



Skipping the Storage Phase in Container Transshipment Operations

M. Flavia Monaco¹ and Marcello Sammarra²

¹ Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica,
Università della Calabria, Rende, Italy

monaco@dimes.unical.it

² Istituto di Calcolo e Reti ad Alte Prestazioni,
Consiglio Nazionale delle Ricerche, Rende, Italy

sammarra@icar.cnr.it

Abstract. The paper deals with the problem of scheduling the loading/discharging operations of two simultaneously berthed vessels, assuming that some of the containers discharged from a vessel must be directly loaded on the other one. For these containers also the stowage position must be decided. The aim is to minimize the time needed to complete all the operations required by the involved vessels. For this problem we present a mathematical model, a heuristic algorithm and discuss the computational results on a set of randomly generated instances.

Keywords: Container terminal · Direct transshipment · Stowage plan

1 Introduction

At a maritime terminal the conventional transshipment flow of containers follows the quay-yard-(yard)-quay cycle, where the yard-to-yard movements concern possible housekeeping operations aimed at reconfiguring the yard and recovering storage spaces. From the operative point of view, the storage of the containers in the yard is essential to decouple in time the ingoing and outgoing container flows. Therefore the discharging and loading operations (of the same containers) are independent and can be planned and scheduled separately and efficiently. On the other hand, the yard is a critical resource, due to its limited capacity, and terminal planners are concerned with the reduction of the sojourn time of the containers in the yard (dwell-time) because that could increase the terminal throughput. The dwell-times have also a relevant economic impact for the shipping operators, since they contribute to determine the port fees. Therefore, reducing the dwell-times is a common target for the two main operators of the transshipment market. In view of that, we investigate the feasibility of a new operative transshipment modality, called *Direct Transshipment*.

The research has been supported by Ministero dell'Istruzione, Università e Ricerca under the PRIN 2015 research program - Grant 2015XAPRKF - Smart PORT Terminals.

We consider two vessels, simultaneously berthed at not necessarily adjacent berths, and we assume that some of the containers discharged from each of them must be directly loaded into the other, while the rest of the cargo follows the conventional transshipment flow (quay-yard-quay). Clearly, in the direct transshipment modality the unloading and loading operations are no longer independent and the related scheduling processes are concurrent: each container to be directly transshipped represents two dependent tasks (unloading/loading), to be executed by different machines (quay cranes) operating on different vessels, linked by a strict precedence relationship. In order to fully take advantage from this operative modality, the stowage decisions for the directly transshipped containers must become a degree of freedom for the planners. This is to say that the stowage positions of such containers and the cranes scheduling will be determined concurrently. The Direct Container Transshipment Problem (DCTP) is then the problem of scheduling all the vessel operations while deciding the stowage positions for the containers to be directly transshipped, so as to minimize the overall service time of the vessels.

The paper is organized as follows. The scientific literature related to the DCTP is discussed in Sect. 2. In Sects. 3 and 4 we state and formulate the problem. The solution algorithm is presented in Sect. 5. Section 6 discusses the computational experience. Finally conclusions are drawn in Sect. 7.

2 Related Works

The direct transshipment of containers between vessels seems to be a relatively new problem in the literature concerning the management of container terminals. In a more wide research stream, it has some similarities with the Cross Docking policy at a distribution terminal in a logistic network [4]. The main analogy between them is the common need of synchronizing the arrival and departure sequences of the carriers, while having low or possibly zero inventory levels.

The effectiveness of the direct transshipment of containers between different means of transport has been investigated in [1] for the case of rail hubs, in [7] for the case of trains and trucks, and in [3] for the case of mother ships and barges. Only few papers refer to the direct ship-to-ship transshipment. In [8] the authors consider a one-way direct transshipment between a mother vessel and a feeder, which “moor at the same berth and utilize the same handling equipment”. Some researchers have recently studied an analogous problem in the case of mobile (or offshore) harbours, which are floating platforms equipped with portal cranes and are used to perform unloading and loading operations in the open sea (see [13]).

The only work that recognizes the effectiveness of the direct transshipment of containers between vessels is described in [15]. The authors present a model that integrates the berth allocation of the vessels, the yard space assignment, and the direct transshipment plan; the aim is to minimize the operative costs of trucks and yard cranes, as well as the delay cost of the vessels.

The DCTP we address in this paper, formerly introduced in [11], completely differs from that described in [15]; we deal with the operative management of the

direct transshipment of containers between vessels, so integrating the scheduling of the quay cranes allocated to the vessels and the stowage of the directly transshipped containers.

3 Problem Statement

In order to derive the mathematical model for the DCTP, we first need to introduce the adopted notation and to detail the complex interactions between the main decision components of the problem, from which the constraints originate. As noted before, the management of the direct transshipment operative modality calls for the integration of two decision processes: the scheduling of the cranes operating on the two vessels and the stowage of the containers to be directly transshipped. Therefore, we will model it as a Quay Crane Scheduling Problem QCSP (see e.g. [2, 14]) on a virtual vessel, with stowage constraints.

