

# Optimal Berth Allocation, Time-variant Quay Crane Assignment and Scheduling with Crane Setups in Container Terminals<sup>1</sup>

Yavuz B. Türkoğulları<sup>a</sup>, Z. Caner Taşkın<sup>b,\*</sup>, Necati Aras<sup>b</sup>, İ. Kuban Altınel<sup>b</sup>

<sup>a</sup>*Department of Industrial Engineering, Maltepe University, İstanbul, Turkey*

<sup>b</sup>*Department of Industrial Engineering, Boğaziçi University, 34342, Bebek, İstanbul, Turkey*

---

## Abstract

There has been a dramatic increase in world's container traffic during the last thirty years. As a consequence, the efficient management of container terminals has become a crucial issue. In this work we concentrate on the integrated seaside operations, namely the integration of berth allocation, quay crane assignment and quay crane scheduling problems. First, we formulate a mixed-integer linear program whose exact solution gives optimal berthing positions and berthing times of the vessels, along with their crane schedules during their stay at the quay. Then, we propose an efficient cutting plane algorithm based on a decomposition scheme. Our approach deals with berthing positions of the vessels and their assigned number of cranes in each time period in a master problem, and seeks the corresponding optimal crane schedule by solving a subproblem. We prove that the crane scheduling subproblem is NP-complete under general cost settings, but can be solved in polynomial time for certain special cases. Our computational study shows that our new formulation and proposed solution method yield optimal solutions for realistic-sized instances.

*Keywords:* Integer programming; container terminal management; berth allocation; crane assignment; crane scheduling.

---

## 1. Introduction

Shorter transit times, lower shipping costs, higher reliability and security, and multi-modality are some of the factors that make containers an attractive means of transportation. There has been a considerable growth in the share of containerized trade in the world's total dry cargo during the last 30 years: the increase between 2000 and 2010 was 53.3%, 11% of which is attributed to year 2010 alone (UNCTAD, 2011). As a consequence of such a drastic increase in container traffic, the efficient management of container terminals has become a crucial issue and attracted a considerable research effort from various disciplines, including Operational Research (Steenken et al., 2004; Stahlbock and Voß, 2008). This is a difficult task since there is a myriad of interdependent operations, which can be grouped as the seaside, transfer and yard operations (Vis and de Koster, 2003). In this work, we concentrate on the integrated planning of seaside operations, which includes the berth allocation problem (BAP), quay crane assignment problem

---

<sup>1</sup>A shorter version of this paper has been presented in the 2015 International Conference on Logistics and Maritime Systems (LOGMS 2015).

\*Corresponding author. Phone: + 90 (212) 359 7074, Fax: + 90 (212) 265 1800

*Email addresses:* yavuz.turkogullari@maltepe.edu.tr (Yavuz B. Türkoğulları), caner.taskin@boun.edu.tr (Z. Caner Taşkın), arasn@boun.edu.tr (Necati Aras), altinel@boun.edu.tr (İ. Kuban Altınel)

(CAP) and quay crane scheduling problem (CSP). We direct any reader having a particular interest in the transfer and yard operations, especially in the transshipment flow of containers and the associated decision problems between the seaside and the yard, to the work by Vacca (2010).

Efficient planning of seaside operations has a direct impact on the dwell time of the vessels (i.e., the period elapsing between the arrival and departure times of the vessels, consisting of the sum of the waiting and processing times), which is one of the main performance measures at a container terminal. Major international maritime ports have several container terminals operated by different companies. Thus, longer dwell times can have a negative impact on the competitiveness of both the port and companies operating terminals there. This explains the existence of studies in the literature that are concerned with BAP, CAP, CSP, and their integrated versions; namely berth allocation and quay crane assignment problem (BACAP), crane assignment and scheduling problem (CASP) and berth allocation, quay crane assignment and scheduling problem (BACASP). The type of integration we aim for here is the so-called deep integration defined by Geoffrion (1994), where subproblems are combined in the form of a single, unified mathematical optimization framework.

Generally, BAP deals with the determination of optimal berthing times and positions of vessels in container terminals. The focus of CSP, on the other hand, is mainly in determining an optimal handling schedule of vessels for available cranes at the terminal. Excellent recent surveys of the related works with a classification according to some specific attributes are provided in Bierwirth and Meisel (2010, 2015), and Carlo et al. (2013). CAP finds the optimal number of cranes assigned to the vessels, and thus can be seen as a special form of the optimal crane splitting problem (Steenken et al., 2004). CAP is relatively simpler and can easily be solved in practice to optimality using intuitive reasoning, which makes it a less attractive research problem. However, as can be realized, crane numbers have a direct effect on the processing times of the vessels. As a result, related decisions should be embedded within either BAP or CSP models.

To the best of our knowledge, the earliest attempts to study BACASP, namely the integration of BAP, CAP and CSP within a unified mathematical optimization model to determine optimal berthing times and positions of the vessels along with the number of cranes and their identities as well as schedules are due to Rashidi (2006), Theofanis et al. (2007), and Zhang et al. (2010). The integration is slightly weaker on the BAP side in the work by Liu et al. (2006). Their BACAP model determines both the optimal crane numbers and specifications, but an optimal solution includes berthing times of the vessels without any information on their berthing positions. Besides, they generate possible handling times for each vessel and each assignable number of cranes as inputs to their BACASP model using a preprocessing scheme. The same approach is adopted by Meier and Schumann (2007), Meisel (2009), Meisel and Bierwirth (2013), and Ak (2008). Meier and Schumann (2007) try to achieve this by functionally integrating their CASP model with BAP. However, Meisel (2009) and Meisel and Bierwirth (2013) functionally integrate BAP, CAP and CSP in three phases. In his work on the optimal planning of the seaside operations, Ak (2008) develops a mixed-integer linear programming (MILP) model that integrates BAP, CAP and CSP. His BACASP model calculates optimal berthing times, berth allocations and crane number assignments of the vessels, and crane schedules simultaneously for calculated crane assignments. In this formulation, berth allocation constraints are based on the relative positions of the vessels in time-space representation, crane scheduling allows crane shifting with no crossing and the objective

minimizes the sum of the total dwell times of the vessels and penalty caused by the late vessels. As can be noticed, Ak (2008) with this objective intends to favor fast vessel throughput. While this objective attempts to treat different vessels fairly, the cost accrued due to the roaming cranes is not considered. Yang (2008) also proposes an MILP formulation that integrates BAP, CAP and CSP. In this formulation, crane shifting is allowed and the berth allocation constraints are based on assigning the rectangles of the time-space grid (i.e., position assignments) to the vessels and the total makespan is minimized as the objective. Unfortunately, the negative effect of crane reallocations is not considered since crane traveling cost is ignored. In addition, both models can be solved exactly only for small instances, and therefore the authors propose heuristics for the solution of their formulations.

Regarding the number and mobility of cranes assigned to the vessels, there are different assumptions made in the literature. Some authors assume that the number of cranes may change dynamically between the minimum and maximum number of cranes (see, e.g. Park and Kim (2003); Meisel and Bierwirth (2009)). Zhang et al. (2010) criticize such models by noting that the number of quay cranes assigned to a vessel cannot be changed frequently in practice due to the significant setup effort and adjustment time needed for moving the quay cranes. They assume that the number of quay cranes assigned to each vessel can be changed only once before the vessel departs. This motivation for reducing large setup losses due to the reallocation of quay cranes leads Türkoğulları et al. (2014) to formulate a time-invariant model where the same subset of quay cranes serves a vessel during its stay at the berth. They essentially adopt the position assignment settings given in Ak (2008), where he considers continuous quay layout discretized by means of fixed length sections (for modeling purposes) and dynamic arrivals with time discretized by means of fixed length unit periods. In other words, in the position assignment formulation the time-space area is divided into unit rectangles having one berth section as height and one time period as length, which reduces BAP to the positioning of a certain number of small rectangles within a large rectangle without overlap. Here, each small rectangle represents a vessel whose height equals to its length measured in the number of berth sections and width equals to its berthing time measured in the number of periods as illustrated by Ak (2008), and Bierwirth and Meisel (2010). Türkoğulları et al. (2014) first provide a new position assignment formulation for time-invariant BACAP minimizing the total cost consisting of the sum of the cost of berthing away from the desired berth section, the cost of late berthing, the cost of late departures and the cost due to the change in the number of active cranes. Then, they propose an efficient cutting plane algorithm that benefits from an optimal solution of their BACAP formulation for exactly solving large BACASP instances. Although this is a very efficient method that can solve realistic instances, the crane assignments are time-invariant (Bierwirth and Meisel, 2010). This means that the group of cranes and their number assigned to a vessel remains the same during its stay at the berth. In Türkoğulları et al. (2012) the authors relax time-invariance assumption and develop an MILP model that considers a time-variant version of the same problem and incorporate the cost due to the change in the number of assigned cranes. It is also a position assignment formulation minimizing the total cost but includes the flexibility that the number of active cranes may change dynamically. However, this approach is still restricted since the setup cost is not considered as realistically as possible and it is assumed that total setup cost is proportional to the number of setups. In other words, the total setup cost term in the objective function is based primarily on the frequency that the number of active cranes changes and does not pay attention to issues such as traveling time and traveling distance of the cranes. Similar issues arise for the integrated formulations of Ak

(2008) and Yang (2008); they both consider time-variant crane assignments but do not take into account additional complications due to crane relocations and the resulting cost figures.

In this work, we follow this line of research and introduce a formulation that deeply integrates BAP, CAP, and CSP (BACASP), as our first major contribution. Our formulation allows the inclusion of realistic costs associated with crane relocations consisting of both variable and fixed costs. We remark that CSP considered here as a part of BACASP is similar to bus/train scheduling with cranes corresponding to buses/trains and berthed vessels waiting for cranes replacing bus stops/train stations. Since there is a distribution of resources, namely cranes, over time, this is a scheduling problem as well. However, it is simpler than crane operations scheduling and thus named as CAP(specific) by Bierwirth and Meisel (2010). The crane operation scheduling is defined by Bierwirth and Meisel (2010, 2015) and Carlo et al. (2013) as the general task scheduling for cranes. As a result, BACASP may be seen as the deep integration of BAP, CAP and CSP (i.e. these three problems are addressed within a unified framework in the form of a mathematical optimization decomposition model), where CSP determines the crane-to-vessel assignments and specific time periods of these assignments when berthing times and locations of the vessels and the number of cranes allocated to vessels are given. We also point out that this definition of CSP has been used before in the literature (Park and Kim, 2003; Imai et al., 2008; Ak, 2008; Yang, 2008). The formulation we present in the following sections is, however, capable of handling various kinds of setup costs due to crane relocations such as the traveling time and distance, and fixed costs that can incur due to possible crane-specific, vessel-specific, and location-specific reasons. Our second major contribution is a new exact solution algorithm for BACASP. It is an efficient cutting plane algorithm, which is based on a decomposition scheme that isolates the determination of crane schedules given the berthing positions of the vessels and the number of cranes assigned to them. We show that the CSP subproblem is computationally difficult in general and discuss some special cases where it is easily solvable. We also develop a new branch-and-bound algorithm using minimum cost flow relaxations and compare its performance with the shortest path implementation on the state network of a dynamic programming formulation, whose roots are in the earlier work by Park and Kim (2003).

The remainder of the paper is organized as follows. The next section is devoted to the formulation of the new model. In Section 3 and Section 4 we explain respectively the new decomposition algorithm that solves BACASP exactly and an algorithm that gives a quick upper bound on the optimal value. In Section 5 we mainly focus on CSP and propose approaches for its efficient solution. In Section 6 we present enhancements increasing the efficiency of the new methods. Computational results demonstrating the efficiency of the proposed approaches are reported in Section 7. Finally, concluding remarks and future research directions are listed in Section 8.

## 2. Formulation of the Integrated Model

Our proposed integrated BACASP formulation is mainly based on the following eight assumptions:

1. The planning horizon is divided into time periods of equal length.
2. The berth is continuous and discretized by equal-sized unit berth sections. They are just as large as a single crane can operate freely in each of them.

3. Each berth section is occupied by no more than one vessel in each time period.
4. The desired berthing sections of the vessels are known. The preference for a berth section may be due to its proximity to the portion of the yard where the containers are to be unloaded (loaded) from (into) the vessel.
5. Each quay crane can work on at most one vessel per time period.
6. Each vessel has a minimum and maximum number of quay cranes that can be assigned to it; and it is long enough to accept the maximum number of cranes.
7. The service of a vessel by the quay cranes starts immediately after its berthing and continues without disruption until its departure with possible changes in the number of allocated cranes.
8. The net amount of work  $q$  cranes can produce while working on the same vessel for one period is  $q^\lambda$  crane-periods. Here,  $\lambda$  is the interference exponent, which can be set to values in  $(0, 1]$ .

Assumptions 1 through 7 are widely used in the literature and are quite realistic. Assumption 8 is related to the decrease in the productivity of the quay cranes due to the interference among them and extends the range of applicability of our model. According to Schonfeld and Sharafeldien (1985) the productivity loss can be modeled by using an interference exponent that reduces the marginal productivity of quay cranes. For a given interference exponent  $\lambda \in (0, 1]$ , the productivity obtained from assigning  $q$  cranes to a vessel for one period is given by a total of  $q^\lambda$  crane-periods. This idea is also adopted by Silberholz et al. (1991) to improve the allocation of human resources in container terminals, and by Oguz et al. (2004) and Dragovic et al. (2006) to describe the berthing process with respect to the quay crane assignment.

