

Modeling and Optimization of Decision-Making Process During Loading and Unloading Operations at Container Port

Mohamed Rida

Received: 8 August 2013 / Accepted: 31 January 2014 / Published online: 31 August 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract The subject of this paper is the process of unloading and loading ships at maritime container terminals. Since decision problems in container terminals are dynamic in nature and must be re-evaluated over time based on the state of some crucial underlying factors (such as ships, cranes, or storage yard characteristics), the problem is formulated as a Markov decision process (MDP). The objective is to provide the optimal sequence of decisions that are to be undertaken at each time during the period of unloading and loading operations in order to minimize the total waiting time of wharf cranes and vehicles allocated to serve a containership. A simulation model is also developed in this paper to test and evaluate the optimal policy provided by the MDP.

Keywords Container port · Markov decision process (MDP) · Simulation · Optimization

الخلاصة

يتضمن موضوع هذه الورقة العلمية عملية تفريغ السفن وتحميلها في محطات الحاويات البحرية. ولأن مسائل اتخاذ القرار في محطات الحاويات ديناميكية في طبيعتها، ويجب إعادة تقييمها مع مرور الوقت استناداً إلى حالة بعض العوامل الأساسية الحاسمة (مثل السفن والرافعات وخصائص ساحة التخزين) فقد تم تكوين المسألة بوصفها عملية اتخاذ قرار ماركوف. والهدف هو توفير التسلسل الأمثل من القرارات التي سيتم اتخاذها في كل مرة خلال فترة عمليات التفريغ والتحميل من أجل تقليل وقت الانتظار الكلي لرافعات الرصيف والمركبات المخصصة لخدمة سفن الحاويات. وتم أيضاً في هذه الورقة العلمية تطوير نموذج محاكاة لاختبار السياسات المثلى وتقييمها التي توفرها عملية اتخاذ قرار ماركوف.

1 Introduction

The maritime container terminal is a place where cargo containers are transferred between different transport vehicles, for onward transportation. Every marine container terminal performs four basic functions: receiving, storage, staging, and loading for both import and export containers [1]. Containers arrive to terminal by multiple means of transport and are stored in the terminal area. Then, containers leave the terminal by the same means to reach their final destinations. The maritime container terminal provides the interface between railroads, ocean-going ships, and over the road trucks, and represents the critical node in the transport network. The wharf crane is the most critical element in container terminals. Thus, managers must make decisions, regarding labor and equipment assignments that directly affect wharf crane productivity. The problem for the container terminal is to achieve different goals in an uncertain and complex environment characterized by container arrivals using different transportation modes (trucks, trains, and ships) and decisions to be taken during each stage. Loading/unloading operations, resources allocation, and storage yard management are the main criteria to be optimized. These operations are to be achieved in sequence or parallel manner. It may also have to take into account some global environment variables (wind, visibility, etc.), which can disturb the entire system operations. So, the final decision can only be made by a human decision maker. To provide adequate policies to manage the container terminal is not a simple task, and it is unlikely that a single model capable of generating an optimal solution and dealing with the complexity of this problem exists. However, it is possible to provide the decision maker with optimal policies to solve some problems such as resource allocation or loading/unloading operations.

M. Rida (✉)

Mathematics and Computer Sciences Department, Faculty of Sciences
Ain Chock, University Hassan 2 Casablanca, Km 8 Route d'El Jadida,
B.P 5366, Maarif Casablanca 20100, Morocco
e-mail: mhd.rida@gmail.com



This paper presents an interesting application of Markov decision process (MDP) to optimize the loading and unloading operations in the container terminal context with the objective to minimize the total waiting time of wharf cranes and vehicles. To unload a ship, the wharf crane picks up containers from the ship and puts them on shuttle trucks that move them to the storage yard in the terminal. In loading procedure, the wharf crane unloads a container from a shuttle truck and puts it on the ship. In the formulated MDP, we consider two queues of shuttle trucks: one under wharf crane and the other in the storage yard (near the yard crane). The states in this model correspond to the numbers of shuttle trucks in the two queues and the actions concerns shuttle trucks and yard cranes allocation to serve the wharf crane at a specific time. The cost function depends on the number of waiting shuttle trucks under each crane.

Our last paper [2] presents an MDP for the same problem where the formulation considers only quay crane operations and the study is limited to provide a policy that affects only its productivity. The established MDP ignores storage and movement operations in the terminal performed by yard cranes and vehicles, and it does not take into account that the shuttle trucks form a closed loop between the quay and the storage yards. Instead, each quay crane is considered as single service facility, and the shuttle trucks are arriving according to Poisson distribution. In this paper, the presented model is an extension to the previous one formulated in [2] where yard cranes and vehicles are also considered in the MDP. The objective is to build the model more in order to make it more realistic. Furthermore, the optimal policy will involve all critical elements that affect the terminal container productivity.

Our basic idea in this paper is that: Since exponential and Poisson distributions may be accepted to represent crane service time and truck arrival time, we can establish an MDP based on this assumption in order to obtain an alternative policy that can be compared to the usual practices in container terminals. To test the validity of the exponential/Poisson assumption, we performed, in Sect. 3, a time–motion and data analysis studies for inter-arrival and service times of shuttle trucks and cranes. Kolmogorov–Smirnov tests were used to determine which theoretical distributions can or cannot be used to describe individual samples of the time–motion study. The exponential was the distribution of service time considered in the testing procedure. This distribution was appropriate for the majority of the sample tested for crane service times and shuttle truck inter-arrivals. On the other hand, the arrival process appears to be properly represented by the Poisson distribution.

Several distributions are, of course, valid to represent crane service time and truck inter-arrival time in the terminal. In the literature, exponential distribution is adopted in [3,4], and other authors propose distribution functions such

as Uniform, Gamma, Erlang, Weibull. In [5], the uniform distribution is proposed to represent wharf crane service time and a triangular distribution for yard gantry crane. [3,4] suggest a triangular distribution function for yard gantry crane operation times. Other works [6,7] suggest using Erlang random variables, whereas [8] propose normal random variables for crane types (quay, yard). Interesting works presented in [9,10] where several distribution functions (normal truncated, Lognormal, gamma, Weibull, exponential, beta) for each handling activity and for different container type have been tested. Estimation results show that only Normal (truncated), Gamma and Weibull random variables were statistically significant and, in particular, that the Gamma random variable produced the best results.

Because existing non-commercial simulator of container terminals is quite limited and cannot simulate a specific scenario like the obtained policy in this paper, a simulation model based on discrete-event-simulation techniques and object oriented approach is presented in Sect. 6. Section 7 is dedicated to the simulation results presenting the comparison between the obtained policy and the current practices in the container terminal. According to congestion criteria, the resolved MDP provides a more efficient policy than the current practices (without policy). It consists to manage dynamically the allocated equipment in the terminal. For example, a shuttle truck allocated to serve a particular wharf crane can start working immediately with other crane when the speed of this wharf crane decreases. Furthermore, this real-time strategy allows allocating exactly the needed number of equipment in storage yards and berth depending on the flow of containers between ships and quay.

2 Literature Review

Many research works in engineering, as well as in computer science, have approached the problem of container terminal management in different ways. Existing literatures report several approaches to managing a container terminal. Most proposed approaches are based on stochastic optimization model [11–13]. Such approaches schematize container terminal activities through single-queue models or through a network of queues. Although the approaches based on optimization models allow a more elegant and compact formulation of the problem [14–17], discrete event simulation (DES) models help to achieve several aims: overcome mathematical limitations of optimization approaches, support and make computer-generated strategies/policies more understandable, and support decision makers in daily decision processes through a what if approach [13]. In the last years, several simulation tools have been developed. Some of those tools offer libraries, graphical interfaces, and different facilities for the user. In our previous work [11], we have

proposed an architecture for container terminal simulation, which is based on object-oriented paradigm and distributed discrete event simulation approach.