3.1 Notation

The mathematical model for the DCTP relies on the following main entities, that are vessels ($V = \{A, B\}$), tasks ($\Omega^v, v \in V$), and quay cranes ($Q^v, v \in V$). For each vessel v , the set of tasks consists of three disjoint subsets $\Omega^v = G^v \cup D^v \cup L^v$. G^v is the set of groups of containers, to be loaded or discharged, following the conventional transshipment flow, also called conventional tasks; D^v is the set of single containers discharged from v and directly transshipped to the other vessel; L^v is the set of containers directly transshipped from the other vessel and to be loaded into v . In the following we will refer to a container to be directly transshipped as a DT container. The vessels V can be seen as a single virtual vessel, with task set $\Omega = \Omega^A \cup \Omega^B$ and crane set $Q = Q^A \cup Q^B$.

In Table 1 we detail the characteristics and the attributes of vessels, tasks, and cranes.

Some of the entries in Table 1 need to be further explained. Each element θ of the sets Θ^v actually represents a single stowage slot in terms of (*bay, row, tier*) coordinates; moreover each $\theta \in \Theta^v$ is able to accommodate only a subset of the containers in L^v , as indicated by the class-based stowage plan (see [12]). Therefore we define by $\Theta_i^v \subseteq \Theta^v, i \in L^v$ the set of slots where container i can be stowed, and by $H_i^v \subset H^v$ the set of bays where a stowage slot compatible with the container $i \in L^v$ is located, that is $H_i^v = \{b \in H^v \mid b = b_\theta, \theta \in \Theta_i^v\}$, where b_θ is the bay coordinate corresponding to θ . In passing, we observe that H_i^v can also be defined for tasks $i \in G^v \cup D^v$, being in this case $H_i^v = \{b(i)\}$ a singleton. Finally, we define $\Theta^v(b, i)$ the set of slots $\theta \in \Theta_i^v$ compatible with i and located in the bay b .

Due to physical restrictions, there are tasks whose processing can not overlap in time. For example, in the same bay unloading tasks always precede the loading ones, while in adjacent bays simultaneous processing of tasks by different cranes is forbidden, due to safety issues. This is to say that, for each vessel, some temporal restrictions on task pairs are defined. They impose either precedence

Table 1. Notation

Vessels	
a^v	Arrival time of vessel $v \in V$
Θ^v	Set of slots available for stowing containers directly transshipped to the vessel $v \in V$
H^v	Set of bays of vessel $v \in V$
Tasks	
p_i	Processing time of task $i \in \Omega^v$
$b(i)$	Bay coordinate of task $i \in \Omega^v$
Cranes	
r_k	Release time of crane $k \in Q^v$
$[s_k, f_k]$	Set of adjacent bays where crane k is allowed to operate
l_k^0	Initial bay-position of crane k ($s_k \leq l_k^0 \leq f_k$)
\hat{t}	Time a crane takes to travel from a bay to an adjacent one
$t_{b_1 b_2}$	Time a crane takes to travel between two generic bays: $t_{b_1 b_2} = \hat{t} b_1 - b_2 $
t_{kb}^0	Time needed to the crane k to reach the bay b from its starting bay-position: $t_{kb}^0 = \hat{t} l_k^0 - b $
$Q(b)$	The set of cranes k such that $b \in [s_k, f_k]$

or non-simultaneity constraints on the processing of the tasks. More clearly, if there is a precedence between tasks i and j , then the processing of j can not start before the processing of i has been completed. Conversely, a non-simultaneity relationship between i and j imposes that either i must precede j , or j must precede i . To model these relations we need to extend our notation.

Precedence Relationships: A first set of precedence relationships is given by

$$\Phi = \bigcup_{v \in V} \Phi^v \cup \bar{\Phi}$$

where

$$\Phi^v = \{(i, j) \mid i \rightarrow j, i, j \in G^v \cup D^v\} \quad v \in V$$

relates to pairs of tasks belonging to the same vessel. Φ^v basically expresses precedences due to the operations the tasks require and can be populated using the stowage plan. Conversely

$$\bar{\Phi} = \{(i, j) \mid i \rightarrow j, (i \in D^A, j \in L^B) \vee (i \in D^B, j \in L^A)\}$$

are the precedence relationships between discharging and loading operations on different vessels for the DT containers.

Further precedence relationships for a given vessel are needed to guarantee that the directly transshipped containers will be loaded according to the stowage plan. To this aim we define

$$\begin{aligned}\Phi_1^v &= \{(i, \theta) \mid i \rightarrow \theta, i \in G^v \cup D^v, \theta \in \Theta^v\} & v \in V \\ \Phi_2^v &= \{(\theta, i) \mid \theta \rightarrow i, \theta \in \Theta^v, i \in G^v\} & v \in V \\ \Phi_3^v &= \{(\theta_1, \theta_2) \mid \theta_1 \rightarrow \theta_2, \theta_1, \theta_2 \in \Theta^v\} & v \in V\end{aligned}$$

The sets Φ_1^v and Φ_2^v induce precedence relationships between a task whose stowage position is known and the container that will be stowed into the ship-slot θ . The sets Φ_3^v define precedence relationships between pairs of ship-slots due to their relative positions within the same bay. As a consequence they will induce, at runtime, also a set of precedence relations on the containers that will be stowed there.

