A multi-vessel quay crane assignment and scheduling problem: Formulation and heuristic solution approach

Yi-Min Fu, Ali Diabat *, I-Tsung Tsai

Department of Engineering Systems and Management, Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates

**Abstract**

This paper presents a new approach to analyze the integrated quay crane assignment and scheduling problem (QCASP). The problem determines the assignment of quay cranes to vessels and the sequence of tasks to be processed by each quay crane simultaneously, and accounts for important considerations such as safety margins between quay cranes (QCs), ordering conditions and vessel priority. Furthermore, QCs can travel from one vessel to another vessel whenever tasks are complete. The integrated problem is difficult to solve with exact methods due to its complexity. Therefore, a genetic algorithm (GA) is proposed to solve the integrated QCASP. Computational results validate the performance of the proposed GA.

**1. Introduction**

International trade plays a crucial role in global economy and it relies heavily on maritime transport, which accounts for approximately 90% of global trade transport (International Chamber of Shipping, 2014). The continuous increase of maritime transport has led to the containerization of goods, which facilitates the growth of this sector, making it an even more important component of international trade. Ports and container terminals are vital nodes in the network as they accommodate the transshipment of goods and then connect them to the rest of the transport network. Fig. 1 shows that global container traffic, as measured in twenty-foot equivalent units (TEUs), has more than doubled between 2001 and 2010.

Port operation involves a sequence of activities that require careful planning and execution to ensure maximum container throughput. Upon arrival at the port, the vessel will be assigned to a berthing position, where the offload and upload operations of containers are performed. Following the assignment of a berth, the vessel’s containers are discharged by assigned QCs according to a work schedule, and distributed to one of the storage blocks in the yard. The containers are then delivered by internal transportation to truck and train operation areas, where the containers are loaded onto trucks and/or trains to be transported to desired destinations. The operations that take place before the container reaches the storage yard are referred to as quayside operations, as they take place near the quay of the container terminal.

Operations taking place on the yard, where containers are stacked and stored, are known as yard side operations.

The cost of constructing berths and operating equipment such as the QCs is extremely high. Therefore, a major focus of the literature has been on the optimization of three quayside operation problems that are related to the utilization of berths and QCs. The first one is the assignment of quay space and service time to vessels, namely the berth allocation problem (BAP). The second one is the quay crane assignment problem (QCAP), which determines the allocation of quay cranes to vessels. In general, the number of QCs assigned to a vessel determines the handling time of the vessel. The fewer cranes assigned to a vessel, the longer time it takes to handle the ship. The last problem is the quay crane scheduling problem (QCSP), which determines the sequence of containers being processed by each crane to minimize the total handling time of the vessel.

The modeling approaches for quayside operations vary by the assumptions on: (1) vessel arrival time: the static berth allocation problem assumes all vessels have arrived when the berthing plan is to be determined, whereas the dynamic berth allocation problem allows vessels to arrive during the planning horizon (Imai, Nishimura, & Papadimitriou, 2001), (2) quay layout: a quay with a discrete layout has a fixed number of berths. Each berth can accommodate up to one vessel at a time (Imai, Nishimura, & Papadimitriou, 2008a; Imai, Nishimura, & Papadimitriou, 2008b). For a quay with a continuous layout, vessels will berth along the quay side as long as space allows. Continuous quay layout is therefore more efficient for space utilization but is also more difficult to solve for the optimal berthing plan (Meisel & Bierwirth, 2009), (3) vessel handling time: the static handling time model...
assumes handling time to be fixed and known in advance, whereas the dynamic handling time model assumes that the handling time is a function of the berthing position of the vessel, the number of cranes assigned to the vessel, and the work schedules of the cranes.

Integrated quayside operation models have been proposed to improve overall efficiency of quayside operations. Park and Kim (2005), as well as Meisel and Bierwirth (2009) study BAP and QCAP with integrated models. Tavakkoli-Moghaddam, Makui, Salahi, Bazzazi, and Taheri (2009) propose a model that studies the quay crane assignment and scheduling problem (QCASP). It is known that BAP needs to be solved together with QCAP when the vessel handling time is dynamic, as the number of quay cranes assigned to the vessels needs to be considered in the berth allocation planning phase. If the problems are solved independently, for instance, deciding the berthing time without considering the number of QCes serving the vessels means that the actual vessel handling time can be different from the given berthing time, which results in idling berths or delays in accepting new vessels. Since QC allocation affects the sequencing of QCs, QCs can be better utilized when QCAP and QCSP are considered simultaneously.