Other approaches are based on simulation models. These approaches schematize container terminal activities through single-queue models or through a network of queues, and the main contributions seek to maximize overall terminal efficiency or the efficiency of a specific sub-area or activity inside the terminal. In [3], the authors propose a simulation model for analyzing the performance of a real container terminal in Korea. The simulation model is developed using an object-oriented approach, and using SIMPLE ++, object-oriented simulation software. In [18], the authors present a solution to the problem of resources allocation and scheduling of loading and unloading operations in a container terminal. The solution of the resource allocation problem is based on a network design formulation that assumes that the loading and unloading processes can be modeled as a network of container flows between the ships and the terminal yard for all the work shifts. In [19], it is proposed an approach for aiding the management of a container transshipment seaport to decide on the balance between the percentage of containers to undergo security inspection and the concomitant departure delays of out-bound vessels and port costs as measured by the number of container moves. Authors use a modified A* (A-star) algorithm for problem modeling and for understanding the relation between the percentage of containers to be inspected and departure delays. However, the paper is a preliminary study; the formulation proposed is quite simple, and it does not accurately recreate a real container terminal scenario. In [13], it is used Witness software to simulate Kwai Chung container terminals. The model is used to predict the actual container terminal operations. The simulation is also proposed in [20] to investigate the positive impact of ship-to-rail direct loading on the capacity of a container terminal and to identify the congested area of the terminal. In our previous paper [11], we propose a simulation model for loading/unloading operations management. The software is based on distributed discrete event simulation approach and object-oriented paradigm. We used Java language to implement the simulator. In [20], the simulation is used as support tool for investment decision (reduction of the queue of incoming vessels). Still on simulation as decision support tool, authors in [21] propose a simulation model for the design and evaluation of multi-terminal systems for container handling. In [21], the authors present a mixed-integer programming model, which considers various constraints related to the integrated operations between different types of handling equipment. They propose a heuristic method, called multi-layer genetic algorithm to obtain the near-optimal solution of the integrated scheduling problem. The algorithm is tested using simulation. In [22], it is studied how to better utilize gate appointment system from terminal operators perspective by regulat-

ing the number of trucks that can enter the terminal. In [17], the authors propose methods for optimizing the block size, by considering the throughput requirements of yard cranes and the block storage requirements. They examine two types of bloc layouts: Blocks for which transfer points are located beside each bay and blocks for which transfer points are located at both ends. They determined that the optimal number of bays in blocks with a transfer point at each side of a bay was larger than that in blocks with transfer points at the ends, and that the optimal number of rows in blocks with a transfer point at each side of a bay was smaller than that in blocks with transfer points at the ends. In [23], it is proposed tow strategies for reducing container re-handles during the drayage truck retrieval process. These strategies are designed to be used real time, allowing for information updates during the retrieval period. The simulation is exploited to evaluate the use of truck arrival information to reduce container re-handles during the import container retrieval process by improving terminal operations. In [24], it is proposed a simulation model combined with statistical and analytical models for container terminal that takes into account the containers inspection activities. The paper presents experiment results that illustrate the impact of the container inspection process on the container.

In [25], the Visual SLAM language for discrete-event simulation is using to implement the developed simulation model for container terminal logistic activities. The model is based on queuing network approach related to arrivals, berthing, and departure processes of ships at the container terminal. A step of results validation is also performed. An application of simulation for the management of the Malaysian Kelang Container Terminal is discussed in [26]. The paper presents a model to improve the logistics processes at the port. The simulation is used in [4] to examine how productivity could be improved with the dynamic planning system for yard tractors utilizing real-time location systems technology in container terminals. As mentioned in [10], in most contributions dealing with container terminal loading/unloading operations there is an heterogeneity concerning the level of detail considered for activities involved and how such activities are aggregated in a single macro-activity. Furthermore, the most followed methodologies are based on stochastic approach. A more detailed state of the art is proposed in [10], where a classification of the published works according to adopted approach, regarding to each equipment, is presented. With regard to calibration and validation, we have published a paper [27], where a simulation model based on object-oriented approach and discrete event simulation techniques accompanied with its calibration and validation phase is presented. We have published other works in the field of container terminal simulation [28,29]. Other recent and interesting works dealing with DES can be found in [30,31].



Several works applying MDPs in the field of the transport are published. In [32], the authors present Markov decision process, applied for modeling and analysis of a public urban transportation system operation process. They assumed that the states of the discussed stochastic process correspond to operation states of the technical object (vehicle) and the decisions (called alternatives in the paper) can represent given modes of operation, events, transportation routes, maintenance, repair, etc., which can be assigned to the state of the modeled process. A simplified computing model using Gamma distribution for buses has been presented. Some simulation experiments are also presented. The authors discuss in [33] a discrete-time MDP approach for shipment consolidation in the logistics strategy. The objective is to determine when to realize consolidation loads of a consolidation program. The minimization criteria are the cost per unit time and cost per hundredweight per unit time. In [34], the Markov decision process is used to produce an optimal maintenance policy answering the following question: Which maintenance action should be chosen when the road segment has reached a certain age and condition? The results of the optimal policy are compared to another policy found using the Equivalent Annual Cost Method (EAC) used by the Road and Hydraulic Engineering Division. A Markov decision process for traffic signal system is formulated in [35]. The objective is to find an optimal control strategy for a signalized traffic intersection that reduces congestion. Statistical analysis of simulation results with different arrival rates is presented to show the effectiveness of this approach. [36] presents an approach based on MDP to determine the optimal maintenance strategy for wind turbines. An example is presented to illustrate the implementation of proposed method. A semi-Markov decision process (SMDP) is utilized in [37] to determine whether maintenance should be performed in each deterioration state and, if so, what type of maintenance to perform for repairable power equipment. In [38], road deterioration is modeled as a semi-Markov process in which the state transition has the Markov property and the holding time in each state is assumed to follow a discrete Weibull distribution. The optimal maintenance policies obtained through linear programming consist to minimize the life-cycle cost of a road network.

3 Data Acquisition and Analysis

In this section, we perform a time motion study to obtain the arrival and service time distributions at both quay and yard crane. Our goal is to determine the validity of Poisson/exponential distribution assumptions for shuttle trucks arrivals and crane service time. The objective of this section is to determine whether these assumptions are appropriate.

If they are not, it will be necessary what distribution can be used to accurately describe the system.

3.1 Data Collection

The collection of inter-arrival and service time is conceptually simple: The service time is the difference between service completions of succeeding vehicles. The assumption is that a vehicle in queue begins service immediately after the preceding vehicle completes service. To track the desired information, we record the time that each vehicle enters the queue or the service stage. Vehicle identification is accomplished by recording the truck or chassis number of each vehicle that appeared on both sides of chassis. The data used in this paper are collected at the container terminal of Casablanca in Morocco.

The figure 1 shows the primary data collection site near the quay crane and the secondary data location site in the storage yard. The motion of wheels was used as the basis of event occurrences, which is described in table 1. The visits to the container terminal of Casablanca in Morocco resulted of multiple data files. These files are transformed to obtain two groups of data: service times and inter-arrival times.

3.2 The Kolmogorov–Smirnov (K–S) Test

The objective of this section is to determine the distributions of the service and inter-arrival times. These analyzes are critical in correctly specifying the theoretical queuing models that are used in studying container terminal operations. The K-S test is a nonparametric test for the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample KS test), or to compare two samples (two-sample K-S test). The Kolmogorov–Smirnov statistic quantifies a distance between the empir-

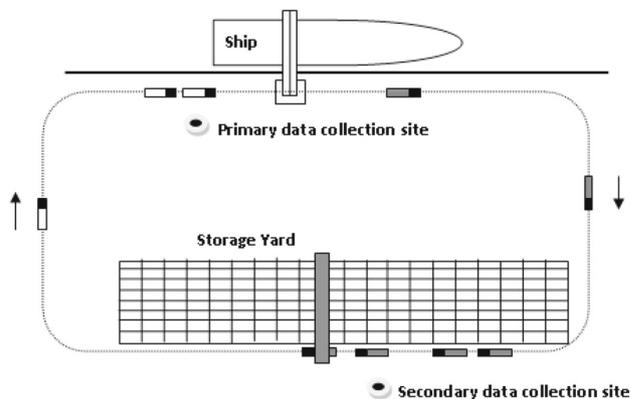


Fig. 1 Primary and secondary data collection sites

Table 1 Description of events used during data collection step

Event Code	Event description
1	Vehicle enters quay queue (wheels of vehicle stop rotating upon arrival in queue)
2	Vehicle completes move-up procedure. Wheels of vehicle stop rotating upon arrival at the service position beneath the quay crane
3	Vehicle departs service (wheels of vehicle rotate beginning the trip from the quay crane to the storage yard)
4	Beginning of quay crane idle period
6	End of quay crane idle period
7	Vehicle enters quay queue (wheels of vehicle stop rotating upon arrival in queue)
8	Vehicle completes move-up procedure (wheels of vehicle stop rotating upon arrival at the service position beneath the quay crane)
9	Vehicle departs service (wheels of vehicle rotate beginning the trip from the quay crane to the storage yard)
10	Beginning of quay crane idle period
11	End of quay crane idle period
12	One container is moved (single move)
13	Two containers are moved (double move)

ical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples. The K-S test operates by comparing the cumulative distribution functions of the theoretical and the sample distributions. The test statistic, D , is the maximum absolute difference between the two distributions, which is expected in the following equation. The theoretical distribution is represented by $F(t)$ and the sample distribution is $G(t)$.

$$D(t) = \max|F(t) - G(t)| \tag{1}$$

with: $F(t) = 1 - e^{-\lambda t}$; and $G(t) = \frac{1}{n} \sum_{x_i \leq t} 1$

The distribution test results represent statistical test for a significance level of $\alpha = 0.05$. Note that the majority of data files that were tested allow several possible distributions. Since the model developed in this paper is based on exponential assumption, we only focus on this distribution.