Non Simultaneity Relationships: Assumed δ to be the safety distance between two adjacent cranes, expressed in number of bays, we define the set of bays that cannot be operated simultaneously by different cranes as follows

$$\Psi^v = \{(b_1, b_2) \mid b_1, b_2 \in H^v, b_1 < b_2, b_2 - b_1 \leq \delta\} \quad v \in V$$

The above sets allow to impose the non-simultaneity constraints between each pair of tasks in close bays, even for the loading ones whose stowage bay must be decided by the model. However, as disclosed in [2], the sets Ψ^v are not sufficient to model the interferences between non adjacent cranes working on the vessel v . To this aim we define

$$\Delta_{b_1 b_2}^{hk}(v) = \max \{ \hat{t}((\delta + 1)(k - h) - (b_2 - b_1)), 0 \} \quad b_1, b_2 \in H^v, h, k \in Q^v, v \in V$$

as the minimum time to elapse between the processing of any task located in the bay b_1 and any task in bay b_2 by cranes h and k , respectively. Therefore

$$\hat{\Delta}(v) = \{(b_1, b_2, h, k) \mid \Delta_{b_1 b_2}^{hk}(v) > 0\}$$

is the set of all combinations of bays and cranes that cause interferences on the vessel v . Observe that if $(b_1, b_2) \in \Psi^v$, then $(b_1, b_2, h, k) \in \hat{\Delta}(v)$ for all cranes $h, k \in Q^v$ such that $h \in Q(b_1), k \in Q(b_2)$. Thus, $\hat{\Delta}(v)$ is a generalization of Ψ^v and extends the corresponding definition in [2].

3.2 The Mathematical Model

The DCTP model involves both binary and continuous variables, as detailed in Table 2. Note that, as $H_i^v = \{b(i)\}$, then $\alpha_{ib(i)} = 1, \alpha_{ib} = 0 \forall b \neq b(i)$ are input data for all tasks $i \in G^v \cup D^v, v \in V$.

The constraints to be imposed for each vessel $v \in V$ can be stated as follows:

Table 2. Variables

Continuous variables	
$c_i \geq 0$	Completion time of task $i \in \Omega^v, v \in V$
$\sigma_{ij} \geq 0$	Transfer time of a DT container from the discharging to the loading bay, $(i, j) \in \bar{\Phi}$
$w^v \geq 0$	Makespan of vessel $v \in V$
$w = \max_{v \in V} w^v$	The makespan of the virtual vessel
Binary variables	
$x_{ijk} = 1$	If tasks $i, j \in \Omega^v$ are performed consecutively by crane $k, v \in V$
$z_{ij} = 1$	If task $i \in \Omega^v$ is completed before the processing of task $j \in \Omega^v$ starts, $v \in V$
$y_{i\theta} = 1$	If container $i \in L^v$ is stowed in the slot $\theta \in \Theta_i^v, v \in V$
$\alpha_{ib} = 1$	If task $i \in \Omega^v$ must be handled in a bay $b \in H_i^v, v \in V$

Crane Routing Constraints: Constraints (1) to (4) define the sequence of tasks performed by each crane. Note that 0 and T are dummy tasks with $p_0 = p_T = 0$, $\Omega_0^v = \Omega^v \cup \{0\}$, $\Omega_T^v = \Omega^v \cup \{T\}$ and $x_{0Tk} = 1$ corresponds to an empty sequence for crane k .

$$\sum_{j \in \Omega_0^v} x_{0jk} = 1 \quad k \in Q^v \quad (1)$$

$$\sum_{i \in \Omega_0^v} x_{iT k} = 1 \quad k \in Q^v \quad (2)$$

$$\sum_{k \in Q^v} \sum_{j \in \Omega_T^v} x_{ijk} = 1 \quad i \in \Omega^v \quad (3)$$

$$\sum_{j \in \Omega_T^v} x_{ijk} - \sum_{j \in \Omega_0^v} x_{jik} = 0 \quad i \in \Omega^v, k \in Q^v \quad (4)$$

Stowage Constraints: Constraints (5) and (6) assign a stowage position, in terms of slot and bay, to the DT containers, while constraints (7) and (8) state the relationships between y 's and α 's, x 's and α 's variables, respectively. Note that, for a fixed $i \in L^v$, summing up constraints (7) on the compatible bays $b \in H_i^v$ and taking into account constraints (6), one gets $\sum_{\theta \in \Theta_i^v} y_{i\theta} = 1$. Constraints (8) impose that if a task must be performed in a bay b , it has to be assigned to a crane able to operate that bay.

$$\sum_{i \in L^v} y_{i\theta} = 1 \quad \theta \in \Theta^v \quad (5)$$

$$\sum_{b \in H_i^v} \alpha_{ib} = 1 \quad i \in L^v \quad (6)$$

$$\sum_{\theta \in \Theta(i,b)} y_{i\theta} = \alpha_{ib} \quad i \in L^v, b \in H_i^v \quad (7)$$

$$\sum_{k \in Q(b)} \sum_{j \in \Omega^v} x_{ijk} \geq \alpha_{ib} \quad i \in \Omega^v, b \in H_i^v \quad (8)$$