In this paper, we develop a model to integrate quay crane assignment and scheduling problems. Then, a genetic algorithm is proposed to determine the near optimal solution for the given problem. This paper thus makes a contribution by developing a new formulation to model the complex problem in an integrated manner, and proposing a customized genetic algorithm to solve the proposed formulation in an efficient manner.

The paper is organized as follows. Section 2 reviews the literature. Section 3 presents the formulation of the model. Section 4 introduces and explains the solution approach. Section 5 shows an experimental analysis of the model. Finally, Section 6 summarizes the results of the study and discusses potential directions for future research.

2. Literature review

Several studies on QCAP and QCSP have been conducted to improve the performance of container terminal operations. The QCSP was first discussed by Daganzo (1989), in which a container group is defined as a task. The problem is formulated as a mixed integer programming (MIP) model and the total weighted departure time of vessels is minimized. Daganzo's definition of a task is different from this paper, in which a task is defined as the loading or unloading operation of a single container, rather than a container group.

Kim and Park (2004) present an MIP model for the QCSP. The authors define a cluster as a collection of adjacent containers of the same group. They then define a loading/unloading operation for every cluster as a task in the formulation. Precedence relationships among clusters are considered. A branch and bound (B&B) method is then proposed to solve the QCSP. The paper provides a comparison between the B&B method and a heuristic search algorithm, namely greedy randomized adaptive search procedure (GRASP).

Interference constraints are introduced in the paper presented by Bierwirth and Meisel (2009). In an attempt to incorporate more realistic assumptions, the authors consider non-crossing constraints and clearance constraints for QCs (i.e. safety margin between adjacent QCs). The problem is also formulated as an MIP model and is solved with a tree-search-based heuristic solution procedure. An optimal solution for the QCSP with container groups can be found with a non-unidirectional schedule. Meisel (2011) introduces the concept of time windows in the QCSP, according to which each QC is restricted to serve a certain vessel within a certain pre-defined time window. QCs are discharged from low-priority vessels to high-priority vessels.

Park and Kim (2005) were the first to integrate the berth allocation and crane scheduling problem through a two-phase process. They determine the berthing time and position of each vessel as well as the number of cranes to be allocated to the vessel in the first phase. The objective of the first phase is to minimize the weighted sum of the handling cost of containers and the penalty cost incurred by unpunctual berthing. In the second phase, they schedule the assignment of individual quay cranes based on the results of the first phase.

To improve the approach presented by Park and Kim (2005), Meisel and Bierwirth (2009) present a model that takes into account the effect of the interference among QCs and the workload of horizontal transport means on terminal productivity. A Squeaky Wheel Optimization heuristic is proposed by the authors to generate the solutions, which they report always yields improved results compared to those obtained from the Lagrangian heuristic proposed by Park and Kim (2005).

Tavakkoli-Moghaddam et al. (2009) extend the model proposed by Kim and Park (2004) to model a set of vessels in parallel. This paper formulates an MIP model for the integrated QCSAP. In the dynamic planning horizon a QC is assigned from the incumbent vessel to the next vessel after the completion of all the tasks in the incumbent vessel. A genetic algorithm is proposed to obtain a near optimal solution for this complex and large-scale problem.

Choo, Klabjan, and Simchi-Levi (2009) introduce a multi-vessel problem in the QCSP. Yard congestion constraints are considered to link the vessels. Clearance constraints are also included in this paper. The authors address the single-ship case by reformulating an MIP model as a generalized set covering formulation, while the multi-ship case is decomposed in the spirit of Lagrangian relaxation. Both cases are then solved using B&P.

The recent work of Diabat & Theodorou, 2014 introduces a new type of formulation, which practically transforms the integrated
assignment and scheduling problem into a QC to bay assignment problem. This approach essentially allows the elimination of the time index, and can be efficiently solved with GA. The authors report very good results with the main advantage of low computational time and complexity, due to the elimination of the time factor.