Figures 2 and 3 compare the sample distribution with the theoretical distribution for service and inter-arrival times of shuttle truck.

We have performed multiple tests. All files were tested for inter-arrival time distributions show that the exponential (i.e., Poisson arrivals at wharf or yard crane) is more appropriate than other distributions (like Normal or Erlang). Only two data files (from eleven) can reject the exponential distribution as statistically similar to the sample distribution. On the other hand, exponential distribution for service time was rejected by at least five of the data files. That demonstrates that the service time distribution at wharf or yard crane is not always exponential due to several reasons. Generally, with the single and double moves (i.e., shuttles can move one or two containers at once according to their size), there is no indication that the service times at wharf or yard cranes can be predicted or modeled as one distribution.

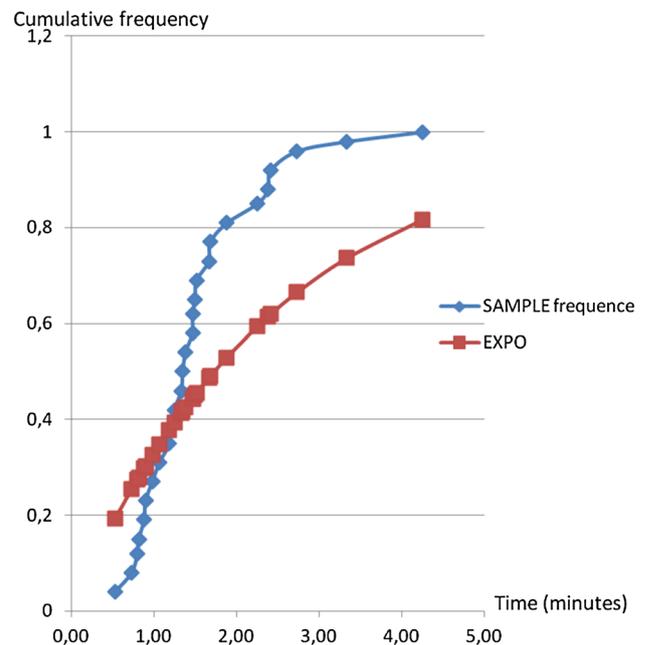


Fig. 2 Comparison of sample distribution and theoretical distribution for wharf crane service time. Sample is 26 observations

4 Formalization of the Container Terminal Problem as a Markov Decision Processes

In this section, we present a formulation of the problem loading and unloading operations as a Markov decision process (MDP) with as objective the computation of optimal decision rules that improve the productivity of each equipment used at the terminal. The main equipments considered in this case are quay cranes, yard cranes, and trucks. In this section, We will first give a review of Markov decision process (MDP), and then we characterize the process that describes the loading and unloading operations within the container terminal.

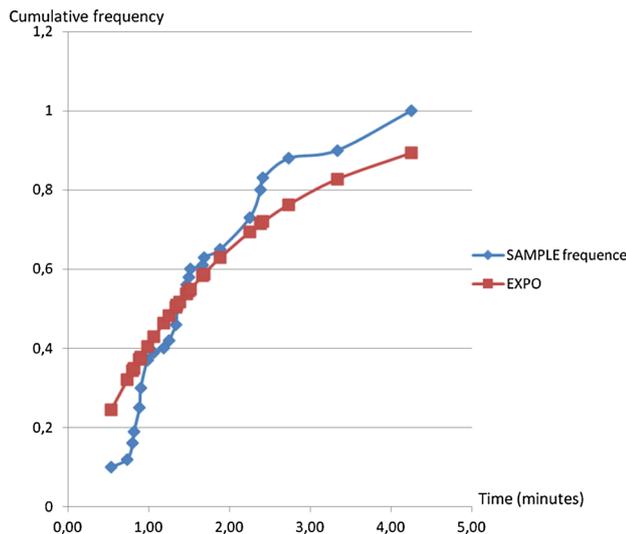


Fig. 3 Comparison of sample distribution and theoretical distribution for shuttle trucks inter-arrivals. Sample is 27 observations

4.1 Review of Markov Decision Processes (MDPs)

Markov decision processes (MDPs), named after Andrey Markov, provide a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker. At a specified point in time, the decision maker observes the state of a system and chooses an action. The chosen action produces two results: The decision maker receives an immediate reward (or incurs an immediate cost), and the system evolves probabilistically to a new state at a subsequent discrete point in time. At this subsequent point in time, the decision maker faces a similar problem. The observation made from the systems state now may be different from the previous observation. The goal is to maximize (or minimize) over an infinite horizon the average, which is the sum of successive rewards (or costs) weighted by a discount factor $0 < \gamma < 1$ that ensures the convergence of the sum, but can also be interpreted as a probability of system failure (end of mission) between two moments of the process [2,23]. A Markov decision process consists of a 4-tuple $(S; A; R; T)$:

1. The set of states S is the finite set of all possible states of the system.
2. The finite set of actions A .
3. The cost function: $C(s; a)$ depends on the state of the system and the taken action. We assume that the cost function is bounded.
4. The Markovian transition model is represented by the probability of going from a state s to another state s' by doing the action a : $P(s' | s; a) = T(s', a, s)$.

If the probabilities or costs are unknown, the problem is one of reinforcement learning [39]. The main task of reinforcement learning is finding a policy that optimizes the value of costs. This policy can be represented by a map $\pi: S \rightarrow A$

$$s \mapsto \pi(x)$$

where $\pi(s)$ is the action, which the agent (it can be a human, a robot, a part of a machine, or anything susceptible to take a decision) takes at the state s .

Most MDP algorithms are based on estimating value functions. The state-value function according to a policy π is given by:

$$J_{\pi}(s) = C(s, \pi(s)) + \sum_{s' \in S} T(s, \pi(s), s') \cdot \gamma \cdot J_{\pi}(s') \quad (2)$$

where:

1. $C(s, \pi(s))$ is the expected value of the cost C_{t+1} when the policy π is followed and such that at time t , the state is s ,
2. A parameter $\gamma (0 \leq \gamma \leq 1)$ is the discount rate, which is necessary to have a present value of the future costs. In this case, the cost is given by the infinite sum: $C_t = \sum_{k=0}^{+\infty} \gamma^k \cdot c_{t+1+k}$; where: c_t is the cost a time t .

The Bellman operator is defined as [14]:

$$T^* J_{\pi}(s) = \min_{a \in A} \left[C(s, \pi(s)) + \sum_{s' \in S} T(s, \pi(s), s') \cdot \gamma \cdot J_{\pi}(s') \right] \quad (3)$$

We denote J^* the fixed point of the Bellman operator. $J^*(s)$ is the minimum, according to actions, of $J(s)$ for a particular state s . $J^*(s)$ is the optimal state-value function.

In each case, there is one equation per state in S . Therefore, finding the policy π to get the right action to do for every state of the system is now equivalent to solving $|S|$ equations, with $|S|$ unknowns. Solving a MDP consists of looking for an efficient algorithm to solve this system. In the literature, several algorithms are proposed to calculate the optimal policy (value iteration, policy iteration, Q-learning, SARSA etc.) [39].

4.2 Markov Decision Process for Loading and Unloading Operations Within the Container Terminal

To unload a ship, the wharf crane picks up containers from the ship and puts them on shuttle trucks that move them to the storage yard in the terminal. To load a ship, the wharf crane unloads a container from a shuttle and puts it on the ship. This operation forms a closed loop that is traveled by shuttles servicing a ship (see Fig. 4). If shuttle trucks are not available underneath the crane, work ceases until a shuttle

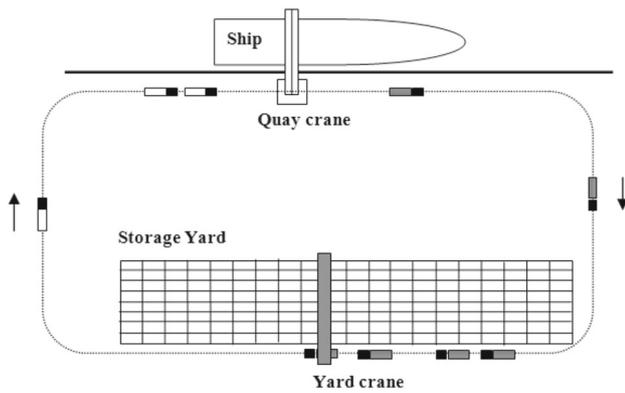


Fig. 4 Schematic of ship loading operations

truck is arrived from the yard or until another is allocated to continue service. In this formalization, we consider two queues of shuttle trucks: The first is under wharf crane (noted Q_1), and the second is in the storage yard (noted Q_2). The two service centers are: wharf and yard cranes, each with their own finite buffer, with the sizes SZ_1 and SZ_2 . We assume that both queues have arrivals according to a Poisson process with parameters λ_1 and λ_2 (respectively). All the shuttle trucks that have been served at Q_1 are routed to Q_2 , and the shuttle trucks that finished service at Q_2 return to Q_1 . We assume also that the service times are exponentially distributed with parameters μ_1 and μ_2 .