Completion Time Constraints: The completion times of the tasks are computed through constraints (9)–(12). Here and in what follows M is a big constant.

$$c_i - p_i \geq a^v \quad i \in \Omega^v \quad (9)$$

$$r_k - c_j + \sum_{b \in H_j^v} \alpha_{jb} t_{l_k^v b} + p_j \leq M(1 - x_{0jk}) \quad j \in \Omega^v, k \in Q^v \quad (10)$$

$$c_i - c_T \leq M(1 - x_{iT_k}) \quad i \in \Omega^v, k \in Q^v \quad (11)$$

$$c_i + \hat{t} \left| \sum_{b \in H_i^v} b\alpha_{ib} - \sum_{b \in H_j^v} b\alpha_{jb} \right| + p_j - c_j \leq M(1 - x_{ijk}) \quad i, j \in \Omega^v, k \in Q^v \quad (12)$$

The non linearity in constraints (12) can be easily handled replacing each of them with the following set of constraints

$$c_i + (\alpha_{ib_1} + \alpha_{jb_2} - 1)t_{b_1 b_2} + p_j - c_j \leq M(1 - x_{ijk}) \quad b_1 \in H_i^v, b_2 \in H_j^v \quad (13)$$

Precedence Constraints: Constraints (14)–(18) impose the precedence relationships between pairs of tasks.

$$c_i + p_j - c_j \leq 0 \quad (i, j) \in \Phi^v \quad (14)$$

$$c_i + p_j - c_j \leq M(1 - y_{j\theta}) \quad j \in L^v, \theta \in \Theta_j^v, (i, \theta) \in \Phi_1^v \quad (15)$$

$$c_j + p_i - c_i \leq M(1 - y_{j\theta}) \quad j \in L^v, \theta \in \Theta_j^v, (\theta, i) \in \Phi_2^v \quad (16)$$

$$c_i + p_j - c_j \leq M(2 - y_{i\theta_1} - y_{j\theta_2}) \quad i, j \in L^v, \theta_1 \in \Theta_i^v, \theta_2 \in \Theta_j^v, (\theta_1, \theta_2) \in \Phi_3^v \quad (17)$$

$$c_i + \sigma_{ij} + p_j - c_j \leq 0 \quad (i, j) \in \bar{\Phi} \quad (18)$$

In particular, (14) define the precedence between tasks whose stowage position is known, while (15) to (18) take into account precedence relationships involving DT containers to be loaded. The variables σ_{ij} in (18) are defined through equations (19), where l_A^F is the last bay of vessel A, d_{AB} is the inter-vessel distance expressed in number of bays, and τ is the time a straddle carrier takes to cover a ship-bay (see [5]).

$$\sigma_{ij} = \tau \left(\sum_{v \in V} \sum_{b \in H_j^v} b\alpha_{jb} - b(i) + l_A^F + d_{AB} \right) \quad (i, j) \in \bar{\Phi} \quad (19)$$

Non Simultaneity Constraints: The relations between the completion times of the tasks and the z 's variables are stated by constraints (20), (21), and (22). They impose a partial time-ordering on the tasks of the same vessel. Actually, for each pair of tasks, either i precedes j ($z_{ij} = 1, z_{ji} = 0$), or j precedes i ($z_{ji} = 1, z_{ij} = 0$), or, finally, the processing of i and j overlap ($z_{ij} = z_{ji} = 0$). Note that if i and j are tasks located in too close bays, constraints (22) avoid that they are processed simultaneously.

$$c_i + p_j - c_j \leq M(1 - z_{ij}) \quad i, j \in \Omega^v \quad (20)$$

$$c_j - p_j - c_i \leq Mz_{ij} \quad i, j \in \Omega^v \quad (21)$$

$$z_{ij} + z_{ji} \geq \alpha_{ib_1} + \alpha_{jb_2} - 1 \quad i, j \in \Omega^v, b_1 \in H_i^v, b_2 \in H_j^v, (b_1, b_2) \in \Psi^v \quad (22)$$

Non Interference Constraints: For each pair of tasks $i, j \in \Omega^v$ and for each pair of compatible bays $b_1 \in H_i^v, b_2 \in H_j^v$, the following constraints must hold for each pair of cranes h and k that would cause interference working simultaneously on bays b_1 and b_2 , that is $(b_1, b_2, h, k) \in \hat{\Delta}(v)$:

$$\sum_{u \in \Omega_0^v} x_{uih} + \sum_{u \in \Omega_0^v} x_{ujk} + \alpha_{ib_1} + \alpha_{jb_2} \leq 3 + z_{ij} + z_{ji} \quad (23)$$

$$c_i + \Delta_{b_1 b_2}^{hk}(v) + p_j - c_j \leq M \left(5 - \alpha_{ib_1} - \alpha_{jb_2} - z_{ij} - \sum_{u \in \Omega_0^v} x_{uih} - \sum_{u \in \Omega_0^v} x_{ujk} \right) \quad (24)$$

$$c_j + \Delta_{b_1 b_2}^{hk}(v) + p_i - c_i \leq M \left(5 - \alpha_{ib_1} - \alpha_{jb_2} - z_{ji} - \sum_{u \in \Omega_0^v} x_{uih} - \sum_{u \in \Omega_0^v} x_{ujk} \right) \quad (25)$$

