

# A SPLIT TASK SOLUTION FOR QUAY CRANE SCHEDULING PROBLEM IN MID-SIZE CONTAINER TERMINALS

*Neven Grubišić, Čedomir Dundović, Srdjan Žuškin*

Original scientific paper

A solution for quay crane scheduling problem with unchanged number of cranes during container vessel handling process is explored to minimize a vessel handling time and optimize quay crane utilization. Targets are small and mid-size terminals where high reliability of forecasted scenario is requested by the shipping companies while, at the same time, terminal resources are limited. Two models for the problem solution are proposed based on the full task and split task algorithms; in both cases grouping of tasks into clusters and operation zone limits are considered. Comparison of two models shows that split-task algorithm may improve crane utilization rate and vessel handling time in many cases. Proposed solutions combine existing research achievements with authors own approach in container terminal optimization development.

**Keywords:** container ports; container terminal logistics; container terminal optimization; quay crane schedule; split task algorithm; tactical logistic problems

## Rješenje problema redosljeda rada obalnih dizalica na kontejnerskim terminalima uz dijeljenje zadatka

Izvorni znanstveni članak

Istražuje se problem određivanja redosljeda rada nepromijenjenog broja obalnih kontejnerskih dizalica za vrijeme prekrcajnih operacija na brodu s ciljem minimiziranja vremena prekrcaja i optimalnog iskorištenja radnog vremena dizalica. Objekt istraživanja su kontejnerski terminali manje i srednje veličine gdje su pouzdanost usluge i ostvarivanje predviđenog scenarija prekrcaja ključni zahtjevi brodarima, a s druge strane prekrcajni resursi ograničeni. Predložena su dva modela rješavanja problema temeljeni na dva algoritma, bez dijeljenja zadatka i uz dijeljenje zadatka. U oba slučaja uzeti su u obzir grupiranje zadataka u klustere i ograničenja operativnih zona. Usporedba dva modela pokazuje da se primjenom algoritma s dijeljenim zadacima u mnogim slučajevima poboljšava iskorištenje obalnih dizalica i krajnje vrijeme dovršetka prekrcajnih operacija. Predloženim rješenjima autori kombiniraju postojeća istraživanja i vlastite spoznaje do kojih su došli razvojem modela optimizacije obalnih prekrcajnih operacija na kontejnerskom terminalu.

**Ključne riječi:** algoritam s dijeljenjem zadatka; kontejnerske luke; logistika kontejnerskih terminala; optimizacija kontejnerskog terminala; redosljed obalnih dizalica; taktički logistički problemi

## 1 Introduction

One of the specific operational problems on container terminals is Quay Crane Scheduling problem (QCSP). This problem may be explained as searching the best scenario of crane workload organization in ship-to-shore container handling operations. The goal is to find optimum assignment of the quay cranes (QCs) to load/unload bunch of containers stowed into ships holds/bays and grouped into tasks. QCSP is one of the standard tactical logistic problems within the container terminal [1] and it is very important for efficiency of port services provided for ships.

This particular problem affects functionality of sea-side terminal subsystem but has also impact on other subsystems of the container terminal [2]. Moreover, there is mutual interest of terminal operator and shipping companies to design and organize terminal handling processes efficiently with high performances and capacity utilization [3]. A range of the crane operating zone may be valuable information for job organization, berth planning and overall terminal resource planning also in multipurpose terminal environment [4].

The QCSP has been studied by many researchers: Among others, there are Daganzo [5], Kim & Park [6], Liu et al. [7], Lim et al. [8], [9], Zhu&Lim [10], Lee et al. [11], Meisel [12], Meisel & Bierwirth [13], Lajjam et al. [14]. The great majority of these researches targeted big-size container terminal, particularly located in Far East ports. However, there are needs for closer approach to the solutions for mid-size terminals where existing

infrastructure and equipment may limit system performances [15].

This research is focused on small and mid-size terminals in public ports where certain number of carrier services exists like in Mediterranean and North Adriatic seaports. Small distances between ports as well as different relation between ships size and average handling demand for each port put pressure on terminal operator to guarantee the quality and reliability of service to carriers.

Two types of QCSP optimization models are redeveloped and evaluated through two main criteria: vessel handling time and crane utilization rate. The main feature of both models is constant allocation of the quay cranes during vessel stay at berth. That means no interference in planning schedule may occur as a result of dependences and delays in crane accessibility during switch task from one vessel/berth to another.

The structure of the paper is as follows: In the Section 2, problem scope and optimization methodology is described; Section 3 specifies model development; Section 4 explains method of data generation used for model testing; finally, in Section 5 and in Conclusion section, analysis and validation of optimization results are summarized.

## 2 Optimization methodology

The objective of this research is to provide optimized handling scenarios for each ship servicing at container berth, for each possible combination of QCs.

For each ship berthed alongside the quay certain number of QCs has to be allocated. The number of QC

depends on cargo handling demand, distribution of containers through holds/bays and availability of cranes. However, availability of cranes is second step in overall seaside optimization modeling and is considered as separated problem known as Quay Crane Allocation Problem (QCAP). Contribution to QCAP solution can be found, e.g. in [16] or [17], where fixed handling time should be known prior to allocation modeling. More valuable concept has been proposed by Meisel [12], where integrated solution for berth and crane assignment has been proposed (BACAP). That integrated solution is based on variable handling time calculation and on QCSP solving.

Therefore, QCSP optimization is first step in overall sea-side container terminal optimization and this approach is followed here. That means that the number of cranes to be assigned to the vessel is not known beforehand. For each rational number of QCs, a separate solution has to be found that is forming set of optimized scenarios which may be interpreted as QCSP solution for designated number of cranes (Fig. 1). Output from each of those solutions gives four important parameters that may be used for QCAP or BACAP problem solving:

- number of QC gangs with overall utilization factor
- total ship handling/processing time
- full/split task selection option.