Model:

The state space S and the action set A are finite. When the action a is chosen in state s , there is a transition to state s' with probability $P(s; a; s')$ and a direct cost $c(s; a)$. Because we have finite state and action sets, it is known that for discounted as well as for average cost there is at least one minimizing policy that is stationary and deterministic [22]. Our MDP is described as follows:

1. Definition of the set of states : The set S , which is the set of states of one three-stage handling system (formed by one wharf crane, a set of shuttle trucks, and a set of yard crane), allocated to serve a container ship. We consider the possible states: $S = \{i = (i_1, i_2) : 0 \leq i_1 \leq SZ_1 \text{ and } 0 \leq i_2 \leq SZ_2\}$ We assume that at time t , the system is at the state $i(t) = (i_1(t), i_2(t)) \in S$; where:
 - (a) i_1 and i_2 respectively, the numbers of shuttle trucks in the queues (noted Q_1 and Q_2) under the wharf and the yard crane; and
 - (b) SZ_1 and SZ_2 are, respectively, the buffer sizes of Q_1 and Q_2 .
2. Definition of the set of actions:
 The set of actions is defined by:

$$A = \{a = (a_1, a_2)\}; \text{ where}$$

$$a_i \in \{(+1T), (-1T), (+1C), (-1C), (0)\}; i = 1, 2$$
 and:

- (a) (+1T): allocate an additional shuttle truck (to queue Q_1 or Q_2) in order to increase the flow of containers between the quay and the storage yard;
 - (b) (-1T): liberate a shuttle truck (from the queue Q_1 or Q_2) in order to reduce congestion inside the terminal;
 - (c) (+1C): allocate an additional yard crane, such as straddle carrier, which is used to load (or unload) shuttle trucks when the operations speed of allocated yard crane is insufficient to ensure the continuity of wharf crane operations.
 - (d) (-1C): liberate the additional yard crane when it is useless.
 - (e) (0): do not take any action.
3. The transition probability:
 The transition probability (noted $P(i, a, i')$) is the probability of being in the state $i(t + \Delta t) = i'$ (at the date $t + \Delta t$) by selecting the action $a(t) \in A$ when the system was at the state $i(t) = i$ (at the date t).

$$P(i', a, i) = P[i(t + \Delta t) = i' \mid a(t) = a; i(t) = i] \quad (4)$$

Since the queues Q_1 and Q_2 are independents, the transition probability can be calculated by:

$$P(i', a, i) = P(i'_1, a_1, i_1) * P(i'_2, a_2, i_2) \quad (5)$$

where:

- (a) $P(i'_1, a_1, i_1) = P[i_1(t + \Delta t) = i'_1 \mid a_1(t) = a_1; i_1(t) = i_1]$;
- (b) $P(i'_2, a_2, i_2) = P[i_2(t + \Delta t) = i'_2 \mid a_2(t) = a_2; i_2(t) = i_2]$; and
- (c) $i = (i_1, i_2); i' = (i'_1, i'_2)$;

Since the action (+1T) consists to add an additional shuttle truck to the queue, thus the probability of $i_1(t + \Delta t) = i'_1$, when the action (+1T) is selected and when $i_1(t) = i_1$ is the same probability of $i_1(t + \Delta t) = i'_1$ when $i_1(t) = i_1 + 1$; Similarly, the probability of $i_1(t + \Delta t) = i'_1$ when the action (-1T) is selected and when $i_1(t) = i_1$ is the same probability of $i_1(t + \Delta t) = i'_1$ when $i_1(t) = i_1 - 1$. Thus, we have the following equations:

$$P(i'_1, (+1T), i_1) = P[i_1(t + \Delta t) = i'_1 \mid a_1(t) = (+1T); i_1(t) = i_1] = P[i_1(t + \Delta t) = i'_1 \mid i_1(t) = i_1 + 1] \quad (6)$$

$$P(i'_2, (+1T), i_2) = P[i_2(t + \Delta t) = i'_2 \mid a_2(t) = (+1T); i_2(t) = i_2] = P[i_2(t + \Delta t) = i'_2 \mid i_2(t) = i_2 + 1] \quad (7)$$

$$P(i'_1, (-1T), i_1)$$

$$\begin{aligned}
 &= P[i_1(t + \Delta t) = i'_1 \mid a_1(t) = (-1T); i_1(t) = i_1] \\
 &= P[i_1(t + \Delta t) = i'_1 \mid i_1(t) = i_1 - 1] \tag{8} \\
 P(i'_2, (-1T), i_2) \\
 &= P[i_2(t + \Delta t) = i'_2 \mid a_2(t) = (-1T); i_2(t) = i_2] \\
 &= P[i_2(t + \Delta t) = i'_2 \mid i_2(t) = i_2 - 1] \tag{9} \\
 P(i'_1, (0), i_1) \\
 &= P[i_1(t + \Delta t) = i'_1 \mid a_1(t) = (0); i_1(t) = i_1] \\
 &= P[i_1(t + \Delta t) = i'_1 \mid i_1(t) = i_1] \tag{10} \\
 P(i'_2, (0), i_2) \\
 &= P[i_2(t + \Delta t) = i'_2 \mid a_2(t) = (0); i_2(t) = i_2] \\
 &= P[i_2(t + \Delta t) = i'_2 \mid i_2(t) = i_2] \tag{11}
 \end{aligned}$$

We note that executing the actions (+1C) and (-1C) is equivalent to modify the value of servicing rate in the storage yard. Therefore, the calculation of $P(i'_1, (+1C), i_1)$ and $P(i'_2, (+1C), i_2)$ looks like the Eqs. (9) and (10). In this paper, we suppose that the service time of wharf and yard cranes is exponentially distributed with parameters μ_1 and μ_2 . So, the probability of begin servicing the next shuttle truck by the wharf crane in the next period Δt is:

$$\begin{aligned}
 P(T \leq \Delta t) &= 1 - e^{-\mu_1 \Delta t} \\
 &= 1 - [1 + (-\mu \Delta t) + \sum_{n=0}^{+\infty} \frac{(-\mu_1 \Delta t)^n}{n!}] \tag{12}
 \end{aligned}$$

where: $T \sim Exp(\mu_1)$.
 When Δt is small, we have $\sum_{n=0}^{+\infty} \frac{(-\mu_1 \Delta t)^n}{n!} \rightarrow 0$. so, $P(T \leq \Delta t) \simeq \mu_1 \Delta t$. Similarly, the probability of begin servicing the next shuttle truck by the yard crane in the next period Δt is: $P(T \leq \Delta t) \simeq \mu_2 \Delta t$. We suppose also that each queue (Q_1 and Q_2) has arrivals according to a Poisson process with parameters λ_1 and λ_2 . Then, the probability of an arrival of a shuttle truck to the queue Q_1 (respectively the queue Q_2) in the next period Δt is proportional to λ_1 (respectively λ_2):

$$P[(X_{t+\Delta t} - X_t) = 1] \simeq \lambda_1 \cdot \Delta t; \text{ when } : \Delta t \rightarrow 0 \tag{13}$$

and

$$P[(Y_{t+\Delta t} - Y_t) = 1] \simeq \lambda_2 \cdot \Delta t; \text{ when } : \Delta t \rightarrow 0 \tag{14}$$

where: $(X_t)_{t>0}$ and $(Y_t)_{t>0}$ are the Poisson processes that describe the numbers of arrivals of shuttle trucks at the queues Q_1 and Q_2 (respectively) before the date t . In the rest of this paper, we note $\rho = \Delta t$. Thus, we have the following equations:

$$P(i'_1, (0), i_1) = \begin{cases} \rho \lambda_1 (1 - \rho \mu_1); & \text{if } : i'_1 = i_1 + 1 \text{ and } i_1 < SZ_1 \\ \rho^2 \lambda_1 \mu_1 + (1 - \rho \mu_1)(1 - \rho \lambda_1); & \text{if } : i'_1 = i_1 \\ \rho \mu_1 (1 - \rho \lambda_1); & \text{if } : i'_1 = i_1 - 1 \text{ and } i_1 > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{15}$$