Objective Function Definition. The objective function to be minimized is a linear combination of two conflicting functions: the makespan of the virtual vessel, defined by (26)–(27), and the average waiting time for the DT containers (28).

$$c_i \leq w^v \quad i \in \Omega^v, v \in V \quad (26)$$

$$w^v \leq w \quad v \in V \quad (27)$$

$$\min \lambda w + \mu \frac{1}{|\bar{\Phi}|} \sum_{(i,j) \in \bar{\Phi}} (c_j - p_j - \sigma_{ij} - c_i) \quad (28)$$

4 Refinement of the DCTP Model

As motivated in [2,10], the search of feasible solutions of the QCSP can be limited to the unidirectional schedules, where all the cranes move from the bow to the stern of the vessel, or in the opposite direction. Therefore, also in the DCTP model (1)–(11), (13)–(28) it is possible to impose the one-way movement

of all the cranes allocated to the single vessels. To this aim we introduce two new binary variables: $\gamma^v = 1$ if the cranes in Q^v move from the bow to the stern, $v \in V$, and a set of additional constraints (see [10]):

$$\sum_{b \in H_i^v} b\alpha_{ib} - \sum_{b \in H_j^v} b\alpha_{jb} \leq M(1 - x_{ijk}) + M(1 - \gamma^v) \quad i, j \in \Omega^v, v \in V \quad (29)$$

$$\sum_{b \in H_j^v} b\alpha_{jb} - \sum_{b \in H_i^v} b\alpha_{ib} \leq M(1 - x_{ijk}) + M\gamma^v \quad i, j \in \Omega^v, v \in V \quad (30)$$

Note that (29), (30) extend the corresponding constraints (16), (17) in [10], also to the DT containers to be loaded, whose stowage bay is unknown.

5 Solution Algorithm

The formulation of the DCTP (1)–(11), (13)–(30) as a QCSP on a virtual vessel with side constraints naturally drives to design a solution algorithm by suitably modifying the Tabu Search Algorithm for the QCSP described in [10].

Given a feasible stowage plan for the DT containers, our algorithm iterates over feasible solutions constructed by a two-phases approach:

1. *Routing phase*: for each crane, a feasible sequence of tasks is determined taking into account precedence, one-way and cranes' operative range constraints.
2. *Scheduling phase*: the completion time of the tasks is computed imposing the non simultaneity and non interference constraints, and the precedence constraints related to the DT containers.

A feasible schedule for the cranes can be represented by a disjunctive graph with node set $\Omega \cup \{0, T\}$ (see [10]), where disjunctive edges model the non simultaneity and the non interference constraints. To perform the scheduling phase and evaluate the makespan for the virtual vessel, we have to find the critical path from 0 to T on such a disjunctive graph. This problem is, in general, \mathcal{NP} -hard, while in our case it becomes easier to solve. The one-way assumption, in fact, uniquely identify the orientation of the disjunctive edges giving rise to an acyclic graph.

To describe the Tabu Search algorithm for the DCTP, it is sufficient to specify the memory mechanism and the neighbourhood structure. We adopt the attributive memory mechanism, meaning that a solution is declared tabu if at least one of the attributes describing that solution is tabu [6]. In order to introduce the neighbourhood structure, let us denote by (\bar{x}, \bar{y}) a given feasible solution in terms of the main scheduling variables (x) and stowage variables (y). We define *swap move* the swapping of the stowage positions of two containers in L^v , $v \in V$; $N_1(\bar{y})$ is the set of all feasible stowage configurations obtained from \bar{y} by performing a swap move. Furthermore, we define *shift move* the shifting of a task currently assigned to the crane k to an adjacent crane ($k - 1$ or $k + 1$), and $N_2(\bar{x})$ as the set of all feasible schedules obtained from \bar{x} by performing a shift move. The neighborhood of (\bar{x}, \bar{y}) can now be defined as follows:

$$\mathcal{N}(\bar{x}, \bar{y}) = \{(\bar{x}, y) \mid y \in N_1(\bar{y})\} \cup \{(x, \bar{y}) \mid x \in N_2(\bar{x})\} \quad (31)$$

6 Computational Experience

The Tabu Search Algorithm (TSA) has been implemented in C++. The stopping criterion is based on a maximum number of iterations equal to 2000. The tabu tenure has been set to 15 iterations; the diversification penalty has been set equal to 0.05. The tests have been carried out on a machine equipped with a 3.1 GHz Intel Core i5 CPU and 16 GB of RAM. TSA has been tested on a set of instances randomly generated as described in the next subsection.