In our solution approach, we decompose the problem into a master problem and a subproblem. The master problem integrates BAP and CAP, and thus its solution yields only feasible berth allocations and crane number assignments to the vessels. The subproblem is the CSP and its solution provides a crane schedule minimizing total setup cost due to crane relocations, given a set of feasible berth allocations and crane number assignments of the vessels. Notice that, because the optimal value of CSP is relative to the solution of BAP and CAP integration (i.e. BACAP) we can eventually obtain upper and lower bounds on the optimal value of BAP, CAP and CSP integration (i.e. BACASP). However, it is possible to gradually improve these bounds to make them converge to each other via a Benders-like decomposition scheme (Benders, 1962), which are explained in Section 3. We use optimality cuts, which is a known solution approach for intractable mathematical programming problems. For example, Laporte and Louveaux (1993) report a successful application of optimality cuts similar to (13) in their early work on stochastic integer programs with complete recourse.

All BACASP formulations mentioned in the literature review integrate somehow the formulations of three problems: BAP, CAP and CSP. Hence, one can notice that the essential body of the constraints of BACASP is grouped in three blocks, each belonging to one of these three problems. The remaining ones are the coupling constraints, whose role is mainly to attach the blocks. This is also the case for our formulation whose parameters and decision variables are summarized with their brief descriptions in Table 1 and Table 2, respectively. In Table 1, parameters in Greek letters are given first in alphabetic order followed by the parameters in Latin letters.

There are many summations with lower and upper limits of the indices, which are long and difficult to follow in the formulation. We have reformulated them using compact representations for the sake of notational convenience.

Table 1: Parameters used in our mathematical model

Parameter	Definition
$\alpha_{it}$	Arrival parameter, which is set to 1 if vessel $i$ can start berthing in period $t$
$\beta_{it'}$	Departure parameter, which is set to 1 if vessel $i$ that berths in period $t$ can depart in period $t'$
$\delta_i^e$	Upper limit on the deviation of vessel $i$ 's berthing time from $e_i$
$\delta_i^s$	Upper limit on the deviation of vessel $i$ 's berthing position from $s_i$
$\lambda$	Interference exponent, a real number in $(0, 1]$
$\theta^m$	Setup cost that corresponds to the $m^{\text{th}}$ feasible solution of the master problem computed by solving the Crane Scheduling Subproblem (CSSP)
$\phi_i^e$	Cost of berthing one period later than the expected arrival time for vessel $i$
$\phi_i^d$	Cost of departing one period later than the due time for vessel $i$
$\phi_i^s$	Cost of deviating one unit from the desired berthing section for vessel $i$
$B$	Number of berthing sections
$e_i$	Expected arrival time of vessel $i$
$K$	Number of quay cranes
$\bar{k}_i$	Maximum number of cranes that can be assigned to vessel $i$
$\underline{k}_i$	Minimum number of cranes that can be assigned to vessel $i$
$\ell_i$	Length of vessel $i$ measured in terms of the number of discretized quay sections
$\bar{p}_i$	Upper bound on the processing time of vessel $i$ , which is set to $\left\lceil w_i / \underline{k}_i^\lambda \right\rceil$
$\underline{p}_i$	Lower bound on the processing time of vessel $i$ , which is set to $\left\lceil w_i / \bar{k}_i^\lambda \right\rceil$
$r_i$	Restarting time of a new journey or departure due time of vessel $i$
$s_i$	Desired berthing section of vessel $i$
$T$	Number of time periods
$V$	Number of vessels
$w_i$	Workload of vessel $i$ in crane-periods

Table 2: Decision variables used in our mathematical model

Decision Variable	Definition
$\theta$	Variable lower bound on the minimum total setup cost
$a_t$	One if there is a berthing vessel in period $t$ , zero otherwise
$d_t$	One if there is a departing vessel in period $t$ , zero otherwise
$x_{ijt'}$	One if vessel $i$ berths at section $j$ in period $t$ and departs in period $t'$ , zero otherwise
$y_{it}$	One if vessel $i$ is in the berth in period $t$ , zero otherwise
$z_{iqt}$	One if the number of cranes assigned to vessel $i$ in period $t$ is $q$ , zero otherwise

The definitions of these representations are given in Table 3. The quantities replacing these limits are listed in the first column; their explanations and the original terms they replace are provided in the second and third columns, respectively. For example,  $s_i - \delta_i^s$  and  $s_i + \delta_i^s$  appear in the summation limits for berth section index  $j$ , and provide lower and upper bounds on the desired berthing section  $s_i$  of vessel  $i$ , respectively. Here,  $\delta_i^s$  can be obtained by generating a feasible solution of BACASP for which the value of the objective function is  $Z_f$ . Since each unit deviation of vessel  $i$ 's berthing position from the desired one contributes  $\phi_i^s$  units to the objective value,  $\delta_i^s$  can be set equal to  $\lceil Z_f / \phi_i^s \rceil$ . In other words, if the total cost were attributed only to this cost component, then the deviation from the desired berthing position would become equal to  $\delta_i^s$ . In a similar fashion, we can set an upper limit on the tardiness of vessel  $i$  with

respect to its expected arrival time  $e_i$ . It is denoted by  $\delta_i^e$  and can be set to  $\lceil Z_f / \phi_i^e \rceil$ , where  $\phi_i^e$  is the cost of vessel  $i$ 's berthing one period later than  $e_i$ . As previously mentioned, a feasible solution of BACASP is needed for obtaining the values of these limits. We describe an efficient computational procedure for its computation in the following section.

Table 3: Lower and upper limits of the indices

Parameter	Explanation	Original term
$\sigma_i^1$	The smallest index of the berth section for vessel $i$	$\max(1, s_i - \delta_i^s)$
$\sigma_i^2$	The largest index of the berth section for vessel $i$	$\min(B - \ell_i + 1, s_i + \delta_i^s)$
$\eta_i^1(t)$	The earliest period in which vessel $i$ can depart. Here $t$ is the berthing time of the vessel	$t + \underline{p}_i - 1$
$\eta_i^2(t)$	The latest period in which vessel $i$ can depart. Here $t$ is the berthing time of the vessel	$\min(t + \bar{p}_i - 1, T)$
$\tau_i$	The latest period in which vessel $i$ can berth	$\min(T - \underline{p}_i + 1, e_i + \delta_i^e)$

We begin with the formulation of the objective function. It is the total cost to be minimized and expressed as the sum of two terms:

$$\sum_{i=1}^V \sum_{j=\sigma_i^1}^{\sigma_i^2} \sum_{t=e_i}^{\tau_i} \sum_{t'=\eta_i^1(t)}^{\eta_i^2(t)} \{ \phi_i^s |j - s_i| + \phi_i^e (t - e_i) + \phi_i^r \max(0, t' - r_i) \} x_{ijtt'} + \theta. \quad (1)$$

The first term is obtained by summing up over the vessels the costs of deviation from the desired berthing section  $s_i$ , the costs of berthing later than the expected arrival time  $e_i$  (i.e. late arrival), and the costs of departing later than the departure due time  $r_i$  that marks the time at which the loading/unloading operation of the vessel should be finished (i.e. late departure). These three cost components are formulated by means of the binary variable  $x_{ijtt'}$ , which represents the allocation of berths to vessels for a time interval; it is set to one if vessel  $i$  berths at section  $j$  from time  $t$  until its departure time  $t'$ , and zero otherwise. The second term,  $\theta$ , represents a lower bound on the minimum total setup cost due to crane movements.

We adopt the position assignment approach in formulating BAP by means of constraints (2) and (3) given below;

$$\sum_{j=\sigma_i^1}^{\sigma_i^2} \sum_{t=e_i}^{\tau_i} \sum_{t'=\eta_i^1(t)}^{\eta_i^2(t)} x_{ijtt'} = 1 \quad i = 1, \dots, V, \quad (2)$$

$$\sum_{\substack{i=1 \\ e_i \leq \hat{t}}}^V \sum_{j=\max(1, \hat{j}-\ell_i+1)}^{\min(B-\ell_i+1, \hat{j})} \sum_{t=e_i}^{\min(\tau_i, \hat{t})} \sum_{t'=\max(\eta_i^1(t), \hat{t})}^{\eta_i^2(t)} x_{ijtt'} \leq 1 \quad \hat{j} = 1, \dots, B; \hat{t} = 1, \dots, T. \quad (3)$$

They respectively enforce that each vessel is allocated to a berth section for a time interval (i.e., from  $t$  to  $t'$ ), and a berth section can be occupied by at most one vessel in any time period. In fact, they help positioning rectangles over

the time-space diagram without any overlap. The next three sets of constraints are related to CAP.

$$\sum_{q=0}^{\bar{k}_i} z_{iqt} = 1 \quad i = 1, \dots, V; t = e_i, \dots, T, \quad (4)$$

$$\sum_{t=e_i}^T \sum_{q=0}^{\bar{k}_i} q^\lambda z_{iqt} \geq w_i \quad i = 1, \dots, V, \quad (5)$$

$$\sum_{\substack{i=1 \\ e_i \leq t}}^V \sum_{q=0}^{\bar{k}_i} q z_{iqt} \leq K \quad t = 1, \dots, T. \quad (6)$$

Binary decision variable  $z_{iqt}$  equals one if the number of cranes assigned to vessel  $i$  in period  $t$  is  $q$ , zero otherwise. Constraints (4) determine the number of cranes assigned to a vessel in a period, while (5) guarantee that the assignments satisfy the required workload of the vessels (measured by crane-periods). Constraints (6) ensure that the number of assigned cranes does not exceed the number of available ones. Notice that assigning  $q$  cranes to vessel  $i$  in time period  $t$  does not necessarily imply a crane productivity of  $q$  crane-periods. This is because of the productivity loss due to the interference of the quay cranes. If the interference exponent  $\lambda$  is less than one, i.e. if there is a loss due to the interference between the cranes, the productivity obtained by assigning  $q$  cranes to vessel  $i$  in time period  $t$  is  $q^\lambda$ , which is less than  $q$ .

BAP and CAP constraints given above are independent of each other. To couple them we define a new binary decision variable  $y_{it}$ . It is an auxiliary variable and its value equals one if vessel  $i$  is berthed in period  $t$ , zero otherwise. Equalities

$$y_{i\hat{t}} = \sum_{j=\sigma_i^1}^{\sigma_i^2} \sum_{t=e_i}^{\min(\tau_i, \hat{t})} \sum_{t'=\max(\eta_i^1(t), \hat{t})}^{\eta_i^2(t)} x_{ijt'} \quad i = 1, \dots, V; \hat{t} = e_i, \dots, T, \quad (7)$$

relate variables  $y_{it}$  with variables  $x_{ijt'}$ . Lower and upper bound inequalities (8) on the number of cranes used for vessel  $i$  in period  $t$ ,

$$\underline{k}_i y_{it} \leq \sum_{q=0}^{\bar{k}_i} q z_{iqt} \leq \bar{k}_i y_{it} \quad i = 1, \dots, V; t = e_i, \dots, T, \quad (8)$$

form the relationship between  $y_{it}$  and  $z_{iqt}$ . Namely, no crane is assigned to a vessel when it is not berthed and the number of assigned cranes must lie between its lower and upper bounds in each period.

CSP constraints are essential for controlling the crane assignments over time. We formulate them by means of new binary decision variables  $a_t$  and  $d_t$ .  $a_t$  equals one if there is an arrival to the berth in period  $t$ , and zero otherwise. Similarly,  $d_t$  equals one if there is a departure from the berth in period  $t$ , and zero otherwise. Constraints (9) identify whether or not there is an arrival in period  $t$ . They are followed by constraints (10), which are their departure-related



versions.

$$a_t \leq \sum_{i=1}^V \sum_{j=\sigma_i^1}^{\sigma_i^2} \sum_{t'=\eta_i^1(t)}^{\eta_i^2(t)} \alpha_{it'} x_{ijt'} \leq Va_t \quad t = 1, \dots, T, \quad (9)$$

$$d_{t'} \leq \sum_{i=1}^V \sum_{j=\sigma_i^1}^{\sigma_i^2} \sum_{t=e_i}^{\tau_i} \beta_{it'} x_{ijt'} \leq Vd_{t'} \quad t' = 1, \dots, T. \quad (10)$$

Notice the use of the arrival and departure coefficients  $\alpha_{it}$  and  $\beta_{it'}$ , which determine whether or not vessel  $i$  can begin or end berthing in period  $t$ . Their values are set as:

$$\alpha_{it} = \begin{cases} 1 & \text{if } e_i \leq t \leq \tau_i \\ 0 & \text{otherwise} \end{cases}$$

and

$$\beta_{it'} = \begin{cases} 1 & \text{if } \eta_i^1(t) \leq t' \leq \eta_i^2(t) \\ 0 & \text{otherwise} \end{cases}.$$

The two sets of constraints

$$\sum_{q=0}^{\bar{k}_i} qz_{iqt} - \sum_{q=0}^{\bar{k}_i} qz_{iq(t-1)} \leq \bar{k}_i (a_t + d_{t-1}) \quad i = 1, \dots, V; e_i + 1 \leq t \leq T, \quad (11)$$

$$\sum_{q=0}^{\bar{k}_i} qz_{iq(t-1)} - \sum_{q=0}^{\bar{k}_i} qz_{iqt} \leq \bar{k}_i (a_t + d_{t-1}) \quad i = 1, \dots, V; e_i + 1 \leq t \leq T, \quad (12)$$

relate vessel arrivals and departures with the number of assigned quay cranes: the number of cranes assigned to vessel  $i$  can change from period  $t - 1$  to period  $t$  only if arrival of another vessel occurs in period  $t$  or departure of another vessel occurs in period  $t - 1$ . In fact, these inequalities follow from the linearization of the dominance relations

$$\left| \sum_{q=0}^{\bar{k}_i} qz_{iqt} - \sum_{q=0}^{\bar{k}_i} qz_{iq(t-1)} \right| \leq \bar{k}_i (a_t + d_{t-1}) \quad i = 1, \dots, V; e_i + 1 \leq t \leq T.$$

