic assumption that discharge-loading times are non deterministic, we focus on a simulation-based optimization approach which embodies the IP formulation. Two different simulation optimization algorithms are tailored to the problem: *simulated annealing* and *adaptive balanced explorative and exploitative search*. Preliminary numerical results are presented on real vessel data.

1 INTRODUCTION

The world container fleet amounts to about 23.2 million TEUs (twenty-foot equivalent units) and in 2006 the container throughput reached 440 million TEUs (UNCTAD 2007). Containerized trade is forecasted to grow by an average annual rate of 5.32% until the year 2025 (UNCTAD 2004). As a result of this trend, the number of maritime and inland container terminals worldwide keeps increasing. Competition has become both price driven and service driven and, therefore, the success of an individual company will depend on its ability to fulfill customer demand with high standard quality service, while keeping operations lean.

Maritime container terminals are the most important crossroads for transshipment and intermodal container transfers, based on the *spokes-hub* distribution paradigm. These facilities have different layouts and they are typically composed by heterogeneous sets of resources deployed within each port sub-area. According to Steenken, Voß, and Stahlbock (2004), the main sub-areas are *i)* the ship operation area (i.e., the *quay*), *ii)* the import/export stacking area (i.e., the *yard*) and *iii)* the truck and train operation area. Referring to the operations that occur within the quay and yard areas, the most common resources are cranes and shuttle vehicles. Quay cranes are usually of two types: rail-mounted gantry cranes (RMGCs) and rubber-tired gantry cranes (RTGCs). Shuttle vehicles are selected according to how container transfer occurs from the quay to the yard and vice versa: most European and North-American container terminals are based upon the "Direct Transfer System" (DTS), which implies the use of *straddle carriers*, special vehicles able to pick-up/set-down and transfer one or more containers per time.

Stahlbock and Voß (2008) claim that container handling (i.e., stacking and transport operations) is a key factor for a container terminal's efficiency. In this context, a complex scheduling problem that arises when multiple quay cranes are assigned to the same ship with the aim of performing discharge and loading operations is known as the *quay crane scheduling problem* (QCSP). The goal is to assign each vessel *hold* or *bay* (task) to a specific quay crane (machine), with the objective of minimizing the overall completion time (*makespan* minimization). Precedence and non-simultaneity constraints between tasks are taken into account, as well as release times on cranes in the IP formulation proposed in this paper. The solution of the QCSP has been successfully dealt with in literature by using both deterministic approaches (and solving the relaxation of the IP formulation) and metaheuristics algorithms (Daganzo 1989; Kim and Park 2004; Lim, Rodrigues, and Xu 2007; Sammarra et al. 2007).

In real life management of logistics at a maritime container terminal, the QCSP arises as a decisional step within the discharge/loading process; thus, we address the issue of using the solution of the QCSP within a simulation model of the above process. The simulation model has to evaluate the key performance measure that should be optimized. Here we show how a simulation-based op-

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

timization is a cost-effective technique in terms of results realism and quality of the solution returned.

The remainder of this paper is organized as follows. In the next section, we provide a detailed description of the logistic processes set around the discharge/loading operations in a maritime container terminal. Afterward, we propose a mathematical formulation of the QCSP. In the following section, we describe two simulation-based optimization approaches to the QCSP using the *simulated annealing* (SA) and the *adaptive balanced explorative and exploitative search* (A-BEES) frameworks. Computational results using real vessel data are provided and compared with the deterministic problem solution obtained through the CPLEX solver. Conclusions are reported in the last section.

2 PROBLEM DESCRIPTION

A simulation model of the (outer) vessel arrival-service-departure process has been recently developed for the container terminal at the port of Gioia Tauro (Canonaco et al. 2007), where attention was drawn to the channel and berth subsystems with the aim of providing suitable weekly plans for the berth allocation office. On the other hand, to further improve the efficiency of berth operations, a very important role is also played by the quay cranes and their ability to perform container discharge/loading operations. As many as 6 units of this expensive handling equipment can be deployed to serve the latest generation containerships during an operational work-cycle.