$$P(i'_2, (0), i_2) = \begin{cases} \rho \lambda_2 (1 - \rho \mu_2); & \text{if } : i'_2 = i_2 + 1 \text{ and } i_2 < SZ_2 \\ \rho^2 \lambda_2 \mu_2 + (1 - \rho \mu_2)(1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 \\ \rho \mu_2 (1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 - 1 \text{ and } i_2 > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{16}$$

$$P(i'_1, (+1T), i_1) = \begin{cases} \rho \lambda_1 (1 - \rho \mu_1); & \text{if } : i'_1 = i_1 + 2 \text{ and } i_1 + 1 < SZ_1 \\ \rho^2 \lambda_1 \mu_1 + (1 - \rho \mu_1)(1 - \rho \lambda_1); & \text{if } : i'_1 = i_1 + 1 \text{ and } i_1 < SZ_1 \\ \rho \mu_1 (1 - \rho \lambda_1); & \text{if } : i'_1 = i_1 \text{ and } i_1 > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{17}$$

$$P(i'_1, (-1T), i_1) = \begin{cases} \rho \lambda_1 (1 - \rho \mu_1); & \text{if } : i'_1 = i_1 \text{ and } i_1 - 1 < SZ_1 \\ \rho^2 \lambda_1 \mu_1 + (1 - \rho \mu_1)(1 - \rho \lambda_1); & \text{if } : i'_1 = i_1 - 1 \text{ and } i_1 > 0 \\ \rho \mu_1 (1 - \rho \lambda_1); & \text{if } : i'_1 = i_1 - 2 \text{ and } i_1 - 1 > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{18}$$

$$P(i'_2, (+1T), i_2) = \begin{cases} \rho \lambda_2 (1 - \rho \mu_2); & \text{if } : i'_2 = i_2 + 2 \text{ and } i_2 + 1 < SZ_2 \\ \rho^2 \lambda_2 \mu_2 + (1 - \rho \mu_2)(1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 + 1 \text{ and } i_2 < SZ_2 \\ \rho \mu_2 (1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 \text{ and } i_2 > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{19}$$

$$P(i'_2, (-1T), i_2) = \begin{cases} \rho \lambda_2 (1 - \rho \mu_2); & \text{if } : i'_2 = i_2 \text{ and } i_2 - 1 < SZ_2 \\ \rho^2 \lambda_2 \mu_2 + (1 - \rho \mu_2)(1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 - 1 \text{ and } i_2 > 0 \\ \rho \mu_2 (1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 - 2 \text{ and } i_2 - 1 > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{20}$$

$$P(i'_2, (+1C), i_2) = \begin{cases} \rho \lambda_2 (1 - \rho \mu'_2); & \text{if } : i'_2 = i_2 + 1 \text{ and } i_2 < SZ_2 \\ \rho^2 \lambda_2 \mu'_2 + (1 - \rho \mu'_2)(1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 \\ \rho \mu'_2 (1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 \text{ and } i_2 > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{21}$$

where: μ'_2 is the new servicing rate after the allocation of an additional yard crane.

After the liberation of the additional yard crane, the servicing rate is μ_2 Thus $P(i'_2, (-1C), i_2) =$

$$= \begin{cases} \rho \lambda_2 (1 - \rho \mu'_2); & \text{if } : i'_2 = i_2 + 1 \text{ and } i_2 < SZ_2 \\ \rho^2 \lambda_2 \mu'_2 + (1 - \rho \mu'_2)(1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 \\ \rho \mu'_2 (1 - \rho \lambda_2); & \text{if } : i'_2 = i_2 \text{ and } i_2 > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{22}$$

Table 2 The cost function related to operations at the quay

Action (a_1)	i_1	$C_1(a_1, i_1)$
(0)	$1 \leq i_1 \leq SZ_1$	i_1
	Otherwise	∞
(+1T)	$i_1 \geq SZ_1$	∞
	$i_1 < 1$	1
(-1T)	Otherwise	$i_1 + 1$
	$i_1 \leq 1$	∞
	$i_1 > SZ_1$	∞
	Otherwise	$i_1 - 1$

Table 3 The cost function related to operations at the storage yard

Action (a_2)	i_2	$C_2(a_2, i_2)$
(0)	$1 \leq i_2 \leq SZ_2$	i_2
	Otherwise	$b\infty$
(+1T)	$i_2 \geq SZ_2 - 1$	∞
	$i_2 \leq 1$	1
	Otherwise	$i_2 + 1$
(-1T)	$i_2 \leq 1$	∞
	$i_2 \geq SZ_2$	∞
	Otherwise	$i_2 - 1$
(+1C)	$1 \leq i_2 \leq SZ_2$	i_2
	Otherwise	∞
(-1C)	$1 \leq i_2 \leq SZ_2$	i_2
	Otherwise	∞

4. the cost function:

We define the cost function as follows:

$$C(a, i) = C_1(a_1, i_1) + C_2(a_2, i_2)$$

The cost function depends of the number of waiting shuttle trucks under each crane. The value of this function tends to be high when the length of the queue Q_1 or Q_2 is high (i.e., the number of waiting shuttle trucks exceeds the size of the queues) or when one of these queues is empty (i.e., the wharf or yard crane is waiting for a shuttle truck). We use to denote high values in the Tables 2 and 3.

5 Algorithms and Solution

5.1 Value Iteration Algorithm

One way to find an optimal policy is to find the optimal value function. It can be determined by a simple iterative algorithm called value iteration. Starting from a bounded cost function J , the iterates $(T^k J)_k$ converge uniformly to the optimal cost J^* . This algorithm is called the value iteration algorithm

(VIA), or sometimes successive approximation. VIA is an iterative method, which in general only converges asymptotically to the value function, even if the state space is finite. In practice, it is often very useful to have methods to accelerate the convergence of this algorithm. VIA is described as follows:

Input: set of states S , set of actions A , vectors of costs for every action $C(i; a)$, transition probability matrices for every action $P(i'|i; a)$, the discount rate γ , a very small number θ near to 0
 Initialise J_0 arbitrarily for every state. (Example: $J_0(s) = 0$)
 do
 for all i in S do
 for all a in A do
 $Q_{k+1}(i, a) = C(i, a) + \gamma \sum_{i' \in S} P(i'|i, a)J_k(i')$;
 and $J_{k+1}(i) = \min_{a \in A} Q_{k+1}(i, a)$; and
 $\pi(i) = \operatorname{argmin}_{a \in A} Q_{k+1}(i, a)$ gives the action a that has given J_{k+1} until $|J_{k+1}(i) - J_k(i)| < \theta$
 Output: J_k takes the at the last iteration, which is the optimal state-value function
 Output: π_k takes the k at the last iteration, which is the optimal policy

5.2 Policy Iteration Algorithm

In this sub-section, we present the policy iteration algorithm (also referred to as PIA) for finding the optimal policy in a discounted infinite horizon problem. As opposed to the value iteration algorithm, the output of PIA is not an approximation of the optimal policy, but the optimal policy itself. The policy iteration algorithm generates a sequence of improving stationary policies. The algorithm is as follows.

Initialization: Start with an initial stationary policy π_0 .
 Policy Evaluation: Given the stationary policy π_k , compute its cost J_{π_k} by solving the linear system of equations:
 $J_{\pi_k}(i) = C(i, \pi_k(i)) + \gamma \sum_{i' \in S} P(i'|i, \pi_k(i))J_{\pi_k}(i')$; $\forall i \in S$
 Policy Improvement: Obtain a new stationary policy π_{k+1} satisfying $\pi_{k+1} = \operatorname{argmin}_a \in A [C(i, a) + \gamma \sum_{i' \in S} P(i'|i, \pi_k(i))J_{\pi_k}(i')]$
 If the policy does not change, then stop. Otherwise repeat the process from step 2.
 Note that when the algorithm stops, we have $\pi_{k+1} = \pi_k$ and so, $J_{\pi_k} = T^* J_{\pi_k}$ and hence J_{π_k} is the fixed point of the Bellman operator. Thus, the policy π_k is optimal.