The core of the optimal scenario development is QCSP solving. For this purpose two different models are redeveloped: model with single crane to task assignment called *full task* assignment algorithm and a model with more than one crane to task assignment called *split task* algorithm. Full task assignment means each crane must complete assigned task to the end once loading/unloading started. Split task means a task may be shared between cranes and more than one crane may perform the same task, consecutively, during processing the vessel.

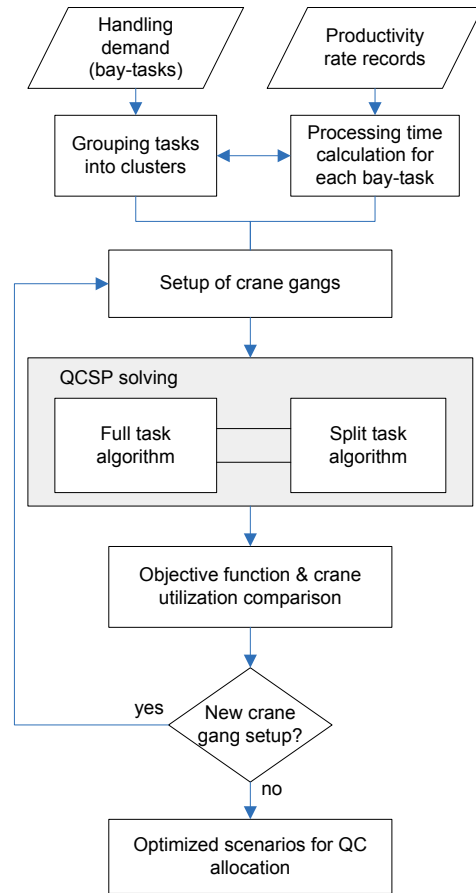


Figure 1 Flow chart for optimal scenario development

Task sharing allows split of a task during handling process where a crane may leave the job before finishing all loading/unloading operations associated with that task. Another crane later on, has to be scheduled to finish uncompleted job.

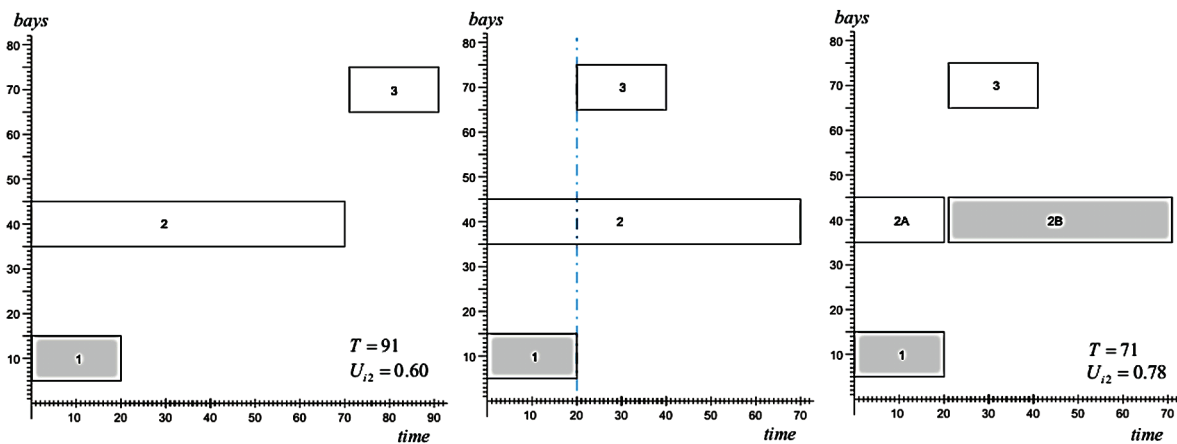


Figure 2 Method and effects of task splitting

This idea is not completely new and was proposed in [7], under so-called "vessel-level models" where authors highlighted problem of extensive partitioning of tasks causing too many unproductive movements of cranes. To eliminate that, operational zone and travel path limits have been included into the model presented here.

The purpose of using split-task algorithm in QCSP solving is to get higher utilization rate for set of QCs engaged in ship-to-shore handling operations and

consequently to get lower completion time of all tasks (makespan). Method and effects of splitting is shown in Fig. 2. There are three tasks represented by the rectangle where task processing time is on x-axis, and bay position on y-axis. In this example task number 2 has the largest quantity of containers for processing. Two QCs, represented by grey/white color combination are available for assignment of jobs. The leftmost graph shows optimal solution for QCSP with full task algorithm, while

rightmost graph shows results of QCSP with split task algorithm applied. The idea behind is to break the operation of task 2 before the finish, and to switch the cranes. It must be noted that non-crossing and safety distance constraints must be satisfied in both cases.

Onboard containers are stowed into cells and all cells within the same bay may correspond to one job task since they are reachable by the *QC* without crane movement alongside the quay. It is reasonable to group the adjacent jobs into one single task if they are too close to be handled by two cranes simultaneously. This method is known as clustering and has been explained in detail in [6]. Where there is no large volume of container demand for particular vessel port call, a stowage plan should allow parallel jobs handling to minimize the negative impact on total vessel service time. However, clustering of jobs into single task may be considered as potential solution, when applicable.