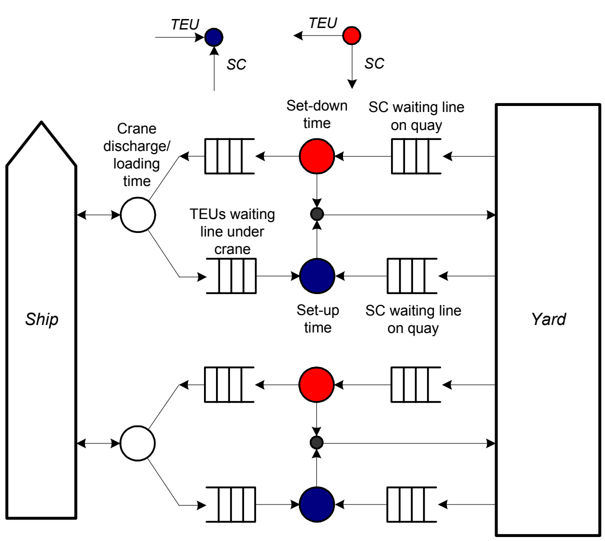

Figure 1: The discharge/loading process

A quay crane operates in the berth area by moving (on wheel or rail) in horizontal directions to reach different holds within the same vessel or on different vessels. For both discharge and loading operations, a very restricted area (e.g. a 6-slot space) for buffering a limited number of containers is naturally provided at the basis of each quay crane. When performing discharge operations, a quay crane picks-up containers from the vessel and "feeds" them to straddle carries (SCs) which provide for their transfer from the quay area to the assigned yard positions within the terminal storage area. As one may observe in Figure 1, the discharge process from the ship to the yard features a joining point (in blue) between the unloaded container and the SC sent for its pick-up and transfer to the yard. As far as loading operations are concerned, a quay crane picks-up containers delivered from the terminal yard by the SCs and places them on the ship in the assigned vessel holds. Figure 1 accounts for this process from the yard to the ship as well: in particular, the forking point (in red) represents the physical separation carried out by an SC when it first sets-down the container in the quay crane buffer area and then returns (empty) to the yard to retrieve other containers. The entire discharge/loading process was also the subject of previous research by the authors (Canonaco et al. 2008). At that time, the main focus was on the representational capabilities offered by some modeling languages and description tools used to incorporate both the low level operational policies and work rules of the above process and the specific scheduling constraints involved in the vessel hold - quay crane assignment. In that case, solution generation and exploration was performed by a "manual" simulation-based optimization procedure.

From here on, we concentrate on a well-known operational problem arising from a different number of cranes working in parallel on the same vessel at the same instant: the quay crane scheduling problem. The objective of our study is to determine the crane split or schedule or, in other words, which and in what order holds should be assigned to the single quay cranes to minimize the vessel's overall completion time, provided that:

- a minimum distance is left between quay cranes to avoid boom collision (i.e. non-simultaneity constraints);
- some holds must be operated before others (precedence constraints);
- not every crane is available immediately (release constraints).

For problem solution, in the following sections we propose both an optimization model and two simulation-based optimization approaches.

3 MATHEMATICAL MODEL

The following notations will be used for the formulation of the quay crane scheduling problem. Let T={1,…, n} be a set of time-slots, Ω={1,…, m} a set of tasks and

C={1,…, c} a set of quay cranes. Each task amounts to perform a fixed number of container moves (discharge/loading) which require a non deterministic number of time slots to be carried out. As first approximation, one may formulate an Integer Programming (IP) model by resorting to the use of the average values for the above processing times. Non-simultaneity relationships between task pairs are expressed by the set Ψ = {(i, j) | i, j $\in \Omega$, task *i* must be completed before task *j* starts or task *i* must start before task *j* is completed}, while precedence relationships between pairs of tasks are expressed by the set Φ = {(i, j) | i, j $\in \Omega$, task *i* must be completed before task *j* starts}. Let *z* be the length of each time-slot $t \in$ T and p_i the processing time of task $i \in \Omega$, therefore we define f_i as the average number of time-slots necessary to perform task *i*, where $f_i = \lceil p_i / z \rceil$. In the end, let r_c be the release time of quay crane c $\in$ C, with $1 \le r_c \le |T| - 1$ and note that, once again, this is considered as a deterministic value. M is a suitably big number.

Let's also introduce the variable θ_{it}^c (i $\in \Omega$, t $\in$ T, c $\in$ C), which is equal to 1 if and only if task *i* is performed by crane *c* starting from time-slot *t*, 0 otherwise. Thus, the decision parameter θ is d-dimensional, where d=|T|*|Ω|*|C|.