5.3 Obtained Solution

Once we have formalized, the problem of handling operations within the container terminal as a Markov decision process,

Table 4 The optimal actions for each state (i_1, i_2) according to the algorithms: VIA and PIA

i_1	i_2	$\pi^*(VIP)$	$\pi^*(PIA)$
0	0	(+1T, +1T)	(+1T, +1T)
1	0	(+1T, +1T)	(+1T, +1T)
2	0	(0, +1T)	(0, +1T)
3	0	(0, +1T)	(0, +1T)
4	0	(0, +1T)	(0, +1T)
5	0	(0, +1T)	(0, +1T)
0	1	(+1T, +1T)	(+1T, +1T)
1	1	(+1T, +1T)	(+1T, +1T)
2	1	(0, +1T)	(0, +1T)
3	1	(0, +1T)	(0, +1T)
4	1	(0, +1T)	(0, +1T)
5	1	(0, +1T)	(0, +1T)
0	2	(+1T, +0)	(+1T, +0)
1	2	(+1T, 0)	(+1T, 0)
2	2	(0, 0)	(0, 0)
3	2	(0, 0)	(0, 0)
4	2	(0, 0)	(0, 0)
5	2	(0, 0)	(0, 0)
0	3	(+1T, +0)	(+1T, +0)
1	3	(+1T, 0)	(+1T, 0)
2	3	(0, 0)	(0, 0)
3	3	(0, 0)	(0, 0)
4	3	(0, 0)	(0, 0)
5	3	(0, 0)	(0, 0)
0	4	(+1T, +0)	(+1T, +0)
1	4	(+1T, 0)	(+1T, 0)
2	4	(0, 0)	(0, 0)
3	4	(0, 0)	(0, 0)
4	4	(0, 0)	(0, 0)
5	4	(0, 0)	(0, 0)
0	5	(+1T, +0)	(+1T, +0)
1	5	(+1T, 0)	(+1T, 0)
2	5	(0, 0)	(0, 0)
3	5	(0, 0)	(0, 0)
4	5	(0, 0)	(0, 0)
5	5	(0, 0)	(0, 0)

the decision strategy, can be elaborated. In Table 4, we show the optimal policy according to value iteration (VIA), and policy iteration (PIA) algorithms. All these algorithms converge to the same solution. Policy iteration is usually recommended as an algorithm demonstrating a better performance than value iteration. However, policy iteration applied to non-discounted models without special restrictions might not even converge to an optimal policy, as in case of the policy iteration algorithm [40]. To overcome doubts, we preferred to execute the two algorithms to obtain the optimal solution for our MPD.



Fig. 5 A snapshot of our simulation software

6 Simulation Model

Because existing non-commercial simulator of container terminals is quite limited and cannot simulate a specific scenario like the obtained policy, we developed a simulation model based on distributed discrete-event-simulation technique (see Fig. 5). This technique was proposed as an alternative of discrete-event-simulation when the number of events to be simulated in the system is very high. This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others. Interactions in the physical system were simulated by message transmission among the processing elements. So, the simulation model is based on process-oriented paradigm and parallel computing. It uses multiple processing elements simultaneously to simulate the container terminal operations.

We used UML to model, through the class diagram, the system classes and their relationships. Class diagrams model the structure of the classes representing these “building blocks” of the system. Classes are coupled to each other through associations. In Fig. 6, we considered more general classes, which coincide in fact with categories in our system. These categories have subcategories that correspond to the elements of our model. Trucks, trains, and ships are means of transport. So, classes *Truck*, *Train*, and *Ship* inherit from the base class *TransportMean*. Import and export Yards of containers are also specializations storage yard. Thus, the classes *ImportYard* and *ExportYard* are all inherited from the super class *StorageYard*.

Some terminal resources such as cranes, shuttle trucks, are considered as processing elements. For this reason, they are inherited from the super class *java.lang.Thread* that represents an independent stream of execution within a program written with Java language. The types of events considered in this model are, for example, the arrivals of shuttle trucks to the queue under the quay or yard crane, loading/unloading

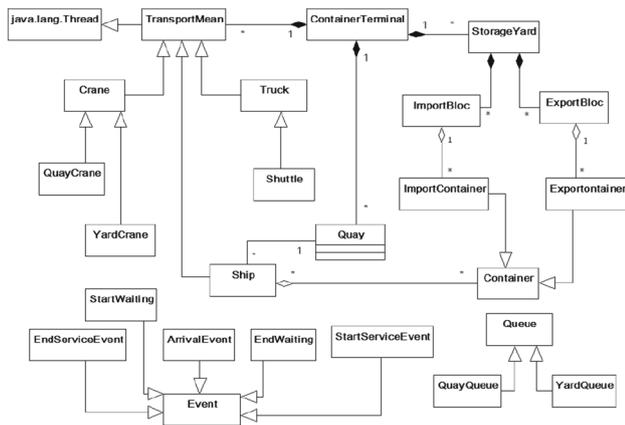


Fig. 6 Class diagram for the container terminal simulator

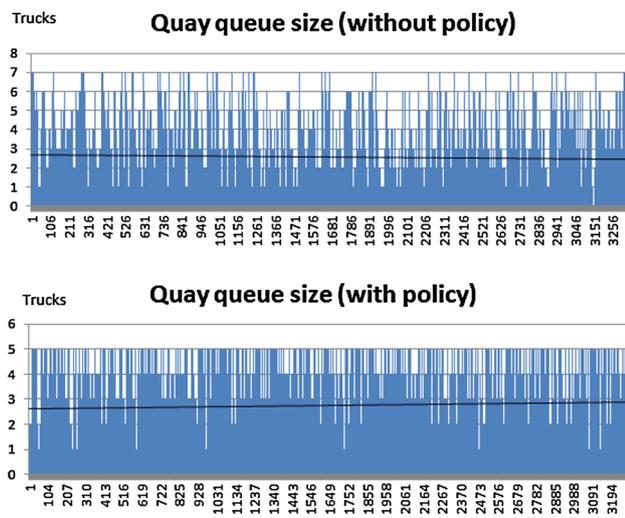


Fig. 7 Queue size under wharf crane during unloading operations in both cases: when the optimal policy is or is not applied (more than 3,000 simulated operations)

a shuttle truck, start waiting or end waiting of a crane or a truck etc.

The implementation of our model is made by Java programming language. The Java Thread API allows us to write the simulation application that can take advantage of multiple processors and perform background tasks while still retaining the interactive feel that users require.

The background of the simulator graphical interface is based the map of Casablanca container terminal in Morocco that consists of: gate, yard, and berth subsystems (see Fig. 5). Container handling equipments, considered in this paper, are storage: yard cranes, loading/unloading (wharf) cranes, and shuttle trucks. The storage yard is divided into several blocks that are served by tire or rail-mounted yard crane. The simulator is charged with a realistic reproduction of activities and flows that occur inside the terminal during loading and unloading operations.

7 Simulation Results

In order to test the obtained policy in Sect. 5, we performed multiple simulations of loading and unloading operations of containers. But only unloading operations are considered in the rest of this paper since the only difference between loading and unloading is the destination of the containers (ship or storage yard). In this section, we present two simulations cases:

- (1) In the first, we simulate unloading operations of more than 3,000 containers on the ship when we apply the optimal policy. The results are compared to the same case when this policy is not applied. We started from the allocation of one wharf crane and one yard crane installed in the reserved area to store the import containers of the ship. The number of required shuttle trucks is determined dynamically by the policy when it is applied, and it is fixed to 8 in the other case. Other parameters such as wharf and yard crane performance are the same in each case. According to the study carried out in Sect. 5, exponential distribution for crane service time and normal distribution for travel times of shuttle trucks assumptions are acceptable. Thus, we assume that the service time of wharf crane and yard crane is exponentially distributed with parameters $\mu_1 = 0.3$ and $\mu_2 = 0.3$ and the travel time of shuttle trucks between the quay and the storage yard is distributed according to normal distribution with parameters $m = 4$ and $\sigma^2 = 0.4$. Other probability distributions are supported by the simulator, but they are not considered to this paper. The output data are presented in the Figs. 7, 8, 9, and 10. They concern the queue length of trucks under wharf and yard cranes, idle times of each crane, and the total number of shuttle truck in the terminal during the time. Each figure presents a comparison of results between the two cases: When the obtained policy is applied or it is not applied. In Fig. 7, we show that the length of the queue can reach up to 7 without application of the policy, but it remains below 5 in the other case, which is not acceptable in most current container terminals where space is increasingly congested. The tendency curves presented in the two histograms in this figure show that the average of truck queue is the same in both cases (about 3 trucks), but the standard deviation around this average is different from one case to another. The applied policy has been harmonized the number of waiting trucks and ensured the normal performance of the quayside crane. In Fig. 8, it is shown a comparison of idle times in the two cases. The policy allows wharf crane to operate without stop unless a failure has occurred. In Fig. 9, we see a significant decrease of the queue length in the case of applying the policy. This is because our policy consists to increase the performances of unload-