The research on QC operational problems is extensive, yet the developed models are generally complex due to the interrelation between different phases, the dynamics of ports, and physical constraints, such as the availability of berths, the number of QCs, and non-crossing constraints for QCs. The majority of these problems are NP-hard and this is why heuristics are the preferred options when addressing them and any extension of these problems. Park and Kim (2005), Choo et al. (2009), and Guan, Yang, and Zhou (2010) propose Lagrangian heuristics for their models. Tavakkoli-Moghaddam et al. (2009), Kavegh, Huynh, and Rahimian (2012), Chung and Choy (2012), and Diabet and Theodorou (2014) propose genetic algorithms for the QCSP models.

3. Methodology

This section provides a detailed description of the quay crane assignment and scheduling problem (QCASP). Then the mathematical formulation is presented.

3.1. Problem description

The objective of the QCSP is to minimize the makespan for each vessel by determining the sequence of loading/unloading operations to be executed by every QC. Table 1 provides an example of a load profile for 2 vessels with 3 and 4 bays, respectively. In this example, QCs are assigned before their schedules are determined. Fig. 2 shows a solution for the example with 3 cranes. In this solution, vessel 1 is assigned to QC 1, vessel 2 is assigned to QC 2 and QC 3. As shown in Fig. 3, the total handling time can be reduced if QC 2 is assigned to vessel 1 in the beginning of its schedule. This example shows that QCs can be utilized more efficiently when QCAP and QCSP are considered simultaneously, since QCs can be assigned to different vessels whenever necessary.

Following Choo et al. (2009), the berthing time and location of each vessel are assumed to be given. The container terminal operator has the information of the load profile of each vessel. The target is then to determine the assignment of QCs as well as the sequence of loading and unloading operations within a planning horizon so that the makespan can be minimized. Fig. 4 illustrates a planning horizon, where the work completion flag is introduced. The work completion flag is set to 1 when the needed operation time of the vessel is achieved. Minimizing the makespan of container terminals thus requires a maximization of the number of work completion flags during a given planning horizon.

3.2. Model assumptions and formulation

The model is built on the following assumptions:

(i) Quay cranes can be moved from a vessel to another while serving it.
(ii) Quay cranes have identical working rates.

(iii) Bidirectional quay crane movement is allowed.
(iv) Quay cranes are positioned on a single track; therefore, they are in ascending order from left to right.
(v) Safety margins between quay cranes are considered.
(vi) Berthing positions for vessels are given.
(vii) Different vessels have different priority.
(viii) Traveling time for QCs is not considered.

The notation used to formulate the problem is shown below:

Sets:

\[ \ K \triangleq \text{set of vessels, indexed by } k \]
\[ \ J_k \triangleq \text{set of bays of vessel } k, \text{ indexed by } j \]
\[ \ Q \triangleq \text{set of quay cranes, indexed by } q \]
\[ \ T \triangleq \text{set of time segments, indexed by } t \]

Parameters:

\[ l_{ij} \triangleq \text{location of bay } j \text{ in vessel } k \]
\[ m \triangleq \text{safety margin} \]
\[ p_{kj} \triangleq \text{work to be done for bay } j \text{ in vessel } k \]
\[ c_v \triangleq \text{unit operation cost of vessel } k \]

Decision Variables:

\[ x_{qjt} = \begin{cases} 1 & \text{if quay crane } q \text{ is assigned to bay } j \text{ of vessel } k \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \]
\[ f_{jt} = \begin{cases} 1 & \text{if bay } j \text{ of vessel } k \text{ is fully handled at time } t \\ 0 & \text{otherwise} \end{cases} \]
\[ r_v = \begin{cases} 1 & \text{if vessel } k \text{ is fully handled at time } t \\ 0 & \text{otherwise} \end{cases} \]

The following is the IP formulation of this model.