A possible, stand-alone, IP model for the QCSP is the following:

$$\min \max_{i \in \Omega} \sum_{c \in C} \sum_{t \in T} (t + f_i - 1)\theta_{it}^c \tag{1}$$

$$\sum_{c \in C} \sum_{t \in T} \theta_{it}^c = 1 \quad \forall i \in \Omega \tag{2}$$

$$\sum_{r=t}^{t+f_i-1} \sum_{\substack{j \in \Omega \\ j \neq i}} \theta_{jr}^c \le M\left(1 - \theta_{it}^c\right) \quad \forall i \in \Omega, t = 1, \ldots n - f_i + 1, c \in C \tag{3}$$

$$\sum_{t \in T} (t + f_i - 1)\theta_{it}{}^c \le n \quad \forall i \in \Omega, c \in C \tag{4}$$

$$\sum_{c \in C} \sum_{r=t}^{t+f_i-1} \theta_{jr}^c + \sum_{c \in C} \sum_{r=t}^{t+f_j-1} \theta_{ir}^c \le 1 \quad \forall (i,j) \in \Psi, t = n - \max\{f_i, f_j\} \tag{5}$$

$$\sum_{s \in T} \left(s \sum_{c \in C} \theta_{js}^c \right) - \sum_{t \in T} \left(t \sum_{c \in C} \theta_{it}^c \right) \ge f_i \quad \forall (i,j) \in \Phi \tag{6}$$

$$\sum_{i \in \Omega} \sum_{t=1}^{r_k - 1} \theta_{it}^c = 0 \quad \forall c \in C \tag{7}$$

$$\theta_{it}^c \in \{0,1\} \quad \forall i \in \Omega, t \in T, c \in C \tag{8}$$

The mathematical model is a *minmax* problem, where the objective function stands for the makespan and should be converted into a linear form whenever one should solve it by an IP commercial package such as CPLEX (ILOG 1999). Constraints (2) specify that every task can only be assigned to one crane and operations must begin at time *t*. Constraints (3) ensure that there is no overlapping between tasks assigned to the same crane. Constraints (4) guarantee that all tasks must be completed before time slot *n*. Constraints (5) ensure that tasks *i* and *j* are not processed simultaneously if, as a pair, they belong to the non-simultaneity set. Constraints (6) guarantee that task *i* will be processed before task *j* if there is a precedence relationship between them. Constraints (7) ensure that a task cannot be assigned to a crane before the crane has been released. Constraints (8) are the constraints on the decision variables.

At this point, one may easily recognize that the IP formulation focuses on the sole allocation/scheduling decisions to be taken within the more complex, dynamic discharge/loading process illustrated in Figure 1.

A practical solution to the optimization of the overall logistic process is proposed, in the following, by resorting to simulation-based optimization.

4 SIMULATION OPTIMIZATION APPROACH

Simulation-optimization (Andradottir 1998) is well known as the optimization of an expected performance measure based on outputs from stochastic simulations of any given system/process, whose dynamic behaviour is partially defined by some decision variables and constraints. Here, the expected performance measure is the expected value of the *makespan* and it should be estimated through simulation of the queuing network model in Figure 1. The formulation of the simulation optimization problem would require to replace the objective function of problem (1)-(8) with the following: $min\ E[f(\theta)]$, which also accounts for implicit additional process features and queuing phenomena when searching for the optimal vector of decision variables, θ.

In the simulation-based optimization methods proposed in the next sub-sections, solution "comparison" is based on statistics for the *makespan* which are computed on a certain number of observations. Since these observations are random variates returned from a simulation process, there are no guarantees of selecting the best design during the solution comparison, despite it being truly representative of the best system configuration. To this end, at the "comparison step" of each algorithm we decided to introduce the *indifference-zone* based *Ranking and Selection* (Goldsman et al. 2002) procedure first computed by Rinott (1978) to perform a correct selection with at least probability P^*.

4.1 The A-BEES search method

The A-BEES is a simulation-based optimization framework developed in the last decade by Prudius and Andradóttir (2004). A detailed and up-to-date description of the framework is available in (Prudius 2007).

In the A-BEES framework the search for the optimal solution through the feasible region is pursued by balancing two phases: exploration and exploitation. Exploration represents the global search for promising solutions within the entire feasible region, while exploitation embodies the local search of promising sub-regions. The A-BEES strategy adaptively alternates between (local) sampling from the neighborhood of a current solution and (global) sampling in the entire space of feasible solutions. Thus the adopted sampling technique is a key performance factor of the methodology.

Let v^* be the function value corresponding to the current optimal solution θ_n found at the iteration n and v_l^* the function value corresponding to the last time local search was performed. Besides, v_k^* is the function value corresponding to the optimal solution found at the last review.

Let Δ be the improvement in the function value between the current and preceding reviews and D the distance between the points where the corresponding function values were achieved.