The last set of constraints determines the minimum total crane setup cost by setting the value of variable  $\theta$  to the largest possible lower bound on it. Since the decision variables in constraints (2)–(12) are binary, there is only a finite number of values for variables  $x_{ijt'}$  and  $z_{iqt}$  that satisfy these constraints. Let  $m = 1, \dots, M$  index these feasible solutions, and consider the set of inequalities

$$\theta \geq \theta^m - \theta^m \left[ \left( V - \sum_{i=1}^V \sum_{\{j,t,t':x_{ijt'}^m=1\}} x_{ijt'} \right) + \sum_{i=1}^V \left( (T - e_i + 1) - \sum_{t=e_i}^T \sum_{\{q:z_{iq}^m=1\}} z_{iqt} \right) \right] \quad m = 1, \dots, M, \quad (13)$$

where  $\theta^m$  denotes the minimum total setup cost obtained by solving the crane scheduling subproblem (CSSP) for given  $\mathbf{x}^m$  and  $\mathbf{z}^m$ . Note that the summation of the  $\mathbf{x}$ -variables on the right-hand side of (13) is  $V$  for  $\mathbf{x} = \mathbf{x}^m$ , and is strictly less than  $V$  for any other feasible  $\mathbf{x} \neq \mathbf{x}^m$ . Similarly, the summation of the  $\mathbf{z}$ -variables on the right-hand side

of (13) is  $T - e_i + 1$  for  $\mathbf{z} = \mathbf{z}^m$ , and is strictly smaller for any other feasible  $\mathbf{z} \neq \mathbf{z}^m$ . Thus, the term in square brackets on the right-hand side of (13) is zero for  $(\mathbf{x} = \mathbf{x}^m, \mathbf{z} = \mathbf{z}^m)$ , and is strictly positive for any other feasible solution of the problem. Hence, the  $m^{\text{th}}$  constraint of type (13) reduces to  $\theta \geq \theta^m$  for the  $m^{\text{th}}$  feasible solution, and is redundant for any other feasible solution.

### 3. Exact Solution Algorithm

Observe that the upper limit  $M$  of index  $m$ , and thus the number of constraints (13), is finite. Observe also that the objective function (1), which is to be minimized, is the total cost including  $\theta$  as a cost term, and constraints (13) ensure that  $\theta$  equals the setup cost corresponding to an optimal solution of the crane scheduling subproblem. Consequently, it is possible to generate all constraints of type (13) to obtain a monolithic formulation, in principle. Then, one can solve the formulation that minimizes (1) subject to the constraints (2)–(13) to find an optimal solution of BACASP. Unfortunately, this is possible only for very small instances, as the number of inequalities (13) can be exponential.

A typical approach considered in similar cases is to decompose the problem into a relaxed master problem and a subproblem whose solution gradually generates the constraints, namely cutting planes, to be added to the relaxed master problem until an optimal solution is found. To be more specific, the relaxed master problem at some stage consists of (1)–(12) with the generated inequalities of type (13) up to that stage. Moreover, the subproblem becomes simply the determination of an optimal crane schedule relative to a set of given feasible berth allocations and crane number assignments. We formally express the steps of this approach with Algorithm 1, where  $H(\mathbf{x}, \mathbf{z}, \theta^t)$  represents the right-hand side of inequality (13) with  $\theta^t$  instead of  $\theta^m$ .

---

#### Algorithm 1

---

**Initialization**  $UB \leftarrow \infty, LB \leftarrow -\infty, \theta^0 \leftarrow -\infty, m \leftarrow 1$   
Initialize the master problem with constraints (2)–(12)  
**loop**  
Solve the master problem,  $\min (1)$  subject to (2)–(12) and  $\theta \geq H(\mathbf{x}, \mathbf{z}, \theta^t)$   $t = 0, 1, \dots, m - 1$  to optimality.  
Record optimal  $\mathbf{x}, \mathbf{z}$ , i.e. optimal solution, as  $\mathbf{x}^m, \mathbf{z}^m$ .  
Record the optimal objective value as the current lower bound  $LB$ .  
Compute the minimum total setup cost  $\theta^m$  corresponding to  $\mathbf{x}^m, \mathbf{z}^m$  by running Algorithm 2.  
Calculate  $O(\mathbf{x}^m, \mathbf{z}^m, \theta^m)$ , the objective value of the master problem corresponding to  $\mathbf{x}^m, \mathbf{z}^m, \theta^m$  by evaluating (1).  
**if**  $O(\mathbf{x}^m, \mathbf{z}^m, \theta^m) \leq UB$  **then**  
     $UB \leftarrow O(\mathbf{x}^m, \mathbf{z}^m, \theta^m), \mathbf{x}^* \leftarrow \mathbf{x}^m, \mathbf{z}^* \leftarrow \mathbf{z}^m, \theta^* \leftarrow \theta^m$ .  
**end if**  
**if**  $LB = UB$  **then**  
    Stop;  $\mathbf{x}^*, \mathbf{z}^*, \theta^*$  are optimal.  
**else**  
    Add constraint (13) corresponding to  $\mathbf{x}^m, \mathbf{z}^m, \theta^m$  to the master problem.  
     $m \leftarrow m + 1$ .  
**end if**  
**end loop**

---

As can be noticed, we initially relax constraints (13) and solve the relaxed master problem. The optimal objective value of the relaxed master is a lower bound ( $LB$ ) on the optimal value of BACASP. We calculate the minimum

total setup cost corresponding to the current solution  $(\mathbf{x}^m, \mathbf{z}^m)$  using Algorithm 2 presented in Section 5.2.3, which determines an optimal crane schedule corresponding to  $(\mathbf{x}^m, \mathbf{z}^m)$ . Quantity  $O(\mathbf{x}^m, \mathbf{z}^m, \theta^m)$  corresponds to the value of the objective function obtained with feasible solution  $\mathbf{x}^m, \mathbf{z}^m, \theta^m$  of BACASP and thus forms an upper bound ( $UB$ ) on its optimal value. We update the  $UB$  if we obtain a better (i.e. smaller) upper bound. If the  $LB$  is equal to the  $UB$ , we terminate the algorithm as the current solution is an optimal solution of BACASP. Otherwise, we add constraint (13) corresponding to  $\mathbf{x}^m, \mathbf{z}^m, \theta^m$  to the master problem and solve it again. Figure 1 includes a diagram illustrating this communication between the relaxed master problem and the subproblem.

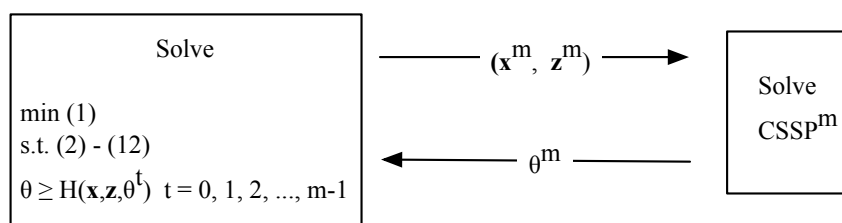


Figure 1: Information transfer between the relaxed master problem and subproblem in Algorithm 1

#### 4. Generating a Feasible Solution of BACASP

We use  $\delta_i^s$  and  $\delta_i^e$  in the lower and upper limits of several summations of the BACASP formulation given with (1)–(13). The calculation of these quantities requires the generation of a feasible solution of BACASP. To this end we adapt the heuristic proposed in Türkoğulları et al. (2014) for solving a time-invariant version of BACASP, where it is assumed that the crane group and the number of cranes assigned to a vessel remain unchanged during the time interval the vessel stays berthed. A crane group is defined as a set of cranes that are pairwise adjacent to each other in the sequence with no other crane in between. For example,  $\{1, 2, 3\}$  and  $\{5, 6, 7, 8\}$  are crane groups, but  $\{1, 2, 4\}$  and  $\{5, 7, 8, 10\}$  are not. Note that the crane group assigned to a vessel may change even if there is no change in the number of cranes (i.e., the size of the group). The proposed heuristic considers each period  $t$  consecutively starting at  $t = 1$ . It is based on the idea of assigning vessels to available berth sections with minimum possible cost. This is accomplished by first determining in each period  $t$  the vessels that have arrived prior to and during period  $t$  and have not been assigned a berthing position yet. Among these vessels, to avoid the cost of late departure, priority is given to the ones which would depart after their due time in case the minimum number of cranes are assigned to them. The missing details, which we omit here for the sake of brevity, can be found in Türkoğulları et al. (2014). Although it is assumed in Türkoğulları et al. (2014) that productivity is proportional to the number of assigned cranes, it is possible to adapt the heuristic easily to the situation where there exists a loss in productivity due to crane interferences as we assume in this paper.

#### 5. Determination of the Total Setup Cost $\theta^m$ by Solving the Crane Scheduling Subproblem

The subproblem considered in this section, which we refer to as the *Crane Scheduling Subproblem* (CSSP), focuses on assigning quay cranes to optimal work positions in each period given the berth allocations of the vessels and number

of cranes that will serve them (i.e., a feasible solution of BACAP) with the objective of minimizing the total setup cost due to crane relocations on the berth over the planning horizon. We discuss two different solution approaches to solve a different network optimization problem. In the first approach, we construct a layered network using vessel positions and formulate a minimum cost network flow problem with additional constraints to prevent arc pairs with positive flows crossing each other, in an optimal solution. The new algorithm is based on the solution of this restricted flow problem by eliminating the crossings that occur in an optimal solution of the ordinary minimum cost flow problem obtained after relaxing constraints preventing crossing by branch-and-bound. In the second approach, we also construct a layered network, but this time the network represents the states of a dynamic programming formulation, which is also used by Park and Kim (2003). The determination of an optimal crane schedule becomes then equivalent to the solution of a single-pair shortest path problem. Both formulations make the following assumptions:

1. A setup cost incurs if one of the following three events occurs: a crane starts working on a new vessel when it was serving another one, a crane is assigned to a vessel when it was idle, or a crane stops serving a vessel and becomes idle.
2. Cranes can serve any berthed vessel but are restricted to move along a single line, and hence cannot pass each other.
3. Cranes are initially labeled according to their sequence along the berth starting from the beginning of the berth.
4. In any period, idle cranes wait for their new assignments at the available positions with given capacities located before and after the vessels.
5. Cranes are identical; the differentiation in the setup costs is only due to the difference in their working conditions.

The setup cost mentioned in the first assumption is not necessarily proportional to the travel distance or time, and can include a fixed cost due to vessel, crane and position-specific characteristics. The existence of constraints used to prevent crossings makes CSSP intractable depending on the underlying cost structure.

### 5.1. The Crane Scheduling Subproblem is NP-complete

Lim et al. (2004), Zhu and Lim (2006), and Lee et al. (2008) study two variants of CSSP. In the work of Lim et al. (2004) the objective is to maximize the total throughput, namely the number of vessels served. The objective becomes the minimization of the maximum completion time, namely the departure time of the last vessel, in the works of Zhu and Lim (2006) and Lee et al. (2008). The authors have shown that these variants are NP-complete and computationally intractable. The objective of CSSP in the form we consider here is the minimization of the total setup cost and results in a decision version of the problem which is different from the ones of Lim et al. (2004), Zhu and Lim (2006) and Lee et al. (2008). As will be seen, in our proof we use an auxiliary problem, the budget restricted version (CSSP-BR), where each crane has a fixed budget to spend on the setups. We define the decision problems associated with CSSP and CSSP-BR in the following.

**CSSP Instance:** There are  $V \in \mathbb{Z}^+$  vessels, which are served by  $K \in \mathbb{Z}^+$  available cranes in  $L \in \mathbb{Z}^+$  time intervals. The setup cost of serving vessel  $j$  after serving vessel  $i$  by some crane is  $c_{ij} \in \mathbb{Z}^+$ ,  $i, j = 0, 1, \dots, V; i \neq j$ . Here,  $c_{0j}$

is the setup cost of a crane when it starts to work on vessel  $j$  initially. A similar definition can be made for  $c_{j0}$  when a crane goes idle. In addition, there is a given number  $C \in \mathbb{Z}^+$ , and the number of cranes required by vessel  $i$  during time interval  $l$  is  $g_{il} \in \mathbb{Z}^+$ ,  $i = 1, 2, \dots, V$ ;  $l = 1, 2, \dots, L$ .

**Question:** Is there a temporal allocation of the quay cranes that satisfy vessels' needs within the planning horizon such that there is no crossing between the crane relocation paths and total setup cost is less than  $C$ ?

**CSSP-BR Instance:** The same as the CSSP instance except that the number  $C$  is replaced with  $C_k \in \mathbb{Z}^+$ ,  $k = 1, \dots, K$ .

**Question:** Is there a temporal allocation of the quay cranes that satisfy vessels' needs within the planning horizon such that there is no crossing between the crane relocation paths, and the total setup cost for each crane  $k$  is less than  $C_k$ ,  $k = 1, \dots, K$ ?