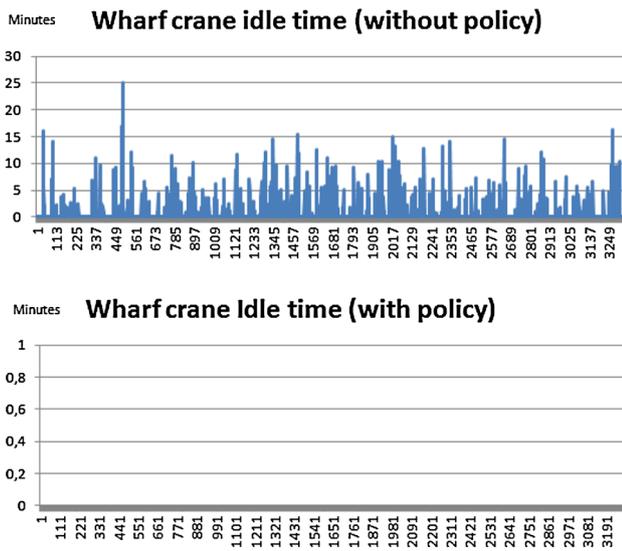


Fig. 8 Comparison of wharf crane idle time during unloading operations in both cases: when the optimal policy is or is not applied (more than 3,000 simulated operations)

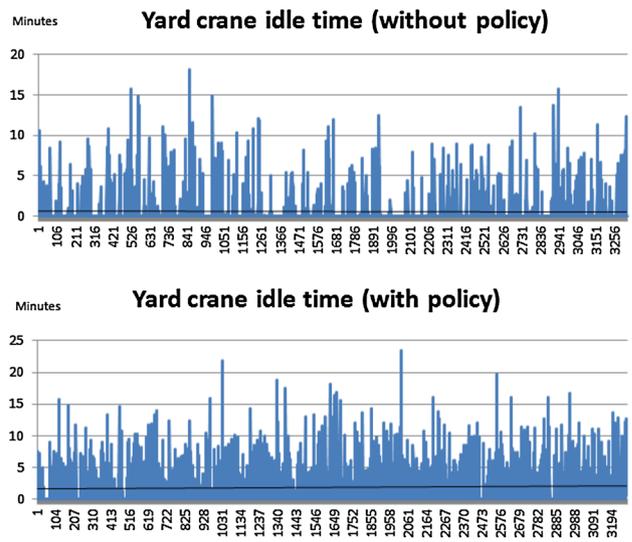


Fig. 10 Comparison of yard crane idle time during unloading operations in both cases: when the optimal policy is or is not applied (more than 3,000 simulated operations)

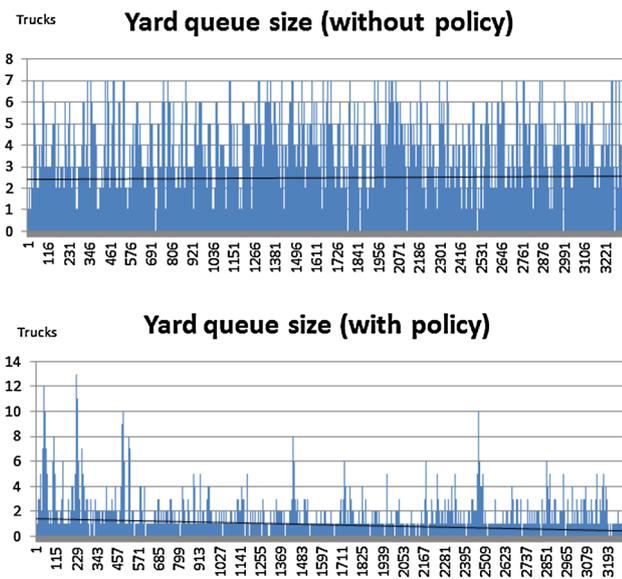


Fig. 9 Queue size under yard crane during unloading operations in both cases: when the optimal policy is or is not applied (more than 3,000 simulated operations)

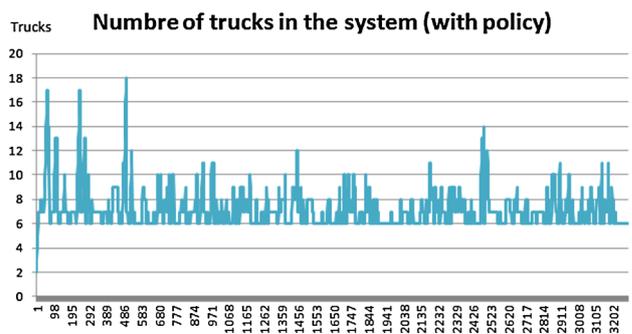


Fig. 11 The number of trucks during the simulation

ing operations in the storage areas when the length of queue becomes so large. This has a negative effect on the storage equipments, which is shown in Fig. 10. The average of idle times becomes superior in the case where the policy is not applied. It is not a big problem, since this increase is rewarded by allowing the wharf crane to operate without idle time. In addition, the total number of trucks allocated to serve the ship decreases over time (see Fig. 11).

(2) In the second case, the objective is to test the performance of the policy when more than one ship operate in the same time, where the trucks allocated to serve a particular ship could move the containers of another. For this, we run the simulation of simultaneous unloading operations of one, two, and three ships separately. The distance between two ships is assumed to be small. For each ship, we allocated one wharf crane and one yard crane. The number of containers is assumed to be more than 1,000 for each one. As in the first case, the number of allocated trucks is determined dynamically when the policy is adopted. Other parameters are the same as those of the first case ($\mu_1 = 0.3$, $\mu_2 = 0.3$, $m = 4$ and $\sigma^2 = 0.4$ for each ship). We note that the values of these parameters do not affect the performance of the policy since it consists to manage the number of shuttle trucks that are independent of these values. Starting from these initial inputs, we run the simulation and we

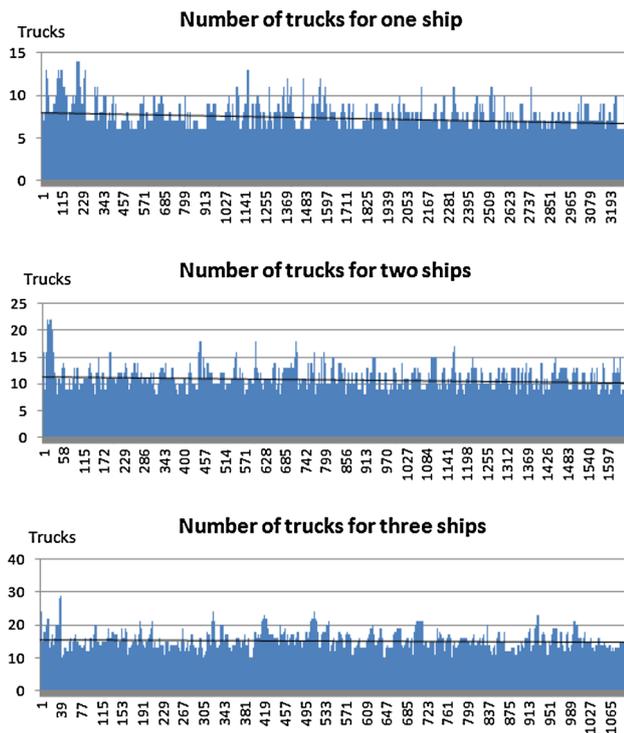


Fig. 12 Number of trucks during unloading operations

retrieve the data that describe the number of waiting shuttle trucks under each crane, idle time of equipments, and the total number of trucks in movement in the system to serve both ships. In this paragraph, we focus on the number of trucks allocated using the policy to serve ships in each case: one, two, or three ships. Figure 12 displays the comparison of the number of trucks between these three cases during unloading operations. If we focus on the trend curve, we will need on average 8 trucks for one ship, 12 trucks for two ships, and 16 trucks for three ships to ensure the unloading operations. In maximum, we will need 14 trucks for one ship, 21 trucks for two ship, and 28 trucks for three ships. So we find that number of trucks is not proportional to the number of ships. Without policy, it should be on average 8 trucks for one ship, 16 trucks for two ships, and 24 trucks for three ships. So we find that implementing the policy reduces significantly the number of trucks required to serve ships when they are more than two, thereby reducing the congestion of trucks traveling between the quay and storage areas. We note also an advantage of the policy at the idle time of trucks in queues.