Algorithm 1: Search Type Update Procedure

if $LS = true$ **then**
  if $\Delta \le \delta$ **then**
    $LS \leftarrow false$, $v_l^* \leftarrow v^*$
else if $\Delta \le \delta$ **then**
  if $v_l^* - v^* \ge \delta$ **then**
    $LS \leftarrow true$
else if $D \le d$ **then**
  $LS \leftarrow true$

As depicted in Algorithm 1, the A-BEES algorithm can switch every k_l iterations from global to local search in two ways: *i)* whenever the improvement Δ is small (less than or equal to a threshold δ), but the method finds a substantial improvement in the objective function value during the last global search stage (i.e., a promising region has been identified); *ii)* when the improvement between Δ is small, but the distance D is small (less than or equal to a measure d). In both cases, the local search flag *LS* is set to *true*. Vice versa, the algorithm switches from local to global search if no meaningful improvement has been achieved during the last k_l algorithm iterations (in this case, *LS* is set to *false*).

The nature of the mathematical model that we propose, composed by complex constraints and binary variables, forced us to use such methods to randomly generate new feasible alternatives. Therefore, we have developed two simple random procedures to generate a feasible solution for the QCSP with respect to the current search nature.

The first procedure (i.e. the *global procedure*) has been designed for searching within the entire feasible region. It attempts to find a feasible schedule, by iteratively and randomly selecting a non-assigned task and trying to assign it to an available crane (chosen randomly) at a random time-slot, with respect to the model constraints. If no feasible random assignment is found after *k* iterations, all the already assigned tasks are de-assigned and the procedure starts again (until the maximum number of attempts is reached).

The aim of the second procedure (i.e. the *local procedure*) is to identify a set of *k* neighboring alternatives from the current alternative. For us, a neighbor is a schedule that differs from the current one only by a task assignment. Thus, the maximum number of neighbor alternatives (unfeasible included) is equal to $(n \cdot c - 1)^m$; otherwise, we have a smaller number of feasible neighbors, usually some hundreds or a thousand. Therefore, the procedure for sampling a local solution randomly selects a task *i* from Ω and tries to find a new feasible assignment, i.e. changing the crane, the schedule-time or both.

The A-BEES implementation for a minimization problem is given in Algorithm 2.

Algorithm 2: A-BEES Procedure

$counter \leftarrow 0$, $n \leftarrow 0$, $LS \leftarrow false$
Generate a feasible schedule θ using the *global procedure* and evaluate $f(\theta)$
Let $v_k^*, v^*, v_l^* \leftarrow f(\theta)$ and $\theta_n \leftarrow \theta$
while *Stopping criterion is not satisfied* **do**
  if $LS = true$ **then**
    Generate a list L of max k_l feasible schedules using the *local procedure*. Extract θ from the top of L
  else
    Generate a feasible schedule θ using the *global procedure*
  Evaluate the objective function at θ
  $counter \leftarrow counter + 1$, $n \leftarrow n + 1$
  if $f(\theta) < v^*$ **then**
    $\theta_n \leftarrow \theta$, $v^* \leftarrow f(\theta)$
  if $LS = false$ **and** $counter = k_g$ **or** $LS = true$ **and** $counter = k_l$ **or** $L = \Theta$ **then**
    $\Delta \leftarrow v_k^* - v^*$, $v_k^* \leftarrow v^*$, and compute D
    Update search nature using *Algorithm 1*
    $counter \leftarrow 0$
end while
Present $\theta_n^* = \theta_n$ as the estimate of the optimal schedule

---

As requisite to understand the algorithm implementation, we describe how we compute the distance between two different alternatives θ^1 and θ^2. We consider for each task i from Ω the bi-dimensional matrix which shows if task i has been assigned to a crane at a certain time-slot, as illustrated by Table 1. We identify a task assignment from the matrix element using the following notation, $(c_a, t_a)^i$, in which c is the row index, t the column index, i the task, and a the alternative indexes.

For each couple of alternatives, we can compute the sub-distance D^i for task i as $|c_1 - c_2| + |t_1 - t_2|$. Thus, the distance between two alternatives is computed as $D = \sum_{i \in \Omega} D^i$.

Table 1: Assignment for task *i* at design *a*

Crane	*Time-slot*							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0

For this particular problem, we experienced that the best-performing values for the A-BEES parameters are $k_g = 20$, $k_l = 32$, $d = 9$ and $\delta = 0.25$.

4.2 Simulated Annealing

The original simulated annealing (SA) algorithm was introduced by Kirkpatrick, Gelatt and Vecchi (1983) by developing the similarities between combinatorial optimization problems and statistical mechanics. In the field of metal sciences, the annealing process is used to eliminate the reticular defects from crystals by heating and then gradually cooling the metal.