**Proposition 1.** *CSSP-BR is NP-complete.*

*Proof.* Given in the Appendix. □

As a consequence of the above proposition, it is possible to say that for  $C \geq \sum_{k=1}^K C_k$  the set of "yes" instances of CSSP includes those of CSSP-BR's. This is a pointer to the NP-completeness of the CSSP, which we show with the following proposition. The proof for the NP-hardness is obtained by reducing CSSP-BR polynomially to CSSP.

**Proposition 2.** *CSSP is NP-complete.*

*Proof.* Given in the Appendix. □

## 5.2. Minimum Cost Noncrossing Flow Formulation

The minimum cost noncrossing flow problem (MCNFP) formulation of CSSP is based on a directed, layered, single source and single sink network as shown in Figure 2. The only node of the first layer is the source node with supply equal to the total number of quay cranes. Similarly, the last layer consists of a single node as well; it is the sink node with demand equal to the total number of quay cranes. The remaining ones belong to internal layers and are pure transshipment nodes.

Except the first and last layers, each one of  $L$  layers represents a snapshot of the berth for the time elapsing between two events, which can be either an arrival or departure of a vessel. In other words, layer  $l$  exists in the network if a vessel berths or departs changing the current snapshot; it is followed by a new one. Notice that the number of cranes assigned to the berthed vessels can change only between consecutive layers, since such a change can only be caused by either an arrival or departure.

Each vessel berthed in layer  $l$  is represented by a node, whose demand is equal to the number of assigned cranes. These nodes are called vessel nodes and they are ordered starting from the bottom of the layer to the top in accordance with their position in the berth from the beginning to the end. There is a second type of node below and above each vessel node. They are called waiting nodes and represent the waiting area for idle cranes. In any layer, the number of waiting nodes is one larger than the number of vessel nodes. To summarize, by letting  $n_l$  denote the number of berthed vessels in time interval  $l$ , there are  $n_l$  vessel nodes and  $n_l + 1$  waiting nodes. Hence, the total number of nodes in layer  $l$  is  $2n_l + 1$  and nodes with an even index (i.e.,  $2, 4, \dots, 2n_l$ ) correspond to a vessel node, while those with an

odd index (i.e.,  $1, 3, \dots, 2n_l + 1$ ) represent waiting nodes with finite capacities. The capacities of the first (last) waiting node in each layer is equal to the number of available berth sections between the beginning (end) of the berth and the first (last) vessel, whereas the capacity of an intermediate waiting node is equal to the number of available berth sections between two consecutive vessels. If there is not enough room in one of these locations, then the capacity of the corresponding node is set to zero, which is a consequence of Assumption 2 given at the beginning of Section 2; a berth section can be occupied by only one quay crane.

Each vessel node in layer  $l$  has two copies. Therefore, we name the first one as the original vessel node and the duplicate as the copy vessel node. The original node  $i_l$  and its copy  $i'_l$ , for even  $i_l$  are connected by arc  $(i_l, i'_l)$ . Since the demand of an original vessel node in layer  $l$  is equal to the number of assigned cranes  $g_{il}$ , so is the demand of a copy vessel node. Therefore, both the lower bound  $\underline{u}_{i_l i'_l}$  and upper bound  $\bar{u}_{i_l i'_l}$  on the capacity of the arc  $(i_l, i'_l)$  are set to  $g_{il}$ , which implies that the flow on the arc  $(i_l, i'_l)$  for even  $i_l$  is equal to  $g_{il}$ . Similar to the vessel nodes, waiting nodes also have duplicates. An original waiting node  $i_l$  and its copy  $i'_l$  for odd  $i_l$  are connected by an arc whose capacity has a lower bound  $\underline{u}_{i_l i'_l} = 0$  and an upper bound  $\bar{u}_{i_l i'_l} = h_{i_l}$ , which is the capacity of the waiting area  $i_l$ , namely the capacity of waiting node  $i_l$ . Notice that  $i_l$  is odd. In short, nodes have associated weights in their original form. Vessel nodes have crane demands, and waiting nodes have capacities. By using duplicates, vessel nodes are transformed from transshipment sinks to pure transshipment nodes, and waiting nodes are transformed into uncapacitated, pure transshipment nodes. This construction is illustrated in Figure 2.

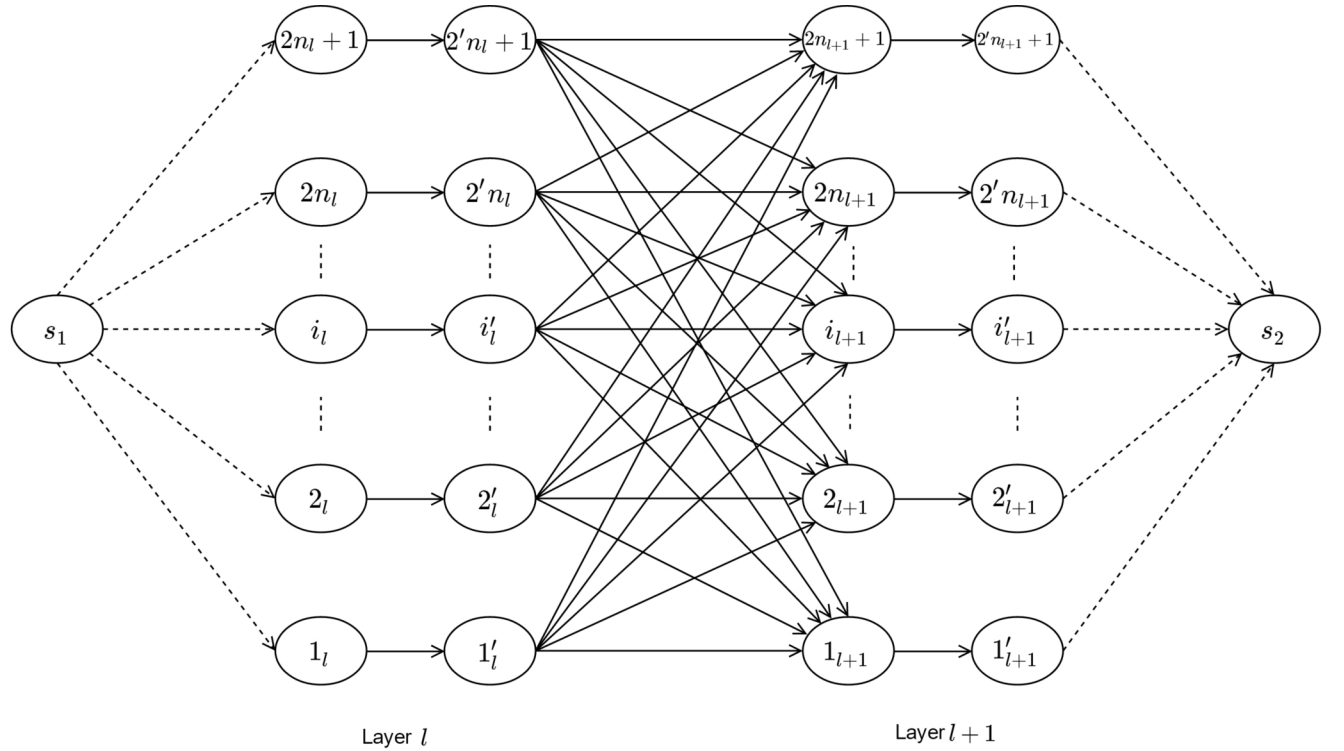


Figure 2: Network presentation of the crane scheduling subproblem

Let  $\mathcal{V}$ ,  $\mathcal{V}'$ ,  $\mathcal{W}$ , and  $\mathcal{W}'$  denote the set of original vessel nodes, copy vessel nodes, original waiting nodes, and

copy waiting nodes, respectively. Then, the overall network  $\mathcal{N}$  has  $(\mathcal{V} \cup \mathcal{V}' \cup \mathcal{W} \cup \mathcal{W}' \cup \{s_1, s_2\})$  nodes, where  $s_1$  and  $s_2$  are the source and sink nodes, respectively. By generating the duplicates of the vessel and waiting nodes, all the nodes become transshipment nodes with zero demand or supply except  $s_1$  and  $s_2$ . The supply of  $s_1$  and the demand of  $s_2$  are equal to  $K$ , that is the total number of cranes. There is an outgoing arc connecting  $s_1$  and an original node of the first layer, and an incoming arc connecting a copy node of layer  $L$  to  $s_2$ . The capacity of these arcs, as well the one of arcs  $(i'_l, j_{l+1})$ , which connect a copy node  $i'_l$  in layer  $l$  with an original node  $j_{l+1}$  in layer  $l + 1$ , have a lower bound of zero and an upper bound of  $K$ .

As can be noticed, the flows on the arcs of this network correspond to crane relocations or movements from waiting areas to vessels, from vessels to vessels (this includes the case where a crane continues serving the same vessel or starts serving a new vessel at the same berth section), or from vessels to waiting areas in each time interval. The costs associated with these relocations are defined as unit flow costs. The unit flow cost on arc  $(i'_l, j_{l+1})$  between two layers, where  $1 \leq i'_l \leq 2n_l + 1$  and  $1 \leq j_{l+1} \leq 2n_{l+1} + 1$  is equal to  $c_{i'_l j_{l+1}}$ . Similarly, the unit flow cost of the arcs between the source node  $s_1$  and the original nodes in the first layer is given as  $c_{s_1 i_1}$ , while that of the arcs between the copy nodes in layer  $L$  and the sink node  $s_2$  is equal to  $c_{i'_L s_2}$ . The arcs  $(i_l, i'_l)$  in each layer  $l = 1, \dots, L$  have a zero unit flow cost, because these arcs are only used for treating vessel and waiting nodes as transshipment nodes.

The subproblem becomes an ordinary minimum cost flow problem (MCFP) on the described layered network if crane crossing is allowed (i.e. Assumption 2 given in Section 5 is relaxed), and it can be solved very efficiently (Ahuja et al., 1993). The objective function

$$\theta = \sum_{i_1=1}^{2n_1+1} c_{s_1 i_1} f_{s_1 i_1} + \sum_{l=1}^L \sum_{i_l=1}^{2n_l+1} \sum_{i'_l=1'}^{2'n_l+1} c_{i_l i'_l} f_{i_l i'_l} + \sum_{l=1}^{L-1} \sum_{i'_l=1'}^{2'n_l+1} \sum_{i_{l+1}=1}^{2n_{l+1}+1} c_{i'_l i_{l+1}} f_{i'_l i_{l+1}} + \sum_{i'_L=1}^{2'n_L+1} c_{i'_L s_2} f_{i'_L s_2}, \quad (14)$$

which represents the total cost of flows in the network, is essentially the total setup cost due to the crane relocations. Notice that the network has only arcs  $(i_l, i'_l)$ , therefore the second term of (14) computes only flow cost on these arcs not on all arc combinations, which can be guaranteed by setting the cost of the arcs different from  $(i_l, i'_l)$  to arbitrarily large values. The total flow cost is minimized subject to the flow balance equations

$$\sum_{i_1=1}^{2n_1+1} f_{s_1 i_1} = K \quad (15)$$

$$f_{i_l i'_l} - \sum_{i'_{l-1}=1'}^{2'n_{l-1}+1} f_{i'_{l-1} i_l} = 0 \quad i_l = 1, \dots, 2n_l + 1; l = 2, \dots, L \quad (16)$$

$$\sum_{i_{l+1}=1}^{2n_{l+1}+1} f_{i'_l i_{l+1}} - f_{i_l i'_l} = 0 \quad i'_l = 1', \dots, 2'n_l + 1; l = 2, \dots, L - 1 \quad (17)$$

$$- \sum_{i'_L=1}^{2'n_L+1} f_{i'_L s_2} = -K, \quad (18)$$

and lower and upper bounds on the flow variables

$$0 \leq f_{s_1 i_1} \leq K \quad i_1 = 1, \dots, 2n_1 + 1 \quad (19)$$

$$\underline{u}_{i_l i'_l} \leq f_{i_l i'_l} \leq \bar{u}_{i_l i'_l} \quad i_l = i'_l = 1, 2, \dots, 2n_l + 1; l = 2, \dots, L - 1 \quad (20)$$

$$0 \leq f_{i'_L s_2} \leq K \quad i'_L = 1', \dots, 2'n_L + 1, \quad (21)$$

which are set to  $\underline{u}_{i_l i'_l} = \bar{u}_{i_l i'_l} = g_{il}$  for  $i_l = i'_l = 2, 4, \dots, 2n_l$ ,  $\underline{u}_{i_l i'_l} = 0$  and  $\bar{u}_{i_l i'_l} = h_{i_l}$  for  $i_l = i'_l = 1, 3, \dots, 2n_l + 1$ , for calculating  $\theta^m$ , as mentioned earlier. Briefly, the determination of  $\theta^m$  for the new optimality cut consists of solving the MCFP formulated as the minimization of (14) subject to (15)–(21), when crossing of crane relocation paths (i.e. crossing arcs of the distinct  $s_1, s_2$ -flow paths) is allowed.

Unfortunately, this is not possible in reality; quay cranes are restricted to move on a single line and thus the relocation paths cannot cross, as also stated with Assumption 2 in Section 5. In other words, it is not enough to solve the MCFP, which consists of the minimization of (14) subject to the constraints (15)–(21), to compute  $\theta^m$ . Additional constraints preventing arc crossings must be considered. As a consequence, it is possible to say that CSSP is in fact equivalent to a MCFP with additional constraints allowing only noncrossing arcs to have positive flow values in an optimal solution. This is what we call a minimum cost noncrossing flow problem (MCNFP).