Fig. 3 shows schedule scenario for four tasks processed by two *QC* when clusters applied. For each task, lower and upper cluster boundary are defined and expressed with corresponding bay numbers. All rectangles have the same height conforming to overall length of each *QC* including safety distance between them. Position of *QCs* on y-axis is relative to bay numbers on the vessel. Small bay numbers from the left side of y-axis represent cluster boundary and cluster ranges for each group of jobs that are joined in one common task. Each *QC* may be in any time located in any position inside cluster boundary. Therefore, safe environment should be guaranteed to avoid interference between them and possible collisions when two adjacent cranes are located on their boundary positions.

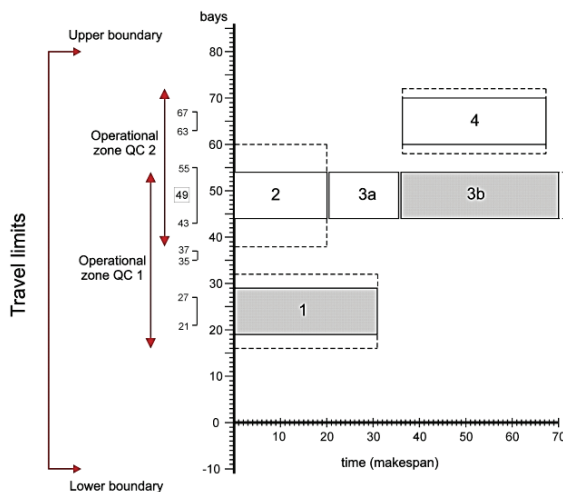


Figure 3 Example of a QCSP modeling

In Fig. 3 dashed lines display required safety distances between *QCs*. Note that task 3 is treated as single task/job (located at bay no. 49) while task 2 is multi-job task with cluster boundary (between bays 43÷55). Task 3 has more volume of containers located on the single bay than overall jobs grouped into task 2; hence if we consider it as cluster task it will affect the total vessel service time and cut down *QCs* utilization factor.

There are three parameters defining position of a task, expressed in a vessel bay number:

- $h_i^{mid}$  – central position of the cluster task  $i$
- $h_i^{lo}$  – lower boundary of the cluster task  $i$

- $h_i^{up}$  – upper boundary of the cluster task  $i$

The proposed models for QCSP solution take into consideration cluster as the whole and its boundaries, when calculating position, movement and interference constraint between *QCs*.

Furthermore, movements of cranes alongside quay may cause interference to neighboring vessel operations. Also, movement is limited by the length of crane rail tracks. For example, in a case when second order *QC* is engaged in the lower bays, the first order *QC* may be positioned far below operational zone. Therefore, travel paths limits are inserted in the model to avoid non-feasible solutions. Operational zone space for each *QC* restricts unnecessary *QC* movement and it is valuable information for resource planning in other terminal subsystems and in multipurpose terminal system environment, such as described in [4].

### 3 Model developments

Two models for QCSP optimization based on full task and split task algorithms were developed. There are common assumptions for both models:

- A1. Handling job demand is known for each vessel and expressed in number of containers for each bay-task.
- A2. Processing time for a single task is pre-calculated on the basis of average productivity rate records.
- A3. Hatch cover manipulation and preparation time are included in processing time.
- A4. Target jobs are grouped into clusters.
- A5. Tasks are marked successively from lower to higher number relative to vessel bays.
- A6. All cranes are available from the beginning to the end of the vessel service time.
- A7. All cranes have identical technical characteristics and performances.
- A8. Free movement of *QC* within the cluster has no impact on task processing time.

In addition both models take into consideration the following constraints and features:

- C1. Non-crossing and safety distance of *QCs* including space for idle cranes.
- C2. Zone and travel limits of *QCs* to cope with terminal space shortcomings.
- C3. Cluster lower and upper boundary.
- C4. Number of cranes available for a single vessel (for each crane-gang scenario optimization is provided).

#### 3.1 Full task algorithm for QCSP solution

This model assigns *QCs* to tasks where each task must be completed with the assigned crane. There are three sets of data:  $A = \{1, 2, \dots, a\}$  is set of tasks with index  $i \in A$ ,  $Q = \{1, 2, \dots, q\}$  is set of *QC* with index  $k \in Q$ , and  $R = \{(1, 2), 1, 3, \dots, (i, j)\} | i, j \in A, i \neq j$  set of task pairs defined according to Kim & Park [4]. If there are preceded tasks requirements additional set  $R'$  is required, such that  $R' = \{(i, j)\} | i, j \in A, i < j$  where  $R' \subset R$ .

Two binary decision variables are set up namely:

$$x_{ik} = \begin{cases} 1, & \text{if task } i \text{ is assigned to QC } k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$y_{ij} = \begin{cases} 1, & \text{if task } j \text{ starts after} \\ & \text{completion of task } i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Contrary to Kim & Park [6] who defined a 3-dimensional variable, two 2-dimensional binary variables

are set up similar to Lee et al. [11] to make the model easier to solve. Other model notations, input parameters and output variables are shown in Table 1. Note that the number of cranes  $q$  is different for each QCSP scenario. Usually, there is an agreement between shipping liners and a terminal operator to assure minimum number of QCs. The objective function minimizes total vessels processing time, earliest release and movement of QCs with the following expression:

**Table1** Model notations

Input parameters	Output variables
$a$	total number of handling task per vessel
$q$	number of cranes assigned to the vessel
$\omega_{1,2}$	weighted coefficients
$p_i$	processing time for task $i$
$p^{min}$	task partitioning limit
$h_i$	bay position of task $i$
$h_i^{mid}$	mid position of cluster task $i$ (bay)
$h_i^{lo}, h_i^{up}$	lower and upper cluster boundaries (bays)
$b$	time buffer
$d$	crane length overall (between bumpers)
$v$	travel speed of QC
$B^{lo}$	lower bound of crane path
$B^{up}$	upper bound of crane path
$M$	big integer
	$T$ handling makespan (last task finish time)
	$F_k$ completion of workload for QC $_k$
	$W_k$ total productive working time for QC $_k$
	$s_i$ start of task $i$
	$c_i$ completion of task $i$
	$x_{ik}$ task-to-crane binary variable assignment
	$y_{ij}$ consecutive tasks binary assignment
	$I_k^{move}$ total travel time of QC $_k$
	$s_{ik}$ start of task $i$ with QC $_k$
	$c_{ik}$ completion of task $i$ with QC $_k$
	$p_{ik}$ processing time of task $i$ with QC $_k$
	$Z$ objective function value

$$Min Z = \omega_1 T + \omega_2 \sum_{k \in Q} (F_k - W_k) + \omega_3 \sum_{k \in Q} I_k^{move} \quad (3)$$

subject to:

$$c_i \leq T \quad \forall i \in A \quad (4)$$

$$s_i + p_i = c_i \quad \forall i \in A \quad (5)$$

$$\sum_{k=1}^q x_{ik} = 1 \quad \forall i \in A \quad (6)$$

$$\sum_{i=1}^a p_i x_{ik} = W_k \quad \forall k \in Q \quad (7)$$

$$c_i - F_k \leq M(1 - x_{ik}) \quad \forall i \in A, \forall k \in Q \quad (8)$$

$$c_i + v^{-1} |h_j - h_i| - s_j \leq M [3 - (x_{ik} + x_{jk} + y_{ij})] \quad (9)$$

$$\forall i, j \in A, \forall k \in Q$$

$$c_i + b \leq c_j - p_j + M(1 - y_{ij}) \quad \forall i, j \in A \quad (10)$$

$$M y_{ij} + c_i + b > c_j - p_j \quad \forall i, j \in A \quad (11)$$

$$\sum_{k=1}^q k x_{ik} - \sum_{l=1}^l l x_{jl} + 1 \leq M(y_{ij} + y_{ji}) \quad (12)$$

$$\forall i, j \in A | h_i < h_j$$

$$d \left[ \sum_{l=2}^q l x_{jl} - \sum_{k=1}^{q-1} k x_{ik} \right] \leq (h_j^{lo} - h_i^{up}) + \quad (13)$$

$$+ M(y_{ij} + y_{ji}) \quad \forall i, j \in A | h_i^{mid} < h_j^{mid}$$

$$\psi_k^{up} \geq h_i^{up} x_{ik} \quad \forall i \in A, \forall k \in Q \quad (14)$$

$$M x_{ik} - M + \psi_k^{lo} \leq h_i^{lo} x_{ik} \quad \forall i \in A, \forall k \in Q \quad (15)$$

$$\psi_i^{lo} - \psi_k^{lo} \geq d(l - k) \quad \forall k, l \in Q | k < l \quad (16)$$

$$\psi_i^{up} - \psi_k^{up} \geq d(l - k) \quad \forall k, l \in Q | k < l \quad (17)$$

$$\psi_k^{lo} \geq B^{lo} \quad \forall k \in Q \quad (18)$$

$$\psi_k^{up} \leq B^{up} \quad \forall k \in Q \quad (19)$$

$$\psi_k^{up} - \psi_k^{lo} \leq v I_k^{move} \quad \forall k \in Q \quad (20)$$

$$y_{ij} + y_{ji} = 1 \quad \forall (i, j) \in R | h_i^{mid} = h_j^{mid} \quad (21)$$

$$c_i \leq c_j - p_j \quad \forall (i, j) \in R' \quad (22)$$

$$x_{ik}, y_{ij} \in \{0, 1\}, c_i \geq 0 \quad \forall i \in A \quad (23)$$

Objective function (3) uses handling makespan as the key criteria of QCSP modeling, similar in [10, 11] and [12] but also targets to eliminate postponed start of operations and to narrow the crane operational zone. That can be adjusted by weighted coefficients where, in normal circumstances,  $\omega_1 \gg \omega_2$  and  $\omega_1 \approx \omega_3$ . Makespan  $T = \max(c_1, \dots, c_a)$  is defined by  $c_i$  boundary in (4) and (23) in order to maintain linearity of the function. Relation between start and finish time is set in (5). Full task algorithm allows only one crane for each task to be assigned (6) while all processing times for the tasks assigned to the particular crane, equal crane workload (7). Similar to makespan definition, constraint (8) defines end of working time for the crane as completion time of the last task assigned to it. For tasks not assigned to the crane ( $x_{ik} = 0$ ) constraint is released because of big integer  $M$ .

Constraint (9) is actually modified solution presented in [6] but contrary to that, movement of cranes and time assignation is calculated according to comparison of successive tasks and value of binary variable  $y_{ij}$  as noted in [11]. If tasks are assigned to the same crane and precede each other ( $x_{ik} + x_{jk} + y_{ij} = 3$ ), next task should start after preceded task has been completed and after crane reaches the next position. Expression  $v^{-1} |h_j - h_i|$  is actually travel time between two tasks. Two constraints (10) and (11) determine values for binary variable  $y$ . Again, big integer  $M$  ensures that inequality is feasible for

any value of  $y$ . Note that time buffer  $b$  is added to cope with variability in task performance and to neutralize interference between cranes when moving between tasks.

Overtaking constraint (12) is taken from [11]. The lower crane cannot perform upper bay task while the upper crane performs lower bay task simultaneously ( $y_{ij} + y_{ji} = 0$ ). Similar to that, constraint (13) provides safety distance between cranes. It should be noted that safety distance has to be guaranteed for cluster boundaries; that is for neighboring cranes  $h_j^{lo} - h_i^{up} \geq d$  (see Fig. 3).