In our case, a reticular defect could be a vessel hold - quay crane assignment schedule that generates a high *makespan. Thus, the annealing process is aimed to generate feasible schedules, explore them in a more or less restricted amount and, finally, stop at a satisfactory solution*. To avoid getting caught in local minima, during the exploration process a transition to a worse feasible solution (higher-energy state) can occur with probability $p = e^{\Delta / T}$, where Δ is the difference between the values of the objective function (energy) of the current solution (state) θ and the candidate solution θ_t and T is the process temperature. A prefixed value of T determines the stop of the entire process and it usually decreases according to a so-called *cooling schema*. Unfortunately, in the literature there is no algorithm that can determine "correct" values for the initial temperature and cooling schema, but, as suggested by empirical knowledge, simple cooling schemas seem to work well (Ingber 1993). More recently, Alrefaei and Andradóttir (1999) have proposed a modification of the algorithm based on the use of a constant, rather than decreasing temperature. They prove that two different approaches are both guaranteed to converge almost surely to the set of optimal global solutions.

In the following, some pseudo-code is given for the original SA algorithm for a minimization problem.

Algorithm 3: Simulated Annealing

$\theta \leftarrow$ *initial state*
for *time = 1* **to** *time-budget* **do**
  $T \leftarrow$ *cooling-schema[time]*
  if $T = 0$ **then**
    Present *current solution* as the estimate of the optimal schedule and **stop**
  Generate a random neighbor θ_t of the current solution θ by performing a *move.*
  $\Delta = f(\theta) - f(\theta_t)$
  if $\Delta > 0$ **then**
    $\theta \leftarrow \theta_t$
  else
    $\theta \leftarrow \theta_t$ *(with probability* $p = e^{\Delta/T}$*)*
end for

When customizing the SA algorithm for the QCSP, some choices need to be made.

To begin with, choosing the proper cooling schema has great impact on reaching a global minimum. In particular, it affects the number of hold-quay crane assignment schedules (solutions) that will be evaluated by running the SA algorithm. To this end, the so-called simple mathematical cooling schema $T_{i+1} = \alpha \cdot T_i$ has been tested, and the best results were returned for an initial temperature $T_0 = 100$ and a decreasing rate $\alpha = 0.995$.

The "move" definition for neighborhood generation is very context-sensitive. For the QCSP, with respect to (eventual) release, precedence and non-simultaneity constraints that determine the feasibility (or lack thereof) of a container discharge/loading schedule, some examples of moves are:

- move hold l assigned to crane i from position r to position s ($r \neq s$) within the same crane i;
- move hold l from crane i to crane j ($i \neq j$);
- swap the positions of holds l and k ($l \neq k$) on crane i;
- swap the positions of holds l and k ($l \neq k$), originally assigned to cranes i and j ($i \neq j$), respectively.

In our current implementation the second option for move definition has been adopted.

As far as the stopping criteria is concerned, QCSP designers can chose among the following possibilities:

- stop when the algorithm has reached a fixed number of iterations n or an upper bound on the available time-budget;
- stop when the current solution has not been updated in the last m iterations;
- stop when the cooling schema has reached a *lower bound* on the temperature.

For this setting we have chosen the lower bound temperature, $T = 10^{-5}$.

5 NUMERICAL RESULTS

Numerical experiments discussed in this section use a simplified simulation model referred to the queuing network in Figure 1. Specifically, we have short-circuited both the "SC waiting line on quay" and the "TEUs waiting line under crane" with the purpose of isolating and highlighting the random effects of process discharge/loading times upon the schedules and, therefore, on the makespan.

The object of the analyses reported in the following is twofold. On one hand, experiments on the QCSP mean to investigate and compare the performance of the SA and A-BEES algorithms when system dynamics are affected by one major source of uncertainty: the discharge/loading service times operated by the quay cranes (measured in container moves per hour). The results returned are also examined in relation to the optimal value found by the commercial LP software CPLEX for the optimization model proposed in section 3, which provides a lower bound on the value of the *makespan* when data is deterministic. On the other hand, the same tests intend to show how simulation-based optimization algorithms are often the only practical solution method available when dealing with difficult-to-solve combinatorial problem instances, embedded in realistic, dynamic environments characterized by several elements of randomness.

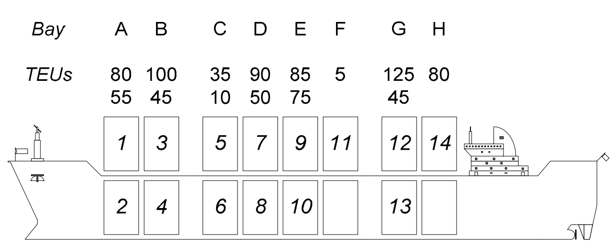

Figure 2: Map with discharge/loading info per vessel bay

This matter is even more evident as soon as one considers the real medium-size vessel illustrated in Figure 2 (courtesy of the terminal container in Gioia Tauro) for which a limited number of holds $n = 14$ must be operated by a small number of cranes $m = 3$. Although the state space of this particular problem is finite, the number of states is very large and equal to the number of unordered partitions of the n holds among the m cranes. By following (Liu 1968), as many as $1.04614 \cdot 10^{13}$ possible combinations may occur. Therefore, the exploration of every alternative schedule could go beyond practical possibilities. In our case, the number of feasible schedules that can be generated and, thus, evaluated is smaller due to the precedence and non-simultaneity constraints summarized in Table 2.