### 5.2.1. Reducing the number of crossings

An optimal solution of the MCFP defined by (14)–(21) may have crossings. We propose a branch-and-bound algorithm that implicitly considers additional constraints and eliminates all crossings between the paths of the cranes occurring in an optimal solution. The efficiency of this algorithm, which eventually solves the MCNFP of CSSP, depends on the number of crossings that can occur in an optimal solution of MCFP, and may be considerably improved if some of the crossings can be detected and prevented in advance during the construction of the network. This is in fact possible as the following proposition shows.

**Proposition 3.** *Arc  $(i'_{l-1}, j_l)$  can be removed from network  $\mathcal{N}$  if the demand of vessel nodes lying below (above) node  $j_l$  in layer  $l$  is larger than the maximum possible inflow to node  $i'_{l-1}$  and all other nodes below (above) it in layer  $l - 1$ .*

*Proof.* Given in the Appendix. □

Although it provides a sufficient condition for an arc to be crossed, all possible crossings cannot be prevented in an optimal solution of the MCFP by the condition described in the proposition. Nevertheless, it can reduce the number of the crossings by neglecting arcs in the network. As a result, the efficiency of the branch-and-bound method may increase not only because the network size (i.e. the number of arcs) becomes smaller, but also a significant number of potential branches (i.e. crossings) of the branch-and-bound tree can be eliminated a priori. Our implementation includes a preprocessing phase that uses this rule.

### 5.2.2. A Polynomially Solvable Special Case

Let us assume that the nonnegative setup costs are symmetric and additive, namely



$$c_{lk} = c_{kl}, \quad (22)$$

$$c_{lk} = \sum_{j=l}^{k-1} c_{j(j+1)}, \quad (23)$$

for an ordering  $p_1, p_2, \dots, p_k$  of crane positions along the berth, where  $p_1$  and  $p_k$  are, respectively, the nearest and the farthest positions with respect to the beginning of the berth. Here,  $c_{ij}$  is the setup (relocation) cost associated with a crane that finishes its work on a vessel berthed at position  $p_i$  and starts to work on a vessel berthed at position  $p_j$ .

**Proposition 4.** *If setup costs are symmetric and additive, the MCFP has an optimal solution with no crane path crossings.*

*Proof.* Given in the Appendix. □

Because of Proposition 4 it is possible to clean-up all the crossings in an optimal solution of the MCFP and obtain an alternate optimal solution in polynomial time (i.e.  $O(VL)$  where  $V$  and  $L$  are respectively the number of vessels and layers) when the setup costs are symmetric and additive. Notice that the setup costs are symmetric and additive when they are proportional to crane travel distances or times, which makes this proposition widely applicable in practice. Furthermore, the quay crane scheduling problem studied in Ak (2008) can be obtained by

1. redefining vessels as berth sections that require work,
2. considering time periods instead of intervals,
3. allowing at most one crane to work at any berth section, namely sending exactly one crane to any berth section that requires work in any period,
4. setting  $c_{ij} = |i - j|$ , which is the number of berth sections between sections  $i$  and  $j$ .

### 5.2.3. Branch-and-Bound Using the Minimum Cost Flow Relaxation

Recall that each crane is labeled according to its initial position along the berth, starting from the one nearest to the beginning of the berth. We say that two vessels are properly aligned in a time interval if the label of the rightmost crane assigned to the vessel in the left position is less than the label of the leftmost crane assigned to the vessel in the right position. Notice that the *proper alignment* of the berthed vessels in an optimal solution of the MCFP corresponds to an optimal solution with no crossings. Unfortunately, such an optimal solution is not guaranteed in general. Hence, we propose the branch-and-bound scheme given below as Algorithm 2.

Algorithm 2 essentially corrects the crossings between the paths of the cranes that occur in an optimal solution of the MCFP. Obviously, the paths of two cranes can cross more than once. Since path crossings imply arc crossings between two consecutive layers of the network on the path of two cranes, it is possible to focus on a single crossing each time there is need for branching at a node of the branch-and-bound tree. The next proposition shows that this branching strategy works.

---

**Algorithm 2**

---

**Initialization**  $v \leftarrow 1, LB = 0, UB = \infty$

Step 1:

Solve the problem associated with node  $v$  of the branch-and-bound tree

**if** the problem associated with node  $v$  of the branch-and-bound tree is infeasible **then**

    Remove node  $v$  from *node list*

**end if**

$l = 2$

**while**  $l \leq L$  **do**

**for** each copy (vessel or waiting) node  $i'_{l-1}$  in layer  $l - 1$  **do**

**for** each positive flow  $f_{i'_{l-1}j_l}$  to an original (vessel or waiting) node  $j_l$  in layer  $l$  **do**

**if** there is also a positive flow  $f_{k'_{l-1}m_l}$  from a node  $k'_{l-1}$  with  $k_{l-1} < i_{l-1}$  (equivalently  $k'_{l-1} < i'_{l-1}$ ) to node  $m_l$  with  $m_l > j_l$ , or from a node  $k'_{l-1}$  with  $k_{l-1} > i_{l-1}$  (equivalently  $k'_{l-1} > i'_{l-1}$ ) to node  $m_l$  with  $m_l < j_l$  **then**

                Remove node  $v$  from *node list*

                Add nodes  $|node\ list| + 1, |node\ list| + 2$  to *node list* with constraints  $f_{i'_{l-1}j_l} = 0, f_{k'_{l-1}m_l} = 0$  respectively

                Record the optimal objective value at node  $v, Z^v$  as the value of the nodes  $|node\ list| + 1, |node\ list| + 2$ . Go to step 2.

**end if**

**end for**

**end for**

    Increment the layer counter by setting  $l \leftarrow l + 1$

**end while**

The solution of the node is feasible. Remove node  $v$  from *node list*

**if** ( $UB > Z^v$ ) **then**

$UB \leftarrow Z^v, f^* \leftarrow f$

**end if**

Step 2:

Let  $\underline{Z}$  be the minimum of the values of the nodes in *node list*.

**if**  $\underline{Z} > LB$  **then**

$LB \leftarrow \underline{Z}$

**end if**

**if** the value of a node is larger than or equal to  $UB$  **then**

    remove that node from the *node list*

**end if**

**if**  $LB = UB$  **then**

$f^*$  is optimal; Stop

**end if**

Find node  $v$  in *node list* with the minimum value

**if**  $|node\ list|$  is zero **then**

$f^*$  is optimal; Stop

**else**

    Execute Step 1

**end if**

---

**Proposition 5.** *An optimal crane schedule that satisfies the proper alignment constraint can be obtained by means of Algorithm 2.*

*Proof.* Given in the Appendix. □

### 5.3. Shortest Path Formulation

To formulate the CSSP as a shortest path problem we construct the layered network  $\overline{\mathcal{N}} = (\overline{\mathcal{V}} \cup \{\overline{s}_1, \overline{s}_2\}, \overline{\mathcal{A}})$ . The layers represent again the  $L$  intervals during which there is no change in the berthed vessels. Each node in  $\overline{\mathcal{V}}$  in layer  $l$  is for a crane-to-vessel assignment combination and arcs in  $\overline{\mathcal{A}}$  are for the relocations between the layers. The nodes  $\overline{s}_1$  and  $\overline{s}_2$  are, respectively, the source and sink nodes. There is an arc connecting  $\overline{s}_1$  to the nodes of layer 1 and the nodes of layer  $L$  to  $\overline{s}_2$ . Each combination represents a sequence of the cranes along the berth after they are allocated to the vessels subject to the non-crossing constraints.

Note that the network  $\overline{\mathcal{N}}$  is a representation of the states of dynamic programming (DP) formulation of (Park and Kim, 2003), which we briefly describe here to serve as a benchmark for our novel branch-and-bound algorithm explained in the previous section. Intervals are the stages and the sequence of the assigned cranes in an interval are the states. Stages are connected with the changes in the crane sequences; each has a cost equal to the sum of the setup cost it requires. The recursion given in Park and Kim's formulation can be restated as

$$TSC(l) = \min_{i=1,2,\dots,S_l; j=1,2,\dots,S_{l+1}} \{SC(i, j) + TSC(l+1)\} \quad l = 0, 1, \dots, L. \quad (24)$$

Here,  $S_l$  and  $S_{l+1}$  are the labels of non-crossing sequences at layers  $l$  and  $l+1$ ,  $TSC(l)$  is the minimum total setup cost for periods  $l, l+1, \dots, L$  and  $SC(i, j)$  is the cost of the setups required to obtain crane sequence  $j$  from sequence  $i$ , which can be calculated by summing up the individual costs associated with the setups transforming sequence  $i$  to sequence  $j$ . Clearly, layers 0 and  $L+1$  have only single nodes, namely  $\overline{s}_1$  and  $\overline{s}_2$  representing initial and terminal crane sequences. Hence,  $TSC(L+1) = 0$  and the minimum total setup cost  $\theta^n = TSC(0)$ .

The size of the network  $\overline{\mathcal{N}}$  depends on the number of non-crossing sequences (i.e., states)  $S_l$ , which gives the number of vertices at layer  $l$  because it has  $\sum_{l=1}^L S_l + 2$  vertices and  $\sum_{l=1}^{L-1} S_l S_{l+1} + S_1 + S_L$  arcs, which can be very large as a consequence of the following proposition.

**Proposition 6.** *Let  $|\overline{\mathcal{V}}|$  and  $|\overline{\mathcal{A}}|$  denote the size of  $\overline{\mathcal{V}}$  and  $\overline{\mathcal{A}}$  respectively, and let  $n_l$  denote the size of  $\mathcal{V}_l$ , the set of vessels berthed in interval  $l$ . Then,*

$$|\overline{\mathcal{V}}| = \sum_{l=1}^L \binom{K - \sum_{i \in \mathcal{V}_l} g_{il} + n_l}{n_l} \quad (25)$$

$$|\overline{\mathcal{A}}| = \sum_{l=0}^L \binom{K - \sum_{i \in \mathcal{V}_l} g_{il} + n_l}{n_l} \binom{K - \sum_{i \in \mathcal{V}_{l+1}} g_{il+1} + n_{l+1}}{n_{l+1}}, \quad (26)$$

with the convention that  $\mathcal{V}_0 = \mathcal{V}_{L+1} = \emptyset$ .

*Proof.* Given in the Appendix. □

In order to clarify the implication of the above proposition let us consider the pathological example where the number of vessels  $V$  is equal to the number of cranes  $K$ , which is equal to the number of intervals  $L$ . Let us assume that there is exactly one new arrival at the beginning of each interval and vessels stay berthed until the end of the planning horizon. Let us also assume that each vessel demands exactly one crane per interval. In other words, in interval  $l$  there are  $l$  vessels and they are served by  $l$  cranes for  $l = 1, 2, \dots, L$ . Then,  $|\overline{V}| = \sum_{l=1}^K \binom{K}{l} = 2^K$ . Also,  $|\overline{\mathcal{A}}| = \sum_{l=0}^{K-1} \binom{K}{l} \binom{K}{l+1} + K$ , which is larger than  $\sum_{l=0}^{K-1} \frac{K}{K-l} \binom{K-1}{l} \binom{K}{l+1}$ . This value is clearly larger than  $\sum_{l=0}^{K-1} \binom{K-1}{l} \binom{K}{l+1}$ , which is equal to  $\binom{2K-1}{K}$ . In short, for the given example  $|\overline{\mathcal{A}}| \geq \binom{2K-1}{K}$ .

## 6. Algorithm and Model Improvements

In this section we present some ideas that improve the performance of our new method. In order to decrease the number of cuts generated in Algorithm 1 we relate  $\theta$  with  $z_{iqt}$  variables. The idea is to set a lower bound on the setup cost  $\theta$  using the number of cranes assigned to each vessel  $i$  in each time period  $t$ . The change in the number of cranes assigned to a vessel in two successive periods, or the decrease in the total number of cranes used by the vessels in two successive periods, sets a lower bound on the total setup cost. Crane assignments that give large setup costs are not chosen by Algorithm 1 because inequalities (27) and (28) that yield the lower bound (29) on the setup cost  $\theta$  enforce it to take high values for these crane assignments.

Notice that if the number of cranes used by vessel  $i$  increases by  $\rho_{it}$  from period  $t-1$  to period  $t$ , the number of setups should increase by  $\rho_{it}$ . Moreover, if the total number of cranes used by the vessels decreases by  $\psi_t$  from period  $t-1$  to period  $t$ , a setup cost incurs again because at least  $\psi_t$  cranes should stop serving vessels. It is important to realize that if we sum all  $\rho_{it}$  and  $\psi_t$ , there is no double counting of setups because  $\rho_{it}$  is related to the assignment of new cranes to vessel  $i$  and  $\psi_t$  is related to the movement of cranes from a vessel to become idle at a waiting area. In other words,

$$\rho_{it} \geq \sum_{q=0}^{\bar{k}_i} qz_{iqt} - \sum_{q=0}^{\bar{k}_i} qz_{iq(t-1)} \quad i = 1, \dots, V; t = e_i + 1, \dots, T \quad (27)$$

$$\psi_t \geq - \sum_{\substack{i=1 \\ t \geq e_i}}^V \sum_{q=0}^{\bar{k}_i} qz_{iqt} + \sum_{\substack{i=1 \\ t-1 \geq e_i}}^V \sum_{q=0}^{\bar{k}_i} qz_{iq(t-1)} \quad t = 2, \dots, T, \quad (28)$$

from which the following lower bound on  $\theta$  is obtained:

$$\theta \geq C_{\min} \left( \sum_{i=1}^V \sum_{q=0}^{\bar{k}_i} qz_{iqe_i} + \sum_{i=1}^V \sum_{t=e_i+1}^T \rho_{it} + \sum_{t=2}^T \psi_t \right). \quad (29)$$

Here,  $C_{\min}$  is the smallest unit setup cost, i.e.  $C_{\min} = \min_{i,j} \{C_{ij}\}$ .