The group of constraints from (14) to (20) deals with operational and travel zone limits for QCs. Formulations (14) and (15) are slightly modified expression as in [7] and defines operational zone for each QC. Here is a point where two variables  $\psi_k^{lo}$  and  $\psi_k^{up}$  may be adjusted according to specific requirements of a terminal operator and possible infrastructure or system limitations. Quay limits are defined by the parameters  $B^{lo}$  and  $B^{up}$  in (18) and (19). For consistency of operational zones, safety distance issues are resolved by constraints (16) and (17). Finally, constraint (20) establishes relation between real movement of the crane and its operational zone limits.

Constraint (21) disables overlapping of two tasks (or two clusters) located at the same bay (middle bay) position. If there are conditional sequences of tasks, defined by the set  $R'$ , then constraint (22) must apply. Domains of the variables  $x$ ,  $y$  and  $c$  are set in (23).

### 3.2 Split task algorithm for QCSP solution

This model has some additional assumptions that should be noted:

First, there is no optimal solution guaranteed when the higher bay task must precede the lower bay task; consequently, there is no optimal solution at all if there is only one crane in the crane-gang.

Second, optimal solution may result in extensive partitioning with many short tasks that may lead to segregation of terminal resources in other terminal subsystem. The solution for that is setup of constant  $p^{min}$  with minimum acceptable process time partition.

The characteristic of the split task model is unidirectional movement of QCs; therefore conditional sequence of task pairs in proposed model is limited to the same bay/cluster task pairs only. Task break lines are determined by the model decision variables  $s_{ik}$  and  $c_{ik}$  marking start and completion times of the task  $i$  with the crane  $k$ . The objective function (3) has slight changes while the optimization goals remain the same, and reads:

$$MinZ = \omega_1 T + \omega_2 \sum_{\substack{i \in A \\ k \in Q}} (c_{ik} - p_{ik}) + \omega_3 \sum_{k \in Q} J_k^{move} \quad (24)$$

subject to:

$$c_{ik} \leq T \quad \forall i \in A, \forall k \in Q \quad (25)$$

$$s_{ik} + p_{ik} = c_{ik} \quad \forall i \in A, \forall k \in Q \quad (26)$$

$$\sum_{k=1}^q p_{ik} = p_i \quad \forall i \in A \quad (27)$$

$$p_{ik} \leq p_i x_{ik} \quad \forall i \in A, \forall k \in Q \quad (28)$$

$$x_{ik} \leq M p_{ik} \quad \forall i \in A, \forall k \in Q \quad (29)$$

$$p_{ik} \geq p^{min} x_{ik} \quad \forall i \in A, \forall k \in Q \quad (30)$$

$$c_{jk} - p_{jk} \geq c_{il} + b + v^{-1}(h_j^{mid} - h_i^{mid}) - M[2 - (x_{jk} + x_{il})] \quad (31)$$

$$\forall i, j \in A \mid h_i^{mid} \leq h_j^{mid}, \forall k, l \in Q \mid k < l$$

$$c_{ik} - p_{ik} \geq c_{jl} + b - M[2 - (x_{ik} + x_{jl})] \quad (32)$$

$$\forall i, j \in A \mid h_i^{mid} \leq h_j^{mid}, \forall k, l \in Q \mid k < l \wedge (h_j^{lo} - h_i^{up}) < d(l - k)$$

$$c_{jk} - p_{jk} \geq c_{ik} + b + v^{-1}(h_j^{mid} - h_i^{mid}) - M[2 - (x_{jk} + x_{ik})] \quad (33)$$

$$\forall i, j \in A \mid h_i^{mid} \leq h_j^{mid} \wedge i < j, \forall k \in Q$$

$$x_{ik} \in \{0, 1\}, c_{ik} \geq 0, p_{ik} \geq 0 \quad \forall i \in A, \forall k \in Q \quad (34)$$

Additionally, constraints (14) ÷ (20) from full task model algorithm have to be added.

Some changes in constraints modeling are necessary to make the model work, but in general the same set of constraints is used as in the full task algorithm. Firstly, constraint (6) needs to be released and constraints (7) and (8) too. Start and completion time for each subtask is calculated in (26) while whole task processing time must conform to the sum of subtasks for every task-crane assignment (27). The threshold for a subtask is determined by the duration of the basic tasks (28) and by the parameter  $p^{min}$  (30). Consistency of assigning tasks and subtasks to QCs is provided by constraints (27) and (28); when task is assigned to the crane ( $x_{ik} = 1$ ) then  $p_{ik} \neq 0$ , because of (29); if not ( $x_{ik} = 0$ ) then  $p_{ik} = 0$ , because of (28).

Constraints (31), (32) and (33) control the unidirectional movement of cranes in the execution of tasks in such a way as to prevent overtaking crane, ensure a safe distance and avoid simultaneous dual operation of cranes. They compare the starting and finishing times of task pairs  $(i, j)$ , where task  $i$  is positioned on a lower bay than task  $j$ , with position of two cranes  $(k, l)$ , where the crane  $k$  is below the crane  $l$ . If the higher task is assigned to the lower crane and lower task is assigned to the higher crane ( $x_{jk} + x_{il} = 2$ ), lower crane may move to its job only after the crane above completes its task and moves upwards; that is prevented by overtaking constraint (31). In the similar way, constraint (32) compares task pairs with smaller distance between the cranes than required,  $h_j^{lo} - h_i^{up} < d(l - k)$  and prevents their overlapping keeping the safe distance between cranes. Dual operation of the crane means simultaneous operations on two tasks/subtasks; that is unrealistic situation prevented by (33) with the same rule applied as in (31).