6.1 Instance Generator Algorithm

Let I an instance of the standard QCSP defined by: number of conventional tasks (T), number of cranes (Q), and number of bays (B). In such an instance each conventional task is characterized by a processing time p and a bay location b . Given two QCSP instances, say I_A and I_B , called *seed* instances, an instance I_{AB} of the DCTP can be constructed as follows. First, we assume that a task i of I_A or I_B with processing time p_i is a group of p_i containers. Let c be the number of container classes and n_c the number of DT containers of class c in the instance I_{AB} . For each instance I_A and I_B , apply the following algorithm (DCTP-G):

1. Randomly select a class c and a bay b .
2. Randomly select in the bay b a conventional task i of class c ; let p_i its processing time.
3. If $n_c \leq p_i$, replace the task i with $n_c + 1$ tasks, where the first tasks represent n_c DT containers, while the last one, if any, represents a residual conventional task i' with processing time $p_{i'} = p_i - n_c$. *GO TO* 1.
4. If $n_c > p_i$, replace the selected conventional task i with p_i DT containers; set $n_c = n_c - p_i$, $b = b + 1$ and *GO TO* 2.

To generate the seed instances, we have adopted the instance generator *QCSPgen* developed by Meisel and Bierwirth in [9]. The interested reader is referred to [9] for more details on the *QCSPgen* algorithm. Here we just mention that, among the input parameters of *QCSPgen*, we have set the distribution of the tasks within the vessel to be uniform; the density of precedence relationships among tasks of the same bay to be one, meaning that within the same bay all the tasks are sorted to reflect the stowage constraints. Finally we have set the crane safety distance to be one bay. We have generated two sets of instances to represent two kind of vessels: mother vessels and feeder vessels. The dimensions of the seed instances are reported in Table 3, where *NoI* indicates the number of instances of each type generated.

The seed-instances of Table 3 are combined each other and become the input for the DCTP-G algorithm, either as I_A or as I_B , together with the number of container classes c and the number of DT containers per class, n_c , giving rise to 33 DCTP instances as detailed in Table 4.

Table 3. Dimensions of the seed instances.

Type	Bays	Tasks	Cranes	NoI
Mother	20	10	3	1
Mother	20	15	3	1
Mother	20	20	4	1
Feeder	10	10	2	5

Table 4. Description of the DCTP instances.

Code	I_A	I_B	c	$\sum_c n_c$	NoI	Direct transshipment flow
MF	Mother	Feeder	2	240	15	$A \rightarrow B$
FM	Feeder	Mother	2	150	15	$A \rightarrow B$
MM	Mother	Mother	2	250	3	$A \rightarrow B$ (115), $A \leftarrow B$ (125)

6.2 Lower Bounds for the DCTP

In order to evaluate the effectiveness of TSA, we need to compute lower bounds for the DCTP. The most natural way to achieve this aim is to consider a relaxed DCTP model obtained by removing constraints (5) to (8) related to the stowage decision. The resulting DCTP-SR relaxed problem, consisting of two standard QCSPs linked by constraints (18), is hard to solve to optimality. Actually standard ILP solvers easily run out of memory, due to the high number of constraints and variables. For these reasons we also relax constraints (18), getting the DCTP-WR problem that decomposes in two QCSPs no longer dependent on each other, and relatively easy to be solved by a standard ILP solver.

6.3 Analysis of the Results

In the following Tables 5, 6, and 7, we summarize the computational results obtained by solving the instances described in Table 4 by our TSA, and the corresponding DCTP-WR problems by ILOG Cplex 12.6.2., setting $\lambda = \mu = 1$ in the objective function. For each instance, in the leftmost columns of the result Tables we report: the data related to the number of bays and cranes (B-Q) of each vessel; the number of tasks T in the form (DT containers - conventional tasks). Observe that the instance code $I_A I_B - \alpha - \beta - \gamma - \delta$ is useful to recognize the seed instances generating it, being $\alpha - \beta$ the number of bays and tasks of I_A and $\gamma - \delta$ the number of bays and tasks of I_B . Then we report the makespan of the two vessels, both for the DCTP-WR and DCTP models, marking in bold the maximum between them. As for the DT containers, in the second last column (AvgD) we report the average waiting time between the discharging and the loading operation. Finally, in the last column, we report the gap computed

Table 5. Results on Mother-Feeder instances.

Code	Vessel A		Vessel B		LB		UB			Gap%
	$B - Q$	T	$B - Q$	T	w^A	w^B	w^A	w^B	AvgD	
MF-20-10-10-10-1	20-3	240-10	10-2	240-6	1482	259	1530	1328	0.00	3.24
MF-20-10-10-10-2	20-3	240-10	10-2	240-6	1482	266	1482	1385	26.29	1.77
MF-20-10-10-10-3	20-3	240-10	10-2	240-7	1482	267	1484	1392	101.40	6.98
MF-20-10-10-10-4	20-3	240-10	10-2	240-5	1482	267	1482	1376	4.17	0.28
MF-20-10-10-10-5	20-3	240-10	10-2	240-6	1482	300	1482	1425	13.83	0.93
MF-20-15-10-10-1	20-3	240-13	10-2	240-7	1482	259	1485	555	77.54	5.43
MF-20-15-10-10-2	20-3	240-13	10-2	240-3	1482	266	1485	555	31.67	2.34
MF-20-15-10-10-3	20-3	240-13	10-2	240-5	1482	277	1482	543	59.23	4.00
MF-20-15-10-10-4	20-3	240-13	10-2	240-6	1482	284	1482	523	24.43	1.65
MF-20-15-10-10-5	20-3	240-13	10-2	240-7	1482	300	1485	580	60.14	4.26
MF-20-20-10-10-1	20-4	240-20	10-2	240-6	1926	259	1926	817	21.50	1.12
MF-20-20-10-10-2	20-4	240-20	10-2	240-6	1926	266	1926	759	26.29	1.37
MF-20-20-10-10-3	20-4	240-20	10-2	240-7	1926	277	1926	768	55.93	2.90
MF-20-20-10-10-4	20-4	240-20	10-2	240-5	1926	267	1926	750	4.17	0.22
MF-20-20-10-10-5	20-4	240-20	10-2	240-6	1926	300	1926	799	13.83	0.72