We propose an efficient strategy in order to accelerate Algorithm 1. At each iteration, instead of solving BACAP exactly, we first determine a feasible solution and calculate the corresponding real setup cost. Then, we add the optimality cut to the problem if the real setup cost associated with the cut is larger than the setup cost estimated by

the current optimal  $\theta$  value. Notice that if the real setup cost corresponding to the cut is equal to the setup cost given by the current master problem, the cut is not added to the master problem and the current master problem is used for finding the next cut. The solution process continues from the last stopping point. We update the upper bound if the new cut provides a better one. Since the master problem is solved at each iteration, we obtain a lower bound for the problem as well. In order to define the optimality cuts we use the number of cranes assigned to vessel  $i$  in period  $t$  with the binary variable  $z_{igt}$ . Yet another improvement is possible when we use the integer variable  $y_{it}$ , which is the number of cranes assigned to vessel  $i$  in period  $t$  instead of  $z_{igt}$  while solving the main problem without any cuts.

We also propose the following enhancement for our branch and bound based CSSP solution procedure. Notice that the definition of our setup cost is general; it also covers costs that are not necessarily proportional to the travel distance or time. However, even for cases where the setup cost is not proportional to the travel distance or time, we slightly perturb the cost figures with very small costs proportional to the travel distance. That way, among the alternative solutions of the minimum cost flow problem the ones where cranes travel less (i.e. with fewer crossings) are favored. Consequently, the number of nodes in the branch-and-bound tree can be decreased. Notice that the perturbing cost figures should be small enough so that there is no unwanted change in the desired real cost structure which may not be proportional to the travel distance or time.

## 7. Computational Study

In this section we perform computational tests in order to assess the performance of the new model and solution methods. They can be grouped in two major sets. The first one is realized with a subset of the test instances listed in Table 4. They are given in (Zhang et al., 2010) and belong to a real container terminal. Also, they have been used before for the same purpose in some of the works, and thus they can be treated as standard benchmark problems. We select seven instances with  $V = 3, 6, 9, 12, 15, 18, 21$  vessels. The second set consists of larger test problems. They are generated randomly, mainly for pushing the new methods to their limits.

In Table 4,  $s_i$  and  $e_i$  refer to the desired berth section and arrival time of vessel  $i$ , respectively.  $r_i$  refers to the due time of vessel  $i$ . For instance  $s_1 = 13$  means that the desired berthing section of the first vessel is the beginning of berth section 13. Similarly,  $e_1 = 1$  means that the arrival time of the first vessel is the beginning of the first time period. The cost coefficients associated with berthing away from the desired berth section, late berthing, and late departure are selected as  $\phi_i^s = 1000$ ,  $\phi_i^e = 1000$ , and  $\phi_i^r = 2000$ . We assume that the setup costs related to crane relocations are all equal and have value 50. As mentioned previously, this choice does not make CSSP polynomially solvable since the additivity condition of the costs is violated. In all instances there are  $K = 12$  cranes,  $B = 24$  berth sections each having a length of 50 meters, and  $T = 200$  periods each corresponding to one hour. The interference exponent  $\lambda$  is set to 0.95 during the experiments, except the ones whose results are reported in Table 11 where  $\lambda = 1$ . The experiments are carried out on a computer with Intel Xeon 3.16 GHz processor and 32 GB of RAM working under Windows 2003 Server operating system. We run commercial solver CPLEX 12.6 on single core to solve linear and integer programming problems.

Table 4: Realistic vessel data given in (Zhang et al., 2010)

Vessel	$\ell_i$	$\underline{k}_i$	$\bar{k}_i$	$s_i$	$e_i$	$r_i$	$w_i$	Vessel	$\ell_i$	$\underline{k}_i$	$\bar{k}_i$	$s_i$	$e_i$	$r_i$	$w_i$
1	3	2	2	13	1	21	18	12	5	2	4	17	78	104	66
2	7	3	6	2	1	26	96	13	7	3	6	15	86	112	109
3	3	2	2	14	2	16	11	14	8	3	7	10	90	112	84
4	8	3	7	11	2	25	113	15	5	2	4	19	94	105	22
5	6	3	5	12	18	51	113	16	4	2	3	16	102	122	31
6	4	2	3	13	22	41	25	17	4	2	3	12	110	129	36
7	7	3	6	9	35	46	18	18	7	3	6	6	122	149	108
8	8	3	7	5	38	54	27	19	3	2	2	18	122	136	13
9	6	3	5	6	50	79	114	20	4	2	3	1	152	183	49
10	6	3	5	16	54	88	105	21	5	2	4	10	152	168	19
11	4	2	3	4	78	100	30								

### 7.1. The Efficiency of the Branch-and-bound Method for the Solution of CSSP

In the first part of our computational study we compare the efficiencies of the branch-and-bound (BB) and shortest path (SP) algorithms that solve CSSP on very large randomly generated layered networks. The results reported in Table 5, where the first two columns contain information about the size of the layered network. The first one shows the number of layers in the test instances. The second one includes the maximum number of nodes in each layer, which is generated randomly between the half of the maximum and maximum number of nodes. The crane requirement of each vessel is randomly generated between 1 and 10 and the available number of cranes is assigned in accordance with the crane requirements. For each layer the size and number of nodes, we generate five instances. The last two columns report the average CPU Times. As can be noticed BB is faster than SP. The difference becomes significant with the increasing instance sizes. This is in harmony with the theoretical worst case analysis of SP summarized with Proposition 6.

Table 5: Efficiency of the BB and SP algorithms for solving CSSP: CPU time in seconds

$L$	Max. no. of nodes in each layer	CPU Time	
		BB	SP
50	10	0.9	1.5
50	20	9.4	17.2
50	30	106.6	235.2
50	40	758.2	1316.9
50	50	4703.5	12642.3
100	60	18815.3	36783.6
100	70	28222.6	55981.4
100	80	34862.5	72982.3
Column average		10934.9	22495.1

We should point out that the sizes of the layered networks corresponding to the CSSP of the largest BACASP instances we could solve exactly is even smaller than the smallest layered networks considered in this comparison. As a result, one can be misled by the insignificance of the differences between the values reported for BB and SP

algorithms in Table 7 and Table 10 and may think that both methods are equally efficient. However, this is not the case, as can be observed.

### 7.2. The Effect of the Valid Inequality (29) on the Efficiency of the Decomposition Algorithm

In the second part we investigate the contribution of inequality (29) presented in Section 6. For this purpose, we solve the test instances with and without this inequality. The second column of Table 6 shows the optimal value in each test instance. The other columns show the number of iterations, the number of cuts and the total CPU time when the decomposition algorithm is executed with and without the inequality. Table 6 shows that the number of iterations and the total CPU Time decreases drastically when the cutting plane algorithm is executed with the valid inequality (29) presented in Section 6.

Table 6: Optimal cost and the number of cuts generated for the selected test instances from Zhang et al. (2010)

V	Optimal Cost	with inequality (29)			without inequality (29)		
		No. of iterations	No. of cuts	Total CPU Time	No. of iterations	No. of cuts	Total CPU Time
3	2800	6	5	10.7	49	45	110.5
6	14,850	40	38	1,293.1	391	382	13,232.5
9	17,550	11	7	543.5	83	79	5,632.8
12	20,550	36	33	2,951.7	321	315	33,546.7
15	35,250	49	46	4,482.3	467	462	46,754.9
18	39,950	41	38	4,543.2	401	396	46,385.6
21	41,700	72	67	10,493.6	736	732	113,215.3
Column average		36.43	33.43	3,474.01	349.71	344.43	36,982.61

Table 7 shows the average CPU time in seconds per optimality cut to solve the main model, the CPU time to solve CSSP by BB algorithm and SP, and the size of the BB tree averaged over the number of cuts added to the master problem. Tree size statistics is only reported for BB since SP also resulted in exactly the same values. We see that the proposed solution methodology can optimally solve all the instances presented in Zhang et al. (2010). To the best of our knowledge, exact optimal solutions could not be obtained for problems of this size before. We see that the largest instance with 21 vessels can be solved in around three hours.

Table 7: Performance of the new method: CPU time in seconds on realistic instances selected test instances from Zhang et al. (2010)

V	Master Problem	CSSP with		Overall Problem with		No. of nodes in the BB tree	
		BB	SP	BB	SP		
3	2.7	0.01	0.02	10.7	10.8	6	
6	35.1	0.01	0.02	1,293.1	1,293.8	13	
9	77.9	0.02	0.03	543.5	544.1	19	
12	95.2	0.02	0.04	2,951.7	2,954.3	25	
15	99.6	0.03	0.04	4,482.3	4,484.8	29	
18	119.6	0.03	0.05	4,543.2	4,545.1	32	
21	159.2	0.04	0.06	10,493.6	10,496.8	35	
Column average		84.19	0.024	0.037	3,474.01	3,475.67	22.71

### 7.3. The Efficiency of the Decomposition Algorithm

The third part of the computational study starts with Figure 3 which depicts the berth time diagram corresponding to an optimal solution for the 21-vessel instance. Table 8 reports the cost components along with the specific crane assignments for each vessel. To clarify the notation, as an example (C1-C5)[11-17] in the fifth column of Table 8 means that cranes from one to five are assigned to vessel two in time periods from eleven to seventeen.

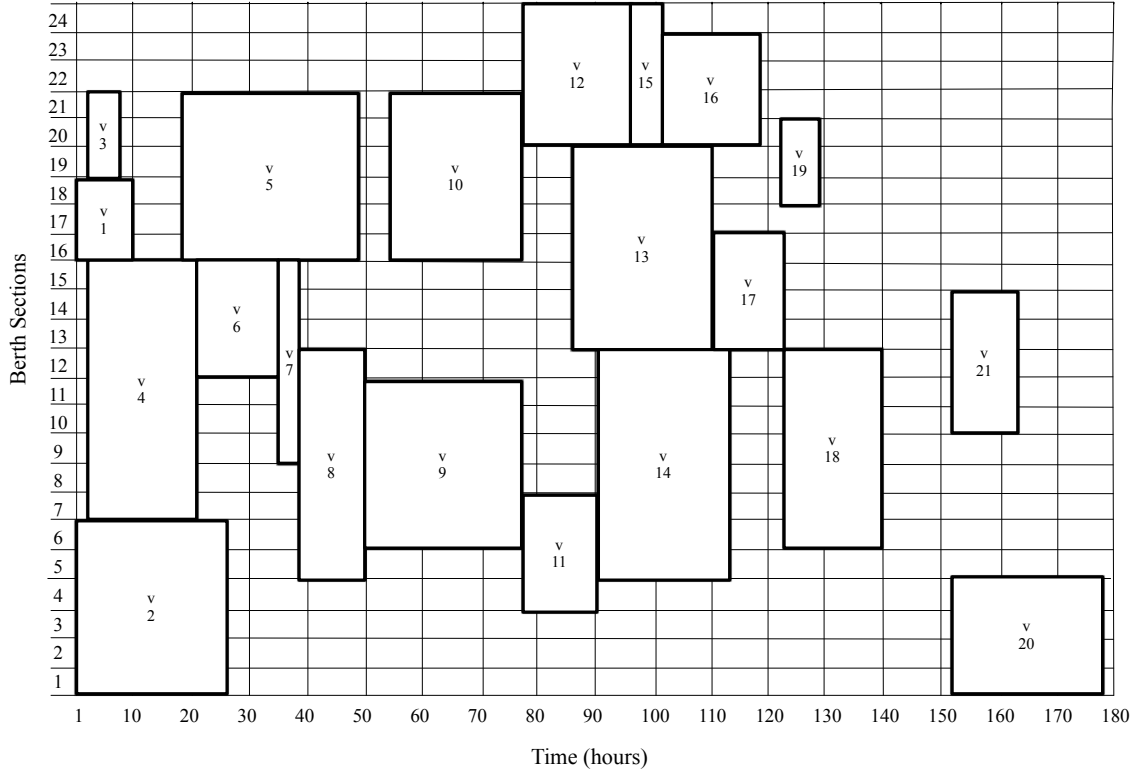


Figure 3: Optimal solution's berth time diagram for the 21-vessel instance of Zhang et al. (2010)

Based on our observations during the computational tests on the test instances selected from Table 4, we can say that there is a direct relation between the solution times and sizes of the instances. As a consequence, in order to test the performance of the new approach better we have generated larger ones randomly. We vary the number of vessels  $V$  from 20 to 60.  $T$  is 400, 500, and 600 when  $V$  takes on values between 20 and 40, 45 and 50, and 55 and 60, respectively. The length of the berth, the number of cranes available and the cost coefficients are the same as explained in the second paragraph of Section 7. The vessel-dependent parameters, namely length ( $\ell_i$ ), desired berth section ( $s_i$ ), workload ( $w_i$ ), lower bound ( $\underline{k}_i$ ), and difference between the bounds ( $\bar{k}_i - \underline{k}_i$ ) on the crane numbers are generated from uniform distributions respectively between 3 and 8 berth sections, 1 and  $B - \ell_i + 1$ , 10 and 120, 1 and 5, and 0 and 5. For each number of vessels we generate five instances. Table 9 shows the average number of iterations and the average number of cuts generated by Algorithm 1 for each group of five large test instances.