The rest of constraints are taken from full task model and explained earlier. Conditional tasks with beforehand determination of sequences are skipped from the model due to unidirectional cranes movement limitations in gaining optimal solutions. However, the exception is when pre-defined task sequence corresponds to the same bay/cluster numbers. In that case the model is capable to find an optimal solution.

### 4 Data generation and selection

For testing purposes the authors created representative set of data based on real vessels and cargo characteristics. So called "vessel-cargo generator" is developed, the programming script that is used to create container demand and distribution across the vessel bays. Vessel characteristics (Tab. 2) and container demand are based on footprint of Northern Adriatic ports with average container demand less than 1000 TEU per vessel call and maximum annual throughput up to 1 mil. TEU.

**Table 2** Characteristics of pattern vessels

Vessel	Range / Length (m)		Cap (TEU)	Bays	$h^{op}$
CMA Agadir	3	139	966	B1-B26	26
CMA Africa IV	4	228	3600	B1-B52	52
APL China	5	276	4832	B1-B62	62
CMA Bizet	6	300	6628	B1-B72	72
CMA Andromeda	7	363	11400	B1-B86	86

Design of the program script is based on the generation of pseudorandom numbers, where it is assumed that container positions across the bays follow the uniform distribution function. However, to make sure that the simulation works for small and medium-sized container terminals, some empirical indicators are used:

- the average containers/vessel demand that is much less than in the larger terminal;
- probability of vessel arrivals depend on size of the pattern vessels;
- there is much more inbound containers than outbound, that means less loadings on the same bay positions (released after cargo discharging);
- cargo distribution across the bays depends more on the route schedule, and is less consolidated in the targeted port area.

In total 100 arrivals or vessel-calls are generated with average container handling demand below 1000 TEU/vessel; demand per vessel can be noted from Tab. 3. Probability distribution of vessel arrivals with dependency on the vessel size is setup as follows: mid-size ships (range 5) 30 %, large ships (7) 10 %, while the rest of ships ranges (3,4,6) each with equal probability of arrival of 20 %.

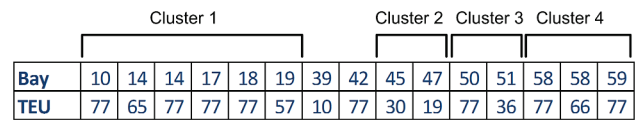
Next step in the data generation was placement of cargo, where uniform distribution is used with function notation  $U(a, b)$  where values for parameters  $a$  and  $b$  were adjusted to  $a = 8, b = 16$ , after series of experimentation. These two numbers define minimum and maximum number of different bay location for cargo stowage to be generated by the script. Note that final number of task  $a$  (see Tab. 3) depends on number of clusters, and that is in most cases less than number of different bay locations.

Depending on the probability distribution and the value of a random variable, selected bays may be loaded at full capacity or in a part. As a result the stowage plan is generated for each ship with containers allocated to corresponding bay positions. That is used as a basis for determination of tasks number and clusters boundaries.

Grouping of tasks into the cluster may be considered as the independent and separated optimization problem. However the purpose of clustering here is to test the

functionality of models, therefore the heuristics are developed for that purpose. The main criteria were the ability of simultaneous work of two or more cranes. The following logic is applied:

- even bay positions (40' compartments) and neighboring odd bay positions (20' compartments) are grouped together;
- odd bay positions on 40' compartments are grouped together too;
- positions with less than 7 containers in cells are attached to adjacent positions if it does not affect cluster interference criteria ( $>d$ );
- if the difference between the upper and lower bays in a group is less than  $d$ , and distance between the outmost bays in the group and next task in a range is greater than  $d$ , all positions in the group represent one cluster;
- if the difference between the upper and lower bays in a group is less than  $d$ , and distance between the outmost bays in the group and next task in a range is less than  $d$ , then positions are released until interference criteria ( $>d$ ) is satisfied.



**Figure 4** Grouping of tasks into clusters

The example in Fig. 4 shows how this heuristic is applied. For the vessel under number 43 there is 4 cluster tasks and 2 standalone tasks forming 6 tasks in total ( $a = 6$ ), with container demand in TEU for each position. In a similar manner tasks for other ships are designated.

### 5 Optimization results and analysis

It can be read from Fig. 2 how split task algorithm influences cranes utilization. When there is no task sharing (leftmost graph) crane 1 contributes with 20 time frames (e.g. minutes) or 18 % of total workload, while crane 2 contributes with 90 time frames or with 82 % of workload and makespan is 91. Capacity utilization of  $QC$ s may be measured by the crane utilization rate  $U_{iq}$  according to [12]:

$$U_{iq} = \frac{m_i}{C_{iq}} = \frac{m_i \cdot 60}{q \cdot QCSP_{iq}} \tag{35}$$

where  $m_i$  is handling demand in  $QC$  hours,  $C_{iq}$  is handling capacity, and  $QCSP_{iq}$  is the makespan  $T$  from the  $QCSP$  solution for the scenario with  $q$  cranes allocated to vessel  $i$  (note index  $i$  indicates a vessel in above expression).

For each of 100 vessel calls generated by the script, both  $QCSP$  optimization functions (3) and (24) were used. As a result, vessel makespan and crane utilization rate was obtained together with task to crane assignments, sequences of operations and operational zone range for each  $QC$ . Values for selected number of vessels are presented in Tab. 3.

For computation purposes, AIMMS modeling tool was used together with CPLEX 12.5 solver. Transport demand, given in the number of TEU (see Tab. 3) was



converted into the handling demand with parameter  $m$  and expressed in  $QC$  hours. For that purpose average productivity rate is set to 27 TEU/h and is assumed to be constant (see assumption A2).