Table 6. Results on Feeder-Mother instances.

Code	Vessel A		Vessel B		LB		UB			Gap%
	$B - Q$	T	$B - Q$	T	w^A	w^B	w^A	w^B	AvgD	
FM-10-10-20-10-1	10-2	150-8	20-3	150-10	259	1457	373	1607	358.99	34.93
FM-10-10-20-10-2	10-2	150-8	20-3	150-10	266	1457	277	1607	325.00	32.60
FM-10-10-20-10-3	10-2	150-8	20-3	150-10	277	1457	295	1607	319.41	32.22
FM-10-10-20-10-4	10-2	150-6	20-3	150-10	273	1606	273	1606	534.82	33.30
FM-10-10-20-10-5	10-2	150-8	20-3	150-10	300	1457	335	1607	338.03	33.50
FM-10-10-20-15-1	10-2	150-8	20-3	150-15	259	1471	425	1616	413.81	37.99
FM-10-10-20-15-2	10-2	150-8	20-3	150-15	266	1471	277	1640	367.69	36.48
FM-10-10-20-15-3	10-2	150-8	20-3	150-15	277	1471	295	1640	362.10	36.10
FM-10-10-20-15-4	10-2	150-6	20-3	150-15	273	1414	273	1532	144.85	18.59
FM-10-10-20-15-5	10-2	150-8	20-3	150-15	300	1471	335	1640	380.72	37.37
FM-10-10-20-20-1	10-2	150-8	20-4	150-18	259	1926	425	1926	87.93	4.57
FM-10-10-20-20-2	10-2	150-8	20-4	150-18	266	1926	266	1926	57.54	2.99
FM-10-10-20-20-3	10-2	150-8	20-4	150-18	277	1926	295	1926	63.68	3.31
FM-10-10-20-20-4	10-2	150-6	20-4	150-19	273	1926	284	1926	35.54	1.85
FM-10-10-20-20-5	10-2	150-8	20-4	150-18	300	1926	335	1926	76.65	3.98

Table 7. Results on Mother-Mother instances.

Code	Vessel A		Vessel B		LB		UB			Gap%
	$B - Q$	T	$B - Q$	T	w^A	w^B	w^A	w^B	AvgD	
MM-20-10-20-15-1	20-3	240-10	20-3	240-14	1482	1482	1482	1600	16.10	9.05
MM-20-10-20-20-2	20-3	240-10	20-4	240-20	1482	1926	1607	1927	158.04	8.20
MM-20-15-20-20-3	20-3	240-13	20-4	240-18	1482	1926	1972	1926	0.67	2.42

as $Gap\% = 100 \times (UB - LB)/LB$ where $LB = \max\{w^A, w^B\}$ and $UB = \max\{w^A, w^B\} + AvgD$. We do not report the computation times, since they are about 300s almost uniformly on all the instances.

A first oversight to the Tables shows that TSA gives satisfactory results on all but the first two groups of instances in Table 6. Actually, very often the overall makespan coincides with its lower bound, and the average delay of the DT containers is small enough, resulting in a gap that does not exceed 7% for the M-F instances, 5% for the last group of F-M instances, and 10% for the M-M instances. In particular, the results obtained on the M-F instances (Table 5) can be motivated as follows: the makespan of the virtual vessel is always attained at the mother vessel, therefore the delay of the DT containers can be reduced by, eventually, letting the cranes working on the feeder vessel wait. This is because the makespan of the feeder can increase without affecting the objective function. This phenomenon is especially evident in the first and fourth instances of the first group, and in the fourth instance of the last group in Table 5.

But the delay of the DT containers could be unavoidable. In fact, it is strictly related to their class and to the class-based stowage plan of both vessels, that is to the data set of the seed instances and to the random procedure implemented by DCTP-G, whose output could likely be an instance not enough suitable for the Direct Transshipment modality. Looking, for example, at the third instance of the first group in Table 5, we observe that the average delay of DT containers is relatively high, in spite of the remarkable growth of the makespan of the feeder vessel. As for the F-M instances, most of them seem to belong to the class of instances for which the Direct Transshipment is not a convenient approach. Actually, even for the last (end best) group of instances in Table 6, we can observe that the delay of the DT containers is unavoidable. This is very clear for the instance FM-10-10-20-20-2, where the makespan of both vessels is computed in an optimal way.