Table 10 reports the average CPU times spent for solving the master problem and the overall problem over five instances, for each group. We report CPU times for solving the master problem, CSSP and the overall problem when



Table 8: Cost components for each vessel in the 21-vessel instance of Zhang et al. (2010)

Vessel	$\phi_i^s \times  j - s_i $	$\phi_i^e \times (t - e_i)$	$\phi_i^r \times \max(0, t' - r_i)$	specific crane assignments
1	3,000	0	0	(C9-C10)[1-7],(C11-C12)[8-10]
2	1,000	0	0	(C1-C4)[1],(C1-C3)[2-10],(C1-C5)[11-17], (C1-C3)[18-21],(C1-C5)[22-26]
3	5,000	0	0	(C11-C12)[2-7]
4	3,000	0	0	(C4)[2-7],(C4-C10)[8-10],(C6-C12)[11-17], (C4-C9)[18-21]
5	4,000	0	0	(C10-C12)[18-21],(C8-C12)[22-34],(C10-C12)[35-49]
6	1,000	0	0	(C6-C7)[22-34]
7	0	0	0	(C5-C9)[35-38]
8	0	1,000	0	(C7-C9)[39-48]
9	0	0	0	(C2-C6)[50-77]
10	0	0	0	(C8-C12)[54-77]
11	0	0	0	(C2-C4)[78-85],(C2-C3)[86-89]
12	3,000	0	0	(C9-C12)[78-95]
13	2,000	0	0	(C4-C8)[86-101],(C6-C10)[102-109]
14	5,000	0	0	(C1-C3)[90-101],(C1-C5)[102-112]
15	1,000	2,000	0	(C9-C12)[96-101]
16	4,000	0	0	(C11-C12)[102-118]
17	1,000	0	0	(C8-C10)[110-121],(C9-C10)[122]
18	0	0	0	(C6-C7)[122],(C6-C10)[123-149]
19	0	0	0	(C11-C12)[122-128]
20	0	0	0	(C6-C7)[152-177]
21	0	0	0	(C9-C10)[152-161]

Table 9: Average number of cuts per group of test instances

V	Number of iterations	Number of Cuts
20	73	70
25	96	92
30	138	133
35	173	168
40	199	194
45	230	224
50	259	252
55	295	290
60	341	335

BB or SP is used, and the number of nodes in the BB tree when CSSP is solved with BB; these values are exactly the same with SP. We can observe that very large problem instances with  $T = 600$  and  $V = 60$  can be solved to optimality. This is a considerable improvement when compared with the performance of the existing exact models in the literature.

#### 7.4. The Effect of Time-variance on the Solution Quality

The last part of the computational study aims to give an idea on the contribution of the flexibility introduced by time-variance of crane allocations. Recall that Türkoğulları et al. (2014) present a time-invariant model for BACAP,

Table 10: Performance of the new method: CPU time in seconds on large instances

$V$	Master Problem	CSSP with		Overall Problem with		No. of nodes in the BB tree
		BB	SP	BB	SP	
20	124.5	0.04	0.05	8,585.2	8,587.3	42
25	163.2	0.04	0.06	14,845.7	14,847.9	55
30	218.0	0.05	0.06	28,768.2	28,771.1	57
35	226.9	0.06	0.07	37,641.3	37,644.2	66
40	248.3	0.06	0.07	47,893.9	47,897.2	69
45	298.2	0.07	0.08	66,192.6	66,196.4	75
50	349.4	0.07	0.08	87,690.7	87,695.1	79
55	374.5	0.07	0.09	108,223.3	108,228.3	86
60	420.8	0.08	0.09	140,505.5	140,511.2	89
Average	346.26	0.06	0.07	77,178.49	77,196.95	88.29

where the number of cranes assigned to a vessel does not change in time and there is no setup cost. Although this approach helps to reduce crane relocations, it is expected to increase the other cost components, i.e., the cost of berthing away from the desired berthing section, the cost of berthing later than the arrival time, and the cost of departure later than the due date. To analyze the magnitude of the reduction in the above-mentioned cost components, we solve the instances with  $V = 3, 6, \dots, 21$  vessels, selected from Table 4, using the new model and the time-invariant model of Türkoğulları et al. (2014). The results are given in Table 11. Based on these results, we can say that the optimal cost of the new model is significantly less than that of the time-invariant model in all test instances. Notice that Zhang et al. (2010) report 36,000 as the best objective value for the 21-vessel instance. The objective value found by our model is smaller than that of Zhang et al. (2010) because they calculate this value using a heuristic.

Table 11: The comparison of the time-variant and invariant models

$V$	Optimal cost	
	Proposed model with zero setup costs	Time-invariant model of Türkoğulları et al. (2014)
3	2,000	2,000
6	13,000	21,000
9	15,000	21,000
12	15,000	21,000
15	27,000	35,000
18	30,000	43,000
21	30,000	43,000
Average	18,857.14	25,571.43

## 8. Conclusions

In this paper we formulate a new mixed integer linear program to obtain an optimal solution of berth allocation, quay crane assignment and scheduling problems simultaneously. We devise a decomposition algorithm in order to solve the problem efficiently based on the separate solution of the scheduling subproblem at each step. We first for-

mulate the quay crane scheduling problem as a minimum cost network flow problem where arc crossing is allowed. In order to obtain a crane schedule without any arc crossing we propose a branch-and-bound algorithm. Then, we develop another method and solve the quay crane scheduling problem by finding the shortest path on a layered network representing the states of a dynamic programming formulation from the literature. Our computational study demonstrates that the new formulation coupled with the proposed solution method gives optimal solutions for realistic size large problem instances. Also it turns out that the branch-and-bound based method is more efficient than the shortest path based method in solving the quay crane scheduling problem. As a further study, a challenging direction can be the extension of the new mathematical optimization model by means of new constraints and objective function terms with the aim of integrating the storage and yard operations with seaside operations. As we can see in Table 8, some vessels have large deviations from their desired berthing positions. Modifying the objective function in such a way that it evenly distributes the deviations among the vessels may be another promising research direction. Finally, although we have shown that CSSP is NP-complete, we conjecture that it is in fact strongly NP-complete, namely it is possible to devise a reduction from a known strongly NP-hard problem.

**Acknowledgements**—We gratefully acknowledge the support of IBM through an open collaboration research award #W1056865 granted to the third author and partial support of the Turkish Scientific and Technological Council (TUBITAK) with Grant No. 213M441. We also would like to send our gratitudes to the anonymous referees for their comments and suggestions they made on the previous versions.

Ahuja, R. K., T. L. Magnanti, J. B. Orlin. 1993. *Networks Flows*. Prentice-Hall.

Ak, A. 2008. Berth and quay crane scheduling: problems, models and solution methods. Ph.D. thesis, Georgia Institute of Technology.

Benders, J. F. 1962. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik* **4** 238–252.

Bierwirth, C., F. Meisel. 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research* **202**(3) 615–627.

Bierwirth, C., F. Meisel. 2015. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research* **244** 675–689.

Carlo, H.J., I.F.A. Vis, K.J. Roodbergen. 2013. Seaside operations in container terminals: literature overview, trends and research directions. *Flexible Services and Manufacturing Journal* **June** 1–39.

Dragovic, B., N. K. Park, Z. Radmilovic. 2006. Ship-berth link performance evaluation: simulation and analytical approaches. *Maritime Policy and Management* **33** 281–299.

Garey, M. R., D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. John-Wiley.

Geoffrion, A. 1994. Structured modeling: survey and future research directions. *ORSA CSTS Newsletter* **15**(1) 11–20.

Imai, A., H.C. Chen, E. Nishimura, S. Papadimitriou. 2008. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E* **44**(5) 900–920.

Laporte, G., F. V. Louveaux. 1993. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters* **13** 133–142.

Lee, D.-H., H. Q. Wang, L. Miao. 2008. Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E* **44** 124–135.

Lim, J. A., B. Rodrigues, F. Xiao, Y. Zhu. 2004. Crane scheduling with spatial constraints. *Naval Research Logistics* **51** 386–406.

Liu, J., Y. Wan, L. Wang. 2006. Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics* **53** 60–74.

Meier, L., R. Schumann. 2007. Coordination of interdependent planning systems, a case study. R. Koschke, O. Herzog, K-H. Rodiger, M. Ronthaler, eds., *Lecture Notes in Informatics (LNI)*. 389–396.

Meisel, F. 2009. *Seaside Operations Planning in Container Terminals*. Physica-Verlag.

- Meisel, F., C. Bierwirth. 2009. Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E* **45**(1) 196–209.
- Meisel, F., C. Bierwirth. 2013. A framework for integrated berth allocation and crane operations planning in seaport container terminals. *Transportation Science* **47**(2) 131–147.
- Oguz, C., J. Blazewicz, T. C. E. Chang, M. Machowiak. 2004. Berth allocation as a moldable task scheduling problem. *Proceedings of Ninth International Workshop on Project Management and Scheduling*. 201–205.
- Park, Y.M., K.H. Kim. 2003. A scheduling method for berth and quay cranes. *OR Spectrum* **25** 1–23.
- Rashidi, H. 2006. Dynamic scheduling of automated guided vehicles. Ph.D. thesis, University of Essex, Colchester.
- Schonfeld, P., O. Sharafeldien. 1985. Optimal berth and crane combinations in containerports. *Journal of Waterway, Port, Coastal and Ocean Engineering* **111** 1060–1072.
- Silberholz, M. B., B. L. Golden, E. K. Baker. 1991. Using simulation to study the impact of work rules on productivity at marine container terminals. *Computers and Operations Research* **18** 433–452.
- Stahlbock, R., S. Voß. 2008. Operations research at container terminals: A literature update. *OR Spectrum* **30** 1–52.
- Steenken, D., S. Voß, R. Stahlbock. 2004. Container terminal operation and operations research – a classification and literature review. *OR Spectrum* **26** 3–49.
- Theofanis, S., M. Goliás, M. Boile. 2007. Berth and quay crane scheduling: a formulation reflecting service deadlines and productivity agreements. *Proceedings of the International Conference on Transport Science and Technology (TRANSTEC 2007)*. Czech Technical University, Prague, 124–140.
- Türkoğulları, Y. B., Z. C. Taşkın, N. Aras, K. Altınel. 2012. Solving the integrated berth allocation and time-variant quay crane assignment problem in container terminals. URL <http://akademik.maltepe.edu.tr/~yavuzturkogullari/Research-Reports/Technical-report-FBE-IE-04-2012-05.pdf>. Tech. rep., Institute for Graduate Studies in Science and Engineering, Boğaziçi University, Istanbul, FBE-IE-04/2012-05.
- Türkoğulları, Y.B., Z. C. Taşkın, N. Aras, K. Altınel. 2014. Optimal berth allocation and time-invariant quay crane assignment in container terminals. *European Journal of Operational Research* **235** 88–101.
- UNCTAD. 2011. Review of maritime transport, united nations conference on trade and development. URL <http://www.unctad.org>.
- Vacca, I. 2010. Container terminal management: integrated models and large scale optimization algorithms. Ph.D. thesis, École Polytechnique Fédérale de Lausanne.
- Vis, I.F.A., R. de Koster. 2003. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research* **147** 1–16.
- Yang, K.-H. 2008. A hybrid approach for container terminal operations. Ph.D. thesis, University of Oklahoma.
- Zhang, C., L. Zheng, Z. Zhang, L. Shi, A.J. Armstrong. 2010. The allocation of berths and quay cranes by using a sub-gradient optimization technique. *Computers and Industrial Engineering* **58** 40–50.
- Zhu, Y., A. Lim. 2006. Crane scheduling with non-crossing constraint. *Journal of the Operational Research Society* **57** 1464–1471.

## APPENDIX: Proofs of the Propositions

**Proposition 1.** *CSSP-BR is NP-complete.*

*Proof.* i.  $\text{CSSP-BR} \in \text{NP}$ : If a crane schedule is given, its feasibility can be checked in polynomial time. Checking the schedule for having non-crossing paths of the cranes can be done in  $O(V^2L)$  time. Checking whether the total setup cost of each crane  $k$  is less than  $C_k$  can be done in  $O(KL)$  time. Hence, there is a polynomial-time certificate checking algorithm and  $\text{CSSP-BR} \in \text{NP}$ .

ii.  $\text{CSSP-BR}$  is hard: The *Set Partitioning Problem* (SPP), which is known to be NP-complete (Garey and Johnson, 1979), reduces polynomially to  $\text{CSSP-BR}$ . The SPP deals with the following question: Given a set  $S$  of  $W$  elements with values  $s_v \in \mathbb{Z}^+$ ,  $v = 1, 2, \dots, W$  and  $\sum_{v \in S} s_v = D$ , is there a subset  $S' \subset S$  such that  $\sum_{v \in S'} s_v = \sum_{v \in S \setminus S'} s_v = \frac{D}{2}$ ?