Max \[ \sum_{k \in K} \sum_{t \in T} c_v r_v \]

subject to

\[ \sum_{k \in K} \sum_{j \in J_k} x_{qjt} \leq 1 \quad \forall q \in Q, \forall t \in T \]  
(2)
\[ \sum_{j \in J_k} x_{qjt} \leq 1 \quad \forall k \in K, \forall j \in J_k, \forall t \in T \]  
(3)
\[ \sum_{k \in K} \sum_{q \in Q} \sum_{j \in J_k} x_{qjt} \leq |Q| \quad \forall t \in T \]  
(4)
\[ \sum_{k \in K} \sum_{j \in J_k} l_{ij} \cdot x_{qjt} \cdot f_{jt} - \sum_{k \in K} \sum_{j \in J_k} l_{ij} \cdot x_{qjt} \geq m \quad \forall q \in Q, \forall t \in T \]  
(5)
\[ \sum_{q \in Q} x_{qjt} = p_{kj} \quad \forall k \in K, \forall j \in J_k \]  
(6)
\[ \sum_{q \in Q} x_{qjt} \geq f_{jt} \cdot r_v \quad \forall k \in K, \forall j \in J_k, \forall t \in T \]  
(7)
\[ f_{jt} \geq r_v \quad \forall k \in K, \forall j \in J_k, \forall t \in T \]  
(8)
\[ x_{qjt}, f_{jt}, r_v \in \{0, 1\} \quad \forall k \in K, \forall j \in J_k, \forall q \in Q, \forall t \in T \]  
(9)

The objective function (1) aims to maximize the sum of the weighted work completion flag. Constraints (2) and (3), respectively, ensure that a QC can only be positioned at a single bay at any time period and that only one QC can be positioned at a bay at any time. Constraint (4) enforces that the total number of in-service QCs cannot exceed the total number of available QCs at any time. Constraint (5) enforces the QC ordering condition by positioning QCs in an increasing order from left to right. It also enforces
the QC clearance conditions by positioning QCs at least m bays away. Constraint (6) restricts all container assignments to be completed within the planning horizon. The sum of total time segments of cranes working on a bay must cover the handling of all containers. Constraint (7) defines the work completion flag for each bay, \( f_{jk} \), is set to 1 after all containers in bay j of vessel k are treated. Constraint (8) ensures that a vessel is finished after the task for every bay in the vessel is completed. Constraint (9) makes sure that variables \( x \), \( f \), and \( r \) are all binary.

4. Solution method

The QCASP is a challenging problem to solve due to the complex decisions to be made for assigning every QC to a vessel and every QC to a bay at every time unit. Genetic algorithms have been used in the literature to tackle large and complex problems when exact solution methods are not able to solve such problems. This section presents the general concept of genetic algorithms, followed by an introduction of the proposed genetic algorithm.

4.1. Genetic algorithm

Genetic Algorithm (GA) is a search heuristic for optimization problems that works by mimicking the natural evolutionary process. Fig. 5 illustrates the process of a genetic algorithm. First, there is an initial population. From this population, the fitness of each individual is evaluated. The individuals with good fitness values are selected as parents for reproducing the next generation. New individuals are then generated through genetic operators, and reinserted into the population to replace less fit individuals (Chipperfield, Fleming, Pohlheim, & Fonseca, 1994).

4.2. Development of the genetic algorithm

4.2.1. Chromosome representation

Chromosome representation is the first step in the implementation of the genetic algorithm. Each time segment needed for all vessels to be discharged represents a gene in the chromosome. There are \((\sum p_{jk})\) genes in a chromosome, where \( p_{jk} \) is the time needed for a container to be discharged from bay j of vessel k. Table 2 shows a loading profile for a vessel. Fig. 6 illustrates an example of a chromosome representation for the loading profile. Each row represents the working schedule for a QC. The value of the gene indicates the location of each QC. The chromosome therefore specifies both the schedules and the assignments of QCs. The number of times each integer appears in the chromosome reflects the time needed to finish unloading/loading, which varies by bay. The example in Fig. 6 shows that 2 time segments are required to finish the tasks in bay 1, and therefore it appears twice in the chromosome.