Testing is performed for different number of  $QC$  allocations, from 1 to 5. For each such scenario, makespan

and crane utilization rates are compared relating to full or split task model algorithms. Where there was no improvement in makespan or utilization rate, such solutions are considered inefficient and are ignored. Therefore, only optimal QCSP solutions with reasonable number of  $QC$ s per vessel are considered further.

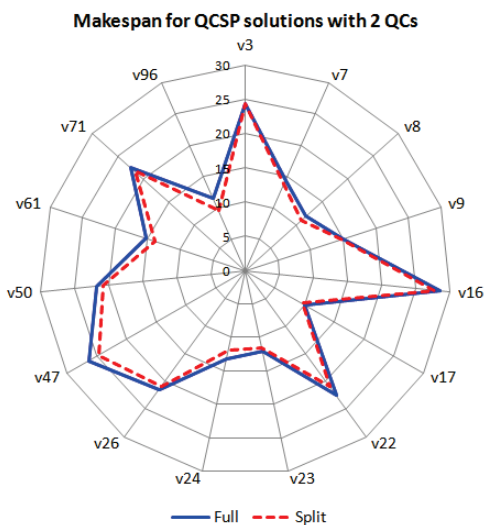
**Table 3** Comparison of QCSP solutions for selected vessels

Ship	TEU	$a$	$q$	Full		Split		Operation range (full task)				Operation range (split task)			
				$T_{iq}$	$U_{iq}$	$T_{iq}$	$U_{iq}$	$QC_1$	$QC_2$	$QC_3$	$QC_4$	$QC_1$	$QC_2$	$QC_3$	$QC_4$
v1	874	9	3	746	0.87	649	0.99	1-7	17-39	35-62	na	1-18	17-39	30-62	na
v7	741	8	3	695	0.79	636	0.86	2-23	26-35	42-51	na	2-23	22-35	42-51	na
v8	589	9	2	707	0.92	656	0.99	2-31	23-50	na	na	2-31	23-50	na	na
v15	1706	12	3	1413	0.89	1265	1.0	1-22	30-47	53-75	na	1-30	30-53	53-75	na
v16	1492	11	3	1196	0.92	1106	0.99	9-27	23-54	33-71	na	9-27	23-54	33-71	na
v18	1228	12	4	778	0.87	685	0.99	1-10	13-30	39-51	54-67	1-15	13-39	25-57	47-67
v20	2023	12	4	1259	0.89	1126	0.99	1-17	23-51	41-63	71-85	1-29	17-51	29-63	51-85
v22	1126	8	3	944	0.88	836	0.99	1-33	42-55	66-69	na	1-33	33-55	50-69	na
v24	638	7	3	633	0.75	481	0.98	1-9	11-19	33-49	na	1-13	11-37	33-49	na
v26	1123	10	3	914	0.91	833	1.0	11-22	29-50	43-62	na	11-30	29-47	43-62	na
v37	271	5	2	338	0.90	302	0.99	3-13	23-26	na	na	3-13	13-26	na	na
v38	1293	11	4	861	0.83	827	0.87	1-14	18-35	42-59	61-71	1-14	18-35	30-59	61-71
v46	729	8	3	605	0.89	549	0.98	1-18	30-37	43-62	na	1-31	14-43	37-62	na
v47	1335	9	4	981	0.75	912	0.81	1-13	25-27	35-41	49-70	1-13	26-35	39-53	49-70
v50	1126	7	2	1310	0.95	1253	0.99	2-22	39-71	na	na	2-39	39-71	na	na
v60	1048	10	4	630	0.92	613	0.95	5-13	17-29	42-53	62-70	5-13	17-29	29-53	62-70
v61	745	7	2	909	0.91	829	0.99	7-43	29-57	na	na	7-33	29-57	na	na
v68	902	9	3	763	0.87	669	0.99	5-15	23-35	45-58	na	5-15	15-35	33-58	na
v71	1165	8	3	941	0.92	864	1.0	14-23	43-51	54-70	na	14-43	43-54	54-70	na
v96	514	7	2	693	0.82	572	1.0	2-31	13-41	na	na	2-27	26-41	na	na

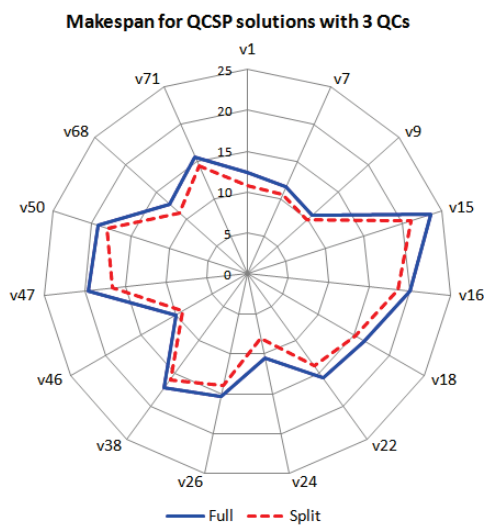
As a result, considering cargo stowage distribution and generated handling demand for targeted type of container terminals, there are options with 2, 3 or 4  $QC$  to be allocated for the particular vessel. To compare two developed models, the most representative option with high utilization rate and minimum time makespan was selected and presented in Tab. 3.

In 50 % cases there is no significant difference in makespan and/or crane utilization rate between two models. However, in all other cases there is minor or major deviation in favor of split task model. Major difference between the models and benefit of split task

algorithm is presented in graphs (Fig. 5 and 6) for selected vessels. Dotted lines in both graphs indicate makespan when QCSP is solved with the split task model. In general, the model is more effective for scenarios with 3 or more cranes than for 2 cranes, but that mainly depends on onboard cargo distribution. Significant difference in makespan and crane utilization rate is spotted for cases when larger handling demand was concentrated at the few central bays and more tasks with smaller handling demand aside. In such cases the split task model is the most useful.