The results in Table 7, related to the bi-directional flow of DT containers between mother vessels, exhibit both the characteristics discussed before. The minimum value of AvgD is attained at the third instance, for which the makespan of vessel A, that was the minimum in the relaxed problem, grows so much to become the makespan of the virtual vessel in DCTP. In the second instance, instead, it is evident that the waiting time for the DT containers can not be reduced any further.

7 Conclusions

In this paper we have addressed the Direct Container Transshipment Problem (DCTP), that is the problem of scheduling the loading/discharging operations of two vessels sharing the berthing time windows, and assuming that some containers discharged from a vessel must be directly loaded on the other one, completely skipping the storage phase in the yard. The aim is to minimize a linear combination of the time needed to complete all the operations and the average waiting time for the directly transshipped containers. The DCTP integrates two operative decision processes: the scheduling of the quay cranes and the stowage of

the container directly transshipped. For this problem we have described a mixed integer linear model and we have derived a Tabu Search heuristic algorithm. We have tested the heuristic algorithm on a set of randomly generated instances. The algorithm is able to find feasible solutions of good quality in almost all the considered instances within a short amount of computation time, despite the intrinsic hardness of the problem.

The DCTP generalizes the Quay Crane Scheduling Problem (QCSP). Actually, when no direct transshipment operation has to be performed, the DCTP separates into two non-standard QCSP, where for some export containers also the stowage position must be decided. Vice-versa, if the stowage plans of the two involved vessels are completely known, the resulting DCTP reduces to two independent QCSP, one for each vessel.

The direct transshipment of containers allows to reduce yard congestions and, at the same time, the storage costs. From this point of view, it seems certainly a profitable modality for the terminal management. As for the shipping line companies, the saving in the storage costs must be evaluated in connection with possible increasing of the berthing times. This is the focus of future developments.

References

1. Alicke, K.: Modeling and optimization of the intermodal terminal Mega Hub. *OR Spectr.* **24**(2), 1–18 (2002)
2. Bierwirth, C., Meisel, F.: A fast heuristic for quay crane scheduling with interference constraints. *J. Sched.* **12**(4), 345–360 (2009)
3. Blumenhagen, D.: Containerization and Hinterland traffic. *Marit. Policy Manag.* **8**(3), 197–206 (1981)
4. Chiarello, A., Gaudio, M., Sammarra, M.: Truck synchronization at single door cross-docking terminals. *OR Spectr.* **40**(2), 395–447 (2018)
5. Garro, A., Monaco, M.F., Russo, W., Sammarra, M., Sorrentino, G.: Agent-based simulation for the evaluation of a new dispatching model for the straddle carrier pooling problem. *Simulation* **91**(2), 181–202 (2015)
6. Gendreau, M., Potvin, J.Y.: Tabu search. In: Gendreau, M., Potvin, J.Y. (eds.) *Handbook of Metaheuristics*. ISOR, vol. 146, pp. 41–59. Springer, Boston (2010). https://doi.org/10.1007/978-1-4419-1665-5_2
7. Lee, B.K., Jung, B.J., Kim, K.H., Park, S.O.: A simulation study for designing a rail terminal in a container port. In: Perrone, L.F., Wieland, F.P. et al. (eds.) *Proceedings of the 38th Conference on Winter Simulation*, pp. 1388–1397 (2006)
8. Lian, C., Hwang, H., Gen, M.: A berth allocation planning problem with direct transshipment consideration. *J. Intell. Manuf.* **23**(6), 2207–2214 (2012)
9. Meisel, F., Bierwirth, C.: A unified approach for the evaluation of quay crane scheduling models and algorithms. *Comput. Oper. Res.* **38**(3), 683–693 (2011)
10. Monaco, M.F., Sammarra, M.: Quay crane scheduling with time windows, one-way and spatial constraints. *Int. J. Shipp. Transp. Logist.* **3**(4), 454–474 (2011)
11. Monaco, M.F., Sammarra, M.: The direct ship-to-ship container transshipment problem at a maritime terminal. In: Gunther, H.-O., Kim, K.-H., Kopfer, H. (eds.) *International Conference on Logistics and Maritime Systems, LOGMS2012*, pp. 79–89 (2012)

12. Monaco, M.F., Sammarra, M., Sorrentino, G.: The terminal-oriented ship stowage planning problem. *Eur. J. Oper. Res.* **239**(1), 256–265 (2014)
13. Nam, H., Lee, T.: A scheduling problem for a novel container transport system: a case of mobile harbor operation schedule. *Flex. Serv. Manuf. J.* **25**, 576–608 (2013)
14. Sammarra, M., Cordeau, J.-F., Laporte, G., Monaco, M.F.: A tabu search algorithm for the quay crane scheduling problem. *J. Schedul.* **10**(4–5), 327–336 (2007)
15. Zeng, Q., Feng, Y., Chen, Z.: Optimizing berth allocation and storage space in direct transshipment operations at container terminals. *Marit. Econ. Logist.* **19**(3), 474–503 (2017)