4.2.2. Generating initial population

There are two ways to generate the initial population. One is by random generation, which gives the initial population great diversity. The other is using heuristics. Although the latter process generates individuals with better quality, it may lead to a local optimal solution due to the limited diversity among the initial population.
In this paper, the random method is used to generate the initial population. However, we allow only the feasible individuals to enter the population. The algorithm to generate an individual is described as follows. An integer array is first generated based on the operation time required for every bay. Then, the sequence of the integers is ordered randomly and the array is reshaped into a matrix with a number of rows equal to the number of QCs. For every column, the integers are in ascending order to satisfy the non-crossing constraint for QCs. Furthermore, the difference between the integers in each column cannot exceed the safety margin. Fig. 7 shows an example of a feasible and an infeasible chromosome.

### 4.2.3. Fitness evaluation and parent selection

After the decision variables are set, the objective function is evaluated for each individual in the population. Since the objective is to maximize the weighted work completion flag, the individual with the highest objective value is assigned the highest fitness value.

The roulette wheel selection is then applied to choose the chromosomes that will serve as parents of the next generation. The probability for a chromosome to be selected is given to each chromosome in the population according to its proportion of fitness (Yu & Gent, 2010). More specifically, a chromosome will be assigned a high probability of being chosen if it has a higher objective value.

### 4.2.4. Genetic operators

After two chromosomes are selected, they are paired up to breed a new chromosome. The new chromosome inherits characteristics from the parents. The procedures performed to recombine the chromosomes are described in this section.

#### 4.2.4.1. Crossover

A patching crossover is carried out in the genetic operation. It is based on uniform crossover. T-Moghaddam et al. (2001) proposed an extended patching crossover for the chromosome to represent the sequence of jobs in vessels. A binary string is generated to determine the position taken by a gene from a specific parent. The offspring takes the gene from parent 1 if the value in the corresponding position in the matrix is 1. Then the genes from parent 2 fill the remaining gene locations. Fig. 8 illustrates the crossover operation proposed in this paper.

First of all, a binary matrix \( A \) is generated with \( \frac{\sum k \sum p_{jk}}{2} \) of ’1’s. Then the gene is taken from Parent 1 when the corresponding position in \( A \) is 1. The gene chosen in the previous step from Parent 2 is deleted and the remaining genes from Parent 2 are filled in the empty locations sequentially. As mentioned before, only feasible chromosomes are allowed to enter the population.

#### 4.2.4.2. Mutation

In order to maintain diversity of the chromosome, mutation is applied in the GA with a low probability to modify the genes in the chromosome. A swap mutation is used in this paper. Fig. 9 shows an example of swap mutation, where two positions in each row are chosen randomly to apply the swaps.

Once an offspring is generated through chromosome selection and genetic recombination, the least fit chromosome in the old population is replaced by the offspring.

### Table 2

A loading profile.

<table>
<thead>
<tr>
<th>bay</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC-hours</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

**Fig. 6.** Chromosome Representation.

**Fig. 7.** An example of a feasible and an infeasible chromosome.

**Fig. 8.** Illustration of the crossover operation.

**Fig. 9.** Illustration of the swap mutation.
population will be replaced by the offspring. This process keeps the size of the population at constant, and will be iterated until the pre-specified number of generations is reached.

5. Experimental analysis

In this section, the proposed model is validated and the performance of the developed GA is tested by different sizes of instances. The problems are also solved to optimality using the commercial software GAMS, in order to have a reliable measure of comparison and accurately evaluate the performance of the GA.

The problem sizes range from small scale, consisting of 2 vessels and 3 cranes, to practical sized problems, consisting of 8 vessels and 12 quay cranes. The number of bays on every vessel, as well as the number of containers carried by each bay were randomly generated, while the number of ships and available quay cranes were predefined, based on the desired problem size.

Table 3 summarizes the computational results for the instances of different sizes, obtained from GAMS and from the GA approach. The proposed GA obtains the optimal or near-optimal solutions for all instances, while producing an average gap of 2.67% for the instances tested in reasonable times. As far as computational efficiency is concerned, it is worth mentioning that GA requires less time when compared to GAMS, as the problem size increases. Even though for small problems GA reports higher times, the significant reduction in time for large problems constitutes an important advantage of the approach. Fig. 10 illustrates the comparison between GAMS and the proposed GA.