Numerical experiments are carried out on three different scenarios according to which the quay crane discharge/loading times can either be deterministic or follow an exponential or hyper-exponential distribution law. We focus on these particular laws because of their aptitude to represent a growing process variance related to the discharge/loading times.

Table 2: Problem constraints

Constraints	*Task pairs*		
precedence	(1,2)	(3,4)	(5,6)
	(7,8)	(9,10)	(12,13)
non-simultaneity	(1,3)	(5,7)	(7,9)
	(12,14)	(2,4)	(6,8)
	(8,10)	(1,4)	(2,3)
	(5,8)	(6,7)	(7,10)
	(8,9)	(10,11)	(13,14)

While the specific settings for the simulation-based optimization procedures have already been reported in paragraphs 4.1 and 4.2, here we list the common parameters of the indifference-zone based Ranking and Selection procedure: the initial number of simulation runs, $n_0 = 10$, the confidence level, $1-\alpha$ with $\alpha = 0.1$ and the indifference zone, $\delta = 0.25\text{h}$ on the makespan value. In addition, we specify both the quay crane discharge/loading rate (i.e. 28 container moves per hour) and the initial vessel hold – quay crane assignment schedule for the QCSP which is selected randomly.

In general, once parameters are set, the estimates of the objective function produced by the two algorithms converge to the same value, as the number of iterations grows. Due to its particular global-local search paradigm, the A-BEES algorithm begins convergence at an earlier stage. This feature can be regulated by conferring different weights to the explorative and exploitative stages. If only one iteration is set for the global search during the explorative stage, then the algorithm behavior will resemble the SA conduct. In the long run, the SA algorithm slightly outperforms the former procedure in terms of average execution time and quality of the makespan estimate. This is due to the algorithm's specific capability of jumping out of local minima by accepting candidate solutions that are worse than the current solution.

As one may observe in Figure 3, under deterministic quay crane service times, the average makespan values determined by the SA and A-BEES algorithms converge to the lower bound of *10.536 hours* returned by CPLEX for the IP formulation (1)-(8). Despite that an exhaustive coverage of all the possible combinations in the quay crane scheduling problem is not performed by the above algorithms, nor is any sort of control running on which part of the feasible set is being explored, the schedules returned as final output (in a large number of numerical tests carried out within these experiments) are already situated within the indifference-zone of the optimal solution (i.e. 15 minutes) after 2000 iterations. Results are provided in just a few seconds, while CPLEX returns the optimal solution after several minutes ($\cong 25$).

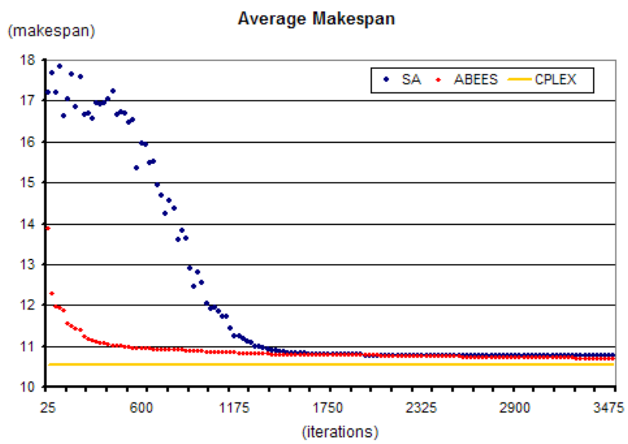

Figure 3: Makespan for deterministic service times

When quay crane service times are non-deterministic the optimal solution of the IP formulation is no longer truly representative of the discharge/loading operations since the above mathematical model does not account for uncertainty. This becomes more pronounced as the process variance increases due to greater randomness in the quay crane operational cycle (e.g. delays, blockages, failures). As shown in Figure 4, after more or less 2500 iterations, the SA and A-BEES algorithms return a makespan value of 10.7 and 10.9 hours, respectively. Both values are still close to the optimal value previously returned by CPLEX (deterministic case) because the randomness introduced by the exponential law does not produce significant effects on the non-simultaneity and precedence constraints. It is worth observing that algorithm performances do not deteriorate when dealing with exponential service times (see Figure 5).