**Figure 5** Handling makespan for 2  $QC$  scenario



**Figure 6** Handling makespan for 3  $QC$  scenario

**6 Conclusion**

Fixed number of  $QC$ s in container handling operations has an advantage because there is no dependences on

scheduled operations on adjacent berths, but this may lead to less efficient crane utilization. The implementation of split task algorithm may in many cases improve crane utilization rate for given set of  $QC$ s engaged in ship

handling process. This is especially important for small and mid-size terminals where pressure from vessel operators from one side and port facility limitation from the other side, influence the terminal operations.

The main limitation of split task approach is its inability in finding optimal solutions for preceding conditioned tasks; although that more concerns the terminals with greater and balanced loading and unloading demand, than terminals targeted in this paper.

Given results based on cargo demand generated by the script should be taken with caution, as the script produces data based on random choices of ship bays, which differs from real cargo stowage plans. However, dependences between onboard container stowage and *QC*s efficiency relating to the model choice are obvious and may be subject to further research and testing with data in real environment.

## 7 References

- [1] Murthy, K. G. et al. A decision support system for operations in a container terminal. // *Decision Support Systems*. 39, 3(2005), pp. 309-332. DOI: 10.1016/j.dss.2003.11.002
- [2] Steenken, D. et al. Container terminal operation and operations research: A classification and literature review. // *OR Spectrum*. 26, (2004), pp. 3-49. DOI: 10.1007/s00291-003-0157-z
- [3] Žuškin, S.; Grubišić, N.; Sumner, M. Ship-owner management in accordance with mutual agreement. // *Pomorstvo: Scientific Journal of Maritime Research*. 29, 1(2015), pp. 69-74.
- [4] Grubišić, N.; Hess, S.; Hess, M. A solution of berth allocation problem in inland waterway ports. // *Technical Gazette*. 21, 5(2014), pp. 1135-1141.
- [5] Daganzo, C. F. The crane scheduling problem. // *Transportation Research, Part B: Methodological*. 23, 3(1989), pp. 159-175. DOI: 10.1016/0191-2615(89)90001-5
- [6] Kim, K. H.; Park, Y. M. A crane scheduling method for port container terminals. // *European Journal of Operational Research*. 156, 3(2004), pp. 752-768. DOI: 10.1016/S0377-2217(03)00133-4
- [7] Liu, J. et al. Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. // *Naval Research Logistics (NRL)*. 53, 1(2005), pp. 60-74. DOI: 10.1002/nav.20108
- [8] Lim, A. et al. Crane scheduling with spatial constraints. // *Naval Research Logistics*. 51, 3(2004), pp. 386-406. DOI: 10.1002/nav.10123
- [9] Lim, A. et al. A m-parallel crane scheduling problem with a non-crossing constraints. // *Naval Research Logistics*. 54, 2(2007), pp. 115-127. DOI: 10.1002/nav.20189
- [10] Zhu, Y.; Lim, A. Crane scheduling with non-crossing constraint. // *Journal of the Operational Research Society*. 57, 12(2005), pp. 1464-1471. DOI: 10.1057/palgrave.jors.2602110
- [11] Lee, D. H. et al. Quay crane scheduling with non-interference constraints in port container terminals. // *Transportation Research Part E: Logistics and Transportation Review*. 44, 1(2008), pp. 124-135. DOI: 10.1016/j.tre.2006.08.001
- [12] Meisel, F. *Seaside Operations Planning in Container Terminals*. Springer, 2009. DOI: 10.1007/978-3-7908-2191-8
- [13] Meisel, F.; Bierwirth, C. Heuristics for the integration of crane productivity in the berth allocation problem. // *Transportation Research Part E: Logistics and Transportation Review*. 45, 1(2009), pp. 196-209. DOI: 10.1016/j.tre.2008.03.001
- [14] Lajjam, A. et al. Mathematical model for Quay Crane Scheduling Problem with spatial constraints. // *International Journal of Innovation and Applied Studies*. 4, 1(2013), pp. 547-551.
- [15] Grubisic, N.; Dundovic, C. Implementation of system engineering in port terminal planning. // *Pomorstvo*. 25, 1(2011), pp. 189-207.
- [16] Zhou, P.; Kang, H. Study on Berth and Quay-crane Allocation under Stochastic Environments in Container Terminal. // *Systems Engineering – Theory & Practice*. 28, (2008), pp. 161-169. DOI: 10.1016/S1874-8651(09)60001-6
- [17] Imai, A. et al. The simultaneous berth and quay crane allocation problem. // *Science Direct – Transportation Research Part E: Logistics and Transportation Review*, Elsevier. 44, (2008), pp. 900-920. DOI: 10.1016/j.tre.2007.03.003

### Authors' addresses

**Neven Grubišić, Ph.D., assistant professor**

Department of Maritime Transportation,  
Faculty of Maritime Studies, University of Rijeka,  
Studentska 2, HR-51000 Rijeka, Croatia  
E-mail: grubisic@pfri.hr

**Čedomir Dundović, Ph.D., full professor**

Department of Maritime Transportation,  
Faculty of Maritime Studies, University of Rijeka,  
Studentska 2, HR-51000 Rijeka, Croatia

**Srdan Žuškin, Ph.D., assistant**

Nautical department,  
Faculty of Maritime Studies, University of Rijeka,  
Studentska 2, HR-51000 Rijeka, Croatia