8 Conclusion

In this paper, Markov decision process (MDP) methodology was proposed and applied to the container terminal in

order to improve efficiency of loading and unloading operations. MDP provides an interesting framework to model the decision-making, and the compact representation of the system provided by a MDP makes the decision strategy easier to implement. Value iteration and policy iteration algorithms mainly consist of finding the optimal policy that associates each state of the system to the action to be taken. The simulation is a valuable tool that allowed us to simulate the obtained solution. Even the approach using value iteration or policy iteration has not used in the domain of controlling a container terminal system, this work, by showing its performance, has made it an attractive one to pursue. Nevertheless, some possible perspectives to this work could be to develop the model more in order to make it more realistic by avoiding some assumptions such as the selected distributions of probability and the possibility of vehicles and stack equipments failure.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Steenken, D.; Voss, S.; Stahlbock, R.: Container terminal operation and operations research: a classification and literature review. *OR Spectr.* **26**, 117–135 (2004)
2. Rida, M.; Mouncif, H.; Boulmakoul, A.: Application of Markov decision processes for modeling and optimization of decision-making within a container port. *Adv. Intell. Soft Comput.* **96/2011**, 349–358 (2011). doi:10.1007/978-3-642
3. Yun, W.Y.; Choi, Y.S.: A simulation model for container-terminal operation analysis using an object-oriented approach. *Int. J. Prod. Econ.* **59**(13), 221230 (1999)
4. Lee, S.Y.; Cho, G.S.: A simulation study for the operations analysis of dynamic planning in container terminals considering RTLS. In: *Proceedings of 2nd International Conference on Innovative Computing, Information and Control*, 116 pp (2007)
5. Merkurjeva, G.; Merkurjev, Y.; Tolujev, J.: Computer simulation and metamodeling of logistics processes at a container terminal. *Stud. Inform. Control* **9**, 1–10 (2000)
6. El Sheikh, A.R.; Paul, R.J.; Harding, A.; Balmer, D.W.: A microcomputer-based simulation study of a port. *J. Oper. Res. Soc.* **38**, 673–681 (1987)
7. Kia, M.; Shayan, E.; Ghotb, F.: Investigation of port capacity under a new approach by computer simulation. *Comput. Indus. Eng.* **42**, 533–540 (2002)
8. Choi, Y.S.; Yun, W.Y.: Simulator for port container terminal using an object-oriented approach. Report, Brain Korea 21 Logistics Team (2000)
9. Carteni, A.; de Luca, S.: Analysis and modeling of container handling equipment activities. *Eur. Transp.* (46), 52–71 (2010)
10. Carteni, A.; Luca, S.de : Tactical and strategic planning for a container terminal: modelling issues within a discrete event simulation approach. *Simul. Model. Pract. Theory* **21**(1), 123–145 (2012). doi:10.1016/j.simpat.2011.10.005
11. Bielli, M.; Boulmakoul, A; Rida, M.: Object oriented model for container terminal distributed simulation. *Eur. J. Oper. Res.* **175**(3), 17311751 (2006)

12. Bruzzone, A.G.; Bocca, E.; Longo, F.; Massei, M.: Logistics node design and control over the whole life cycle based on web based simulation. *Int. J. Intern. Manuf. Serv.* **1**(1), 32–50. ISSN:1751-6048 (2004)
13. Shabayek, A.A.; Yeung, W.W.: A simulation model for the Kwai Chung container terminals in Hong Kong. *Eur. J. Oper. Res.* **140**(1), 111 (2004). doi:[10.1016/S0377-2217\(01\)00216-8](https://doi.org/10.1016/S0377-2217(01)00216-8)
14. Guestrin, C.; Koller, D.; Parr, R.; Venkataraman, S.: Efficient solution algorithm for factored MDPs. *J. Artif. Intell.* **19**, 399–468 (2003). doi:[10.1613/jair.1000](https://doi.org/10.1613/jair.1000)
15. Kim, K.H.: A crane scheduling method for port container terminals. *Eur. J. Oper. Res.* **156**, 75268 (2004)
16. Kim, K.H.; Kim, H.B.: The optimal sizing of the storage space and handling facilities for import containers. *Transp. Res. Part B* **36**(9), 821835 (2002)
17. Lee, B.K.; Kim, K.H.: Optimizing the block size in container yards. *Transp. Res. Part E* **46**, 120135 (2010). doi:[10.1016/j.tre.2009.07.001](https://doi.org/10.1016/j.tre.2009.07.001)
18. Gambardella, L.M.; Mastrolilli, M.; Rizzoli, A.E.; Zaffalon, M.: An optimization methodology for intermodal terminal management. *J. Intell. Manuf.* **12**(5-6), 521–534 (2001)
19. Lewis, B.; Erera A.; White, C.: Efficient container security operations at transshipment seaports. In: Proceedings of the Inaugural Annual Intelligent Transportation Society (Singapore) Symposium (2002)
20. Alattar, M.A.; Karkar, B.; Rajhans, N.: Simulation of container queues for port investment decisions. In: Proceedings of the Sixth International Symposium on Operations Research and its Applications (2006)
21. Ottjes, J.A. et al.: Simulation of a multiterminal system for container handling. *OR Spectr.* **28**, 447468 (2006)
22. Huynh, N.; Walton, C.M.: Robust scheduling of truck arrivals at marine container terminals. *J. Transp. Eng.* **134**, 347353 (2008)
23. Zhao, W.; Goodchild, A.V.: The impact of truck arrival information on container terminal rehandling. *Transp. Res. Part E* **46**, 327343 (2010)
24. Longo, F.: Design and integration of the containers inspection activities in the container terminal operations. *Int. J. Prod. Econ.* **125**, 272283 (2010). doi:[10.1016/j.ijpe.2010.01.026](https://doi.org/10.1016/j.ijpe.2010.01.026)
25. Legato, P.; Mazza, R.M.: Berthing planning and resources optimisation at a container terminal via discrete event simulation. *Eur. J. Oper. Res.* **133**, 537–547 (2001)
26. Tahar, R.M.; Hussain, K.: Simulation and analysis for the Kelang container terminal. *Logist. Inf. Manag.* **13**, 14–20 (2000)
27. Rida, M.; Boulmakoul, A.; Laurini, R.: Calibration and validation of container terminal simulation. In: Signal Processing and Information Technology, 2003. ISSPIT 2003. In: Proceedings of the 3rd IEEE International Symposium on, pp. 774–777, 14–17 Dec (2003). doi:[10.1109/ISSPIT.2003.1341235](https://doi.org/10.1109/ISSPIT.2003.1341235)
28. Boulmakoul, A.; Rida, M.: Discrete event simulation component specification for container terminals operational management. In: Proceedings of the 2nd IEEE ISSPIT (International Symposium on Signal Processing and Information Technology), pp. 266–271. Marrakesh, Morocco 18–21 Dec (2002)
29. Rida, M.; Boulmakoul, A.; Laurini, R.: Object oriented approach and java-based distributed simulation for container terminal operational management. In: Proceedings of the 2003 International Conference on Information Systems and Engineering (ISE/SCS 2003). ISBN: 1-56555-270-9
30. Barrientos, A.H.; Longo, F.; Mirabelli, G.: Simulation modeling approach to maritime container terminals: a state-of-the-art review 2012. Proceedings of the 2012—Summer Computer Simulation Conference, SCSC 2012, Part of SummerSim 2012 Multiconference 44 (10), pp. 89–96
31. Longo, F.; Huerta, A.; Nicoletti, L.: Performance analysis of a southern mediterranean seaport via discrete-event simulation. *Strojnikski Vestnik/J. Mech. Eng.* **59**(9), 517–525 (2013)
32. Landowski, B.: Example of applying Markov decision process to model vehicle maintenance process. *J. KONES Powertrain Transp.* **20**(4) (2013)
33. Higginson, J.K.; Bookbinder, J.H.: Markovian decision processes in shipment consolidation. *Transp. Sci.* **29**(3), 242–255 (1995)
34. Dekker, R.; Nicolai, R.P.; Kallenberg, L.C.M.: Maintenance and Markov decision models. Published online: (15 MAR 2008). doi:[10.1002/9780470061572.eqr085](https://doi.org/10.1002/9780470061572.eqr085)
35. Yu, X.; Stubberud, A.R.: Markovian decision control for traffic signal systems. In: Proceeding of the 36th Conference on Decision and Control San Diego, California, USA (December 1997)
36. Wu, Y.; Zhao, H.: Optimization maintenance of wind turbines using Markov decision processes. (01/2010). doi:[10.1109/POWERCON.2010.5666092](https://doi.org/10.1109/POWERCON.2010.5666092)
37. Tomasevicz, C. L.; Asgarpour, S.: Optimum maintenance policy using semi-Markov decision processes. *Electr. Power Syst. Res.* (01/2009). doi:[10.1016/j.epr.2009.03.008](https://doi.org/10.1016/j.epr.2009.03.008)
38. Gao, H.; Zhang, X.: A Markov-based road maintenance optimization model considering user costs. (8 MAR 2013). doi:[10.1111/mice.12009](https://doi.org/10.1111/mice.12009)
39. Sutton, R.S.; Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)
40. Russel, S.J.; Norvig, P.: Artificial intelligence: a modern approach. Prentice Hall, Englewood Clis (1995)