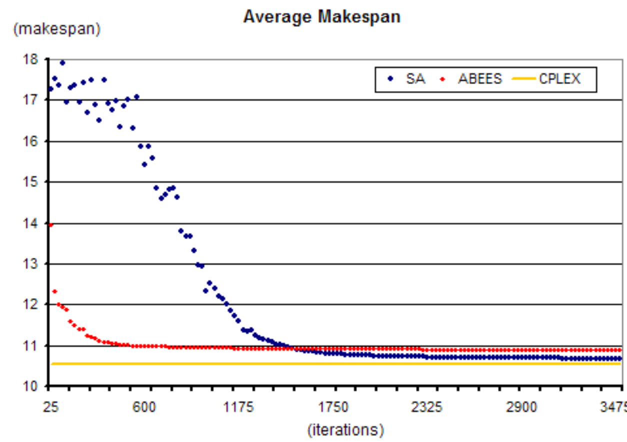

Figure 4: Makespan for exponential service times

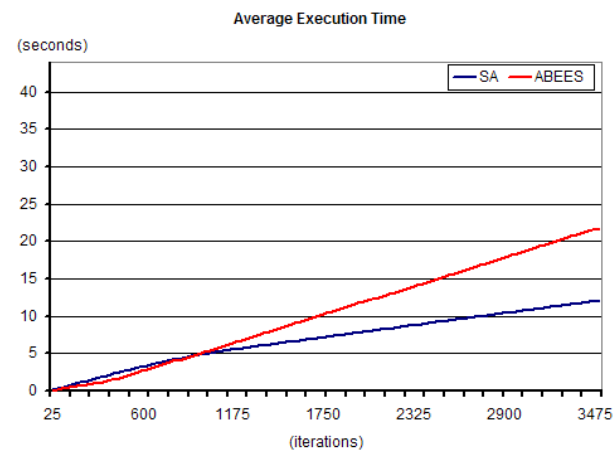

Figure 5: Algorithm performance for exponential service times

Conclusions differ a great deal when in the last scenario we chose to represent the discharge/loading operations with a hyper-exponential distribution (according to which quay crane service occurs with probability 0.95 at a rate of 28 container moves per hour and with probability 0.05 at a rate of 2 container moves per hour). As mentioned previously, a similar set-up is particularly suitable for modeling quay crane stoppage events during operations. Figure 7 shows how the SA and A-BEES achieve average makespan estimates which both depart from the value of the objective function of the solution found with CPLEX by more than 80%.

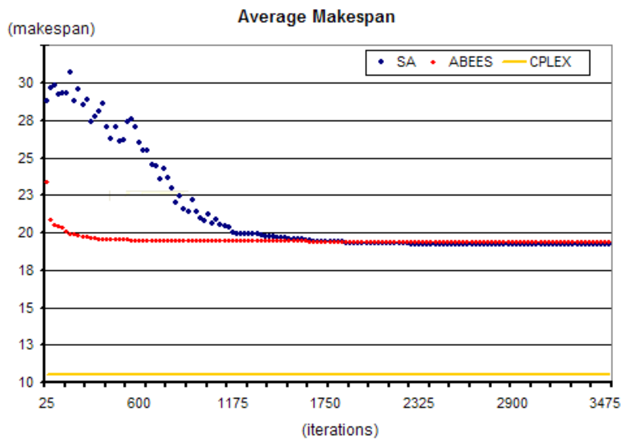

Figure 6: Makespan for hyper-exponential service times

Thus, as the uncertainty of the logistic process grows, the simulation-based optimization procedures become the only suitable solution for representing system dynamics.

6 CONCLUSIONS

We have presented two OR models for the optimal management of a core logistic process at a maritime container terminal, as they were inspired by the authors experience at the port of Gioia Tauro in Italy. The first model was a queuing network aimed to capture the key features of the logistic process at hand, viewed as a dynamic, non deterministic process; the second model was an integer programming model to be used for supporting allocation-scheduling decisions regarding quay cranes and vessel holds to be discharged and/or loaded. Both OR models have been successfully integrated in a simulation-based optimization procedure developed around metaheuristics. The A-BEES metaheuristic seems particularly promising in quickly providing cost effective solutions to the practical problem of determining the (sub)optimal assignment/schedule of quay cranes to vessel holds. Currently, we are including more operational details in the simulation model to provide a finer representation of the discharge/loading process.

REFERENCES

Andradottir, S. 1998. Simulation Optimization. In *Handbook of Simulation*. ed. J. Banks, John Wiley and Sons, Inc. New York.