Another important factor to mention is that for large sizes GAMS is not able to reach a solution. On the other hand, GA reports solutions for all problem sizes within reasonable time. Therefore, it is safe to state that the proposed approach is recommended for use in large problems, as it yields results with satisfactory gaps and most importantly within short computational time. However, it would be useful to further investigate the performance of the proposed approach in terms of solution quality, in order to solidify this statement.

In order to acquire this information, regarding the quality of solutions provided by GA in the instances for which it was not possible to obtain an optimal solution through GAMS, we compare the best feasible solution (not necessarily optimal) achieved by GAMS with the result from GA and we report the results in Table 4. Through this comparison it becomes evident that the quality of solutions obtained by GA is always better than those obtained using GAMS. Furthermore, the improvement actually becomes greater as the size of the problems increases.

6. Conclusion and future research directions

In this paper, we develop a model to analyze the integrated QCSAP for container terminals taking into account practical constraints such as safety margins and QC ordering. A genetic algorithm is then presented as the solution approach. Several problem instances are tested to verify the performance of the GA. The results show a reasonable gap of approximately 2.67% between the optimal solution from GAMS and the proposed GA, with the GA approach having a considerable advantage in computational efficiency for larger problems.

Table 4: Comparison between CPLEX and GA in solving large-sized instances.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Number of vessels</th>
<th>Number of bays per vessel</th>
<th>Computational time (min)</th>
<th>Best upper bound by CPLEX</th>
<th>Best upper bound by GA</th>
<th>GA solution improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>Random integer between 20 and 50</td>
<td>30</td>
<td>15,372</td>
<td>13,975</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>Random integer between 20 and 50</td>
<td>30</td>
<td>24,226</td>
<td>19,075</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>Random integer between 20 and 50</td>
<td>30</td>
<td>36,101</td>
<td>23,908</td>
<td>51</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>Random integer between 20 and 50</td>
<td>30</td>
<td>44,540</td>
<td>27,494</td>
<td>62</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>Random integer between 20 and 50</td>
<td>60</td>
<td>49,093</td>
<td>26,394</td>
<td>86</td>
</tr>
<tr>
<td>6</td>
<td>35</td>
<td>Random integer between 20 and 50</td>
<td>60</td>
<td>58,565</td>
<td>28,022</td>
<td>109</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>Random integer between 20 and 50</td>
<td>60</td>
<td>66,291</td>
<td>29,463</td>
<td>125</td>
</tr>
<tr>
<td>8</td>
<td>45</td>
<td>Random integer between 20 and 50</td>
<td>60</td>
<td>74,578</td>
<td>29,831</td>
<td>150</td>
</tr>
<tr>
<td>9</td>
<td>50</td>
<td>Random integer between 20 and 50</td>
<td>120</td>
<td>84,074</td>
<td>31,488</td>
<td>167</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>Random integer between 20 and 50</td>
<td>120</td>
<td>403,731</td>
<td>59,198</td>
<td>582</td>
</tr>
</tbody>
</table>
achieving these results within significantly lower computational time. Furthermore, for large instances of the problem, where GAMS fails to reach a solution, the GA obtains one within good time. Seeing as we could not evaluate the quality of the solution from GA for larger problems, as GAMS was incapable of reaching optimality, we further investigated the matter through the comparison of the best attainable feasible solution between the two methods. In this case, the superiority of GA was clearly demonstrated by the fact that it achieved better solutions in all cases, and very importantly within reasonable computational time. In fact, the improvement was greater when comparing for problems of large sizes. Therefore, as a final remark, the method is deemed suitable and it is recommended for the quay crane assignment and scheduling problem.

This research can be extended in a number of important ways. Since berth planning affects the allocation and sequencing of QCs, future extension of the work may involve the analysis on the integration of berth allocation and QC assignment and scheduling to optimize container terminal operations. This research can also be expanded to include the effect of traveling times of quay cranes on QCSAP as QC traveling time can affect QC sequencing significantly due to the physical constraints of terminals. The effects of dynamic arrival of vessels on crane availability can also be included. Finally, in terms of differently formulating the problem, alternative objective functions can be considered.

References