Alrefaei, M. H., and S. Andradóttir. 1999. A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science* 45(5): 748-764.

Canonaco, P., P. Legato, and R.M. Mazza. 2007. An Integrated Simulation Model for Channel Contention and Berth Management at a Maritime Container Terminal. In *Proceedings of the 21st European Conference on Modelling and Simulation*, ed. I. Zelinka, Z. Oplatková, and A. Orsoni, 353-362. Prague, Czech Republic.

Canonaco, P., P. Legato, R.M. Mazza, and R. Musmanno. 2008. A queuing network model for the management of berth crane operations. *Computers and Operations Research* 35:2432–2446.

Daganzo, C. F. 1989. The crane scheduling problem. *Transport Research, Part B*. 23(3): 159–175.

Goldsman, D., S.-H. Kim, W. S. Marshall, and B. L. Nelson. 2002. Ranking and selection for steady-state simulation: procedures and perspectives. *INFORMS Journal on Computing* 14 (1): 2-19.

ILOG. 1999. CPLEX 6.5. Mountain View, California, USA.

Ingber, L. 1993. Simulated annealing: Practice versus theory. *Journal of Mathematical and Computational Modeling* 18(11):29–57.

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

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science*, *New Series* 220(4598): 671-680.

Lim, A.., B. Rodrigues, and Z. Xu. 2007. A m-parallel crane scheduling problem with a non-crossing constraint. *Naval Research Logistics* 54(2): 115–235.

Liu, C. L. 1968. *Introduction to Combinatorial Mathematics*. McGraw-Hill, New York.

Prudius, A. A., and S. Andradóttir. 2004. Simulation optimization using balanced explorative and exploitative search. In *Proceedings of the 2004 Winter Simulation Conference*, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 545–549. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Prudius, A. A. 2007. *Adaptive random search methods for simulation optimization*. Ph.D. thesis, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA. Available via <http://etd.gatech.edu/theses/available/etd-06252007-161154> [accessed April 10, 2008].

Sammarra, M., J.-F. Cordeau, H. Laporte, and M. F. Monaco. 2007. A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling* 10: 327–336.

Stahlbock, R. and S. Voß. 2008. Operations research at container terminals: a literature update. *OR Spectrum* 30: 1–52.

Steenken, D., S. Voß, and R. Stahlbock. 2004. Container terminal operation and operations research – a classification and literature review. *OR Spectrum* 26: 3–49.

UNCTAD 2004. Transport Newsletter No. 24 UNCTAD/WEB/SDTE/TLB/2004/2. United Nations Conference on Trade and Development, Transport Section, Trade Logistics Branch (United Nations, New York and Geneva) Available via <http://www.UNCTAD.org> [accessed September 28, 2007].

UNCTAD 2007. United Nations Conference on Trade and Development. Review of Maritime Transport 2007. (United Nations, New York and Geneva).

AUTHOR BIOGRAPHIES

PASQUALE LEGATO is an Assistant Professor of Operations Research at the Faculty of Engineering (University of Calabria), where he teaches courses on simulation for system performance evaluation. He has published on queuing network models for job shop and logistic systems, as well as on integer programming models. He has been involved in several national and international applied research projects and is serving as reviewer for some international journals. Current research activities focus on the development and analysis of queuing network models for logistic systems, discrete-event simulation and the integration of simulation output analysis techniques with combinatorial optimization algorithms for real life applications in Transportation and Logistics. His home-page is <http://www.deis.unical.it/legato>.

RINA MARY MAZZA was born in Leyden Township, Illinois (USA) and went to the University of Calabria, Rende (Italy), where she received her Laurea degree in Management Engineering in 1997. She is currently pursuing a Ph.D. degree in Operations Research at the same university. She has a seven-year working experience on Knowledge Management and Quality Assurance in Research Centers. She has also been a consultant for operations modeling and simulation in terminal containers. Her current research interests include discrete-event simulation and optimum-seeking by simulation in complex logistic systems. Her e-mail address is: <rmazza@deis.unical.it>.

ROBERTO TRUNFIO received his Laurea degree (*cum laude*) in Management Engineering from the University of Calabria, Rende, Italy, in 2005, and is currently pursuing a Ph.D. degree in Operations Research at the same university. Moreover, he is responsible for R&D activities in logistics and numerical simulation at NEC Italia's Center for High-Performance Computing and Computational Engineering (CESIC). His research activity lies in discrete-event simulation and simulation-based optimization (with emphasis in container terminal logistics), as well as parallel methodologies for simulation output analysis and optimization via simulation approaches. His home-page is <http://www.trunfio.it>.