For  $L \geq W + K$ , an instance of CSSP-BR corresponding to an arbitrary SPP instance can have  $K$  cranes,  $W + K$  vessels with

$$g_{il} = \begin{cases} 1, & \text{if } i = 1, \dots, W + 2; l = 1, \dots, L; l = i \\ 0, & \text{if } i = 1, \dots, W + 2; l = 1, \dots, L; l \neq i \\ 1, & \text{if } i = W + 3, \dots, W + K; l = 1, \dots, L. \end{cases}$$

and  $C_k = D$ ,  $k = 1, \dots, K$ . Besides,  $c_{ij}$  which is the setup cost of relocating a crane from vessel  $i$  to vessel  $j$  can be chosen as

$$c_{ij} = \begin{cases} \frac{D}{2}, & \text{if } i = 1, \dots, W + 2; l = 1, \dots, L; j = i \\ s_i, & \text{if } i = 1, \dots, W; j = i + 1 \\ s_{j-1}, & \text{if } i = 3, \dots, W + 2; j = i - 1 \\ D, & \text{if } i = 0; j = W + 3, \dots, W + K \\ D + 1, & \text{otherwise.} \end{cases}$$

Figure 4 illustrates this transformation. It shows  $K$  cranes,  $W + K$  vessels, the setup costs and crane allocations for each vessel.

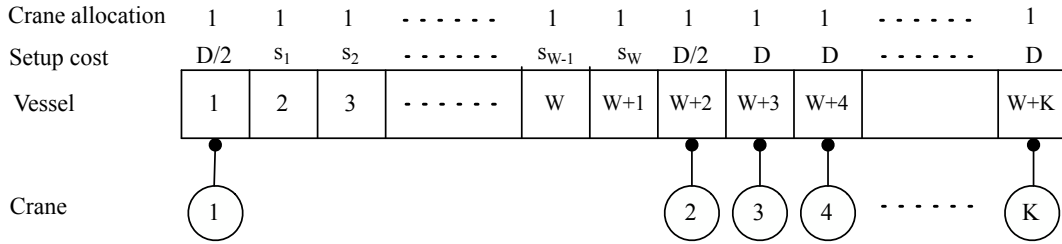


Figure 4: Transformation from SPP to CSSP-BR

Then, it must be shown that  $S$  has a subset  $S'$  such that  $\sum_{v \in S'} s_v = \sum_{v \in S \setminus S'} s_v = \frac{D}{2}$  if and only if all the  $W + K$  vessels are served by  $K$  cranes each of which spends no more than  $C_k = D$  units for setups during  $L$  time intervals without crossing.

First, suppose that  $S$  has a subset  $S'$  such that  $\sum_{v \in S'} s_v = \sum_{v \in S \setminus S'} s_v = \frac{D}{2}$ . Then,  $K$  cranes can be scheduled without any crossing as follows: crane 1 first works on vessel 1 and then on vessels  $v = 2, 3, \dots, |S'| + 1$ , where  $i_{v-1} \in S'$ ; crane 2 first serves vessel  $W + 2$ , and then vessels  $v = W + 1, W, \dots, |S'| + 2$ , where  $i_{v-1} \in S \setminus S'$ ; cranes 3 to  $K$  work on vessel  $W + 3$  to vessel  $W + K$ , respectively. Observe that there is no crossing in this schedule and the setup cost for each crane is  $D$ . Thus, if the set  $S$  has a subset  $S'$  such that  $\sum_{v \in S'} s_v = \sum_{v \in S \setminus S'} s_v = \frac{D}{2}$ , all  $W + K$  vessels can be served by  $K$  cranes with the setup cost of each crane being equal to  $D$  within  $L$  time intervals.

Conversely, suppose that all the  $W + K$  vessels can be served by  $K$  cranes (within  $L$  intervals) spending no more than  $D$  cost units for each crane. According to this, the sum of the setup costs for crane 1 and crane 2 is equal to  $2D$ , which follows from the fact that  $\sum_{v \in S} s_v = D$ . This implies that crane 1 and crane 2 fully use their budget since each of them cannot spend more than  $D$ . Besides, neither crane 1 nor crane 2 can serve both vessel 1 and vessel  $W + 2$  due

to the non-crossing constraints. Therefore,  $\sum_{v \in S'} s_v = \sum_{v \in S \setminus S'} s_v = D - \frac{D}{2} = \frac{D}{2}$ . This transformation can be done in  $O(W + K)$  time, and the proof is complete.  $\square$

**Proposition 2.** *CSSP is NP-complete.*

*Proof.* First of all, any certificate of CSSP can be checked polynomially using an algorithm similar to the one used for CSSP-BR. Hence,  $\text{CSSP} \in \text{NP}$ . Then, consider any arbitrary instance of CSSP-BR with setup costs  $c_{ij} \in \mathbb{Z}^+$ ,  $i, j = 1, 2, \dots, V; i \neq j$ , crane allocations  $g_{il} \in \mathbb{Z}^+$ ,  $i = 1, 2, \dots, V; l = 1, 2, \dots, L$ , and budget restrictions  $C_k$ ,  $k = 1, \dots, K$ . Also define  $Q = \sum_{i=1}^V \sum_{l=1}^L g_{il}$ .

To generate an instance of CSSP let us define the setup costs  $c'_{ij} = c_{ij} + \lambda$  where  $\lambda = \sum_{k=1}^K C_k$ , and choose crane allocations  $g'_{il}$ ,  $i = 1, 2, \dots, V; l = 1, 2, \dots, L$  such that  $Q' = Q + 1$  where  $Q' = \sum_{i=1}^V \sum_{l=1}^L g'_{il}$ . Let us also set  $C = C_{k^*} + \lambda Q'$  where  $C_{k^*} = \min_k \{C_k\}$ .

First, suppose that CSSP-BR has a non-crossing schedule satisfying budget restrictions and let  $P_k$  denote the route crane  $k$  follows in this schedule. In other words,  $P_k$  is a directed path of size  $|P_k|$  representing the sequence of vessels served by crane  $k$ . Clearly,  $c_{ij}$  is the setup cost associated with the adjacent pair  $(i, j) \in P_k$ . Since the budget restrictions are satisfied,

$$\sum_{(i,j) \in P_k} c_{ij} \leq C_k \quad k = 1, \dots, K,$$

and thus

$$\sum_{k=1}^K \sum_{(i,j) \in P_k} c_{ij} \leq \sum_{k=1}^K C_k.$$

Then,

$$\begin{aligned} \sum_{k=1}^K \sum_{(i,j) \in P_k} c'_{ij} &= \sum_{k=1}^K \sum_{(i,j) \in P_k} (c_{ij} + \lambda) \\ &= \sum_{k=1}^K \sum_{(i,j) \in P_k} c_{ij} + \lambda \sum_{k=1}^K |P_k| \\ &= \sum_{k=1}^K \sum_{(i,j) \in P_k} c_{ij} + \lambda Q \\ &\leq \sum_{k=1}^K C_k + \lambda Q = \lambda(Q + 1) \\ &\leq \lambda Q' + C_{k^*} = C. \end{aligned}$$

The last inequality follows from the fact that  $Q' = Q + 1$  and  $C_{k^*} > 0$ .

Conversely, suppose that CSSP has a non-crossing schedule with an overall setup cost less than  $C$ . Let  $P'_k$  denote the directed path associated with crane  $k$ 's route. Then,

$$\begin{aligned}
\sum_{k=1}^K \sum_{(i,j) \in P'_k} c'_{ij} &= \sum_{k=1}^K \sum_{(i,j) \in P'_k} (c_{ij} + \lambda) \\
&= \sum_{k=1}^K \sum_{(i,j) \in P'_k} c_{ij} + \lambda \sum_{k=1}^K |P'_k| \\
&= \sum_{k=1}^K \sum_{(i,j) \in P'_k} c_{ij} + \lambda Q' \leq C.
\end{aligned}$$

As a consequence,

$$\sum_{k=1}^K \sum_{(i,j) \in P'_k} c_{ij} \leq C - \lambda Q' = C_{k^*},$$

from which

$$\sum_{(i,j) \in P'_k} c_{ij} \leq C_{k^*} \quad k = 1, \dots, K$$

follows since  $c_{ij} \geq 0$ ,  $i, j = 1, \dots, V$ ;  $i \neq j$ , implying  $\sum_{(i,j) \in P'_k} c_{ij} \geq 0$ ,  $k = 1, \dots, K$ . Therefore,

$$\sum_{(i,j) \in P'_k} c_{ij} \leq C_k \quad k = 1, \dots, K$$

by the definition of  $C_{k^*}$ .

Finally, this transformation can be done in  $O(V^2L + VL + K + K)$  time, which is polynomial and the proof is complete.  $\square$

**Proposition 3.** *Arc  $(i'_{l-1}, j_l)$  can be removed from network  $\mathcal{N}$  if the demand of vessel nodes lying below (above) node  $j_l$  in layer  $l$  is larger than the total inflow to node  $i'_{l-1}$  and all other nodes below (above) it in layer  $l - 1$ .*

*Proof.* Suppose that there is an arc from the copy node  $i'_{l-1}$  in layer  $l - 1$  to the original node  $j_l$  in layer  $l$ . Recall that original vessel nodes in layer  $l$  have crane requirements which must be satisfied by the crane flows from the copy nodes in layer  $l - 1$ . Now, if the total crane demand of the vessel nodes below node  $j$  in layer  $l$  is larger than the total possible inflow from the copy node  $i'_{l-1}$  and other copy nodes in layer  $l - 1$  that lie below  $i'_{l-1}$ , then it is necessary that there be a flow on an arc connecting a copy node located above node  $i'_{l-1}$  in layer  $l - 1$  and a node located below node  $j_l$  in layer  $l$  to obtain a feasible solution of the MCFP. This arc, however, crosses the arc from  $i'_{l-1}$  to  $j_l$ , which is assumed to exist. Therefore, we can remove the latter arc. Similarly, if the total crane demand of the vessel nodes above node  $j$  in layer  $l$  is larger than the total possible inflow from the copy node  $i'_{l-1}$  and other copy nodes in layer  $l - 1$  that lie above  $i'_{l-1}$ , then it is necessary that there be a flow on an arc connecting a copy node located below node  $i'_{l-1}$  and a node located above node  $j_l$ . This arc will also cross the arc  $(i'_{l-1}, j_l)$ . To put it differently, in order for an arc  $(i'_{l-1}, j_l)$  to exist, the demand of vessel nodes lying below (above) node  $j_l$  must be less than or equal to the total inflow to node  $i'_{l-1}$  and all other nodes below (above) it in layer  $l - 1$ . As a consequence, the existence of either condition helps us remove the corresponding arc from further consideration.  $\square$

**Proposition 4.** *If setup costs are symmetric and additive, the MCFP has an optimal solution with no crane path crossings.*

*Proof.* Let us assume that two of the relocation paths cross in an optimal solution of the MCFP in some interval. Let us also assume that the first path is from position  $p_1$  to position  $p_2$  along the berth, the second one is from  $p_3$  to  $p_4$ , and  $p_1$  is before  $p_3$  (i.e.,  $p_1 < p_3$ ), namely closer to the beginning of the berth. Based on the structure of the network, the arcs  $(p_1, p_2)$  and  $(p_3, p_4)$  must cross, which implies that  $p_4 < p_2$ . The setup cost associated with this relocation is  $c_{12} + c_{34}$ , where  $c_{ij}$  denotes the setup cost associated with the path  $(p_i, p_j)$  path. The relocations on these paths can occur in the same or opposite directions. There are two and four possibilities when the direction is the same and opposite, respectively. They are represented in Figure 5. The positions are labeled with numbers. Solid arcs represent the crossing relocations given in the solution, whereas dashed ones represent their non-crossing equivalents.

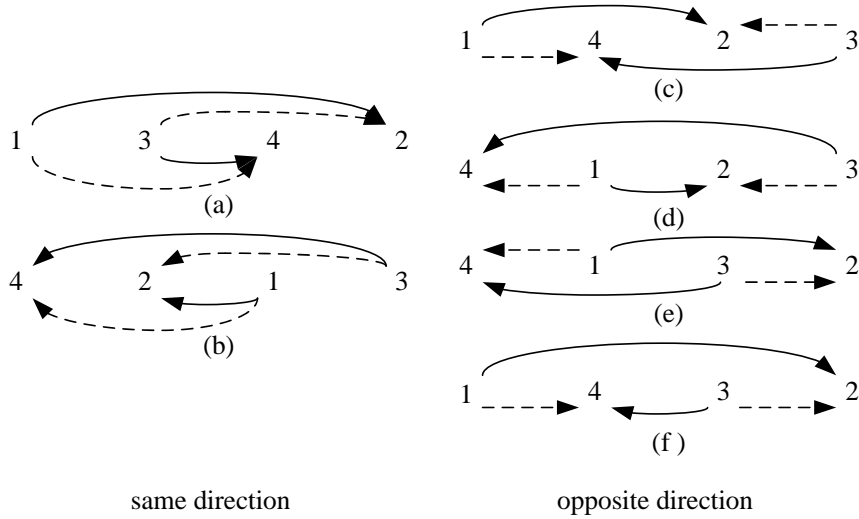


Figure 5: Crossing and non-crossing relocations

In case (a) the crossing relocations  $(p_1, p_2)$  and  $(p_3, p_4)$  are replaced with their non-crossing equivalents  $(p_1, p_4)$  and  $(p_3, p_2)$ . Observe that relocations from  $(p_3, p_2)$  are realized first. The setup cost associated with this relocation is  $c_{14} + c_{32}$ , which is clearly equal to  $c_{12} + c_{34}$  under the symmetry and additivity assumptions of the setup costs. Similarly,  $c_{34} + c_{12} = c_{14} + c_{32}$  for case (b).

Cases (c)–(f) are for the relocations in opposite directions and the setup cost associated with crossing relocations is higher than that of their non-crossing versions. For example, in case (c)  $c_{12} + c_{34} = c_{12} + c_{43} = c_{14} + c_{42} + c_{42} + c_{23} \geq c_{14} + c_{23} = c_{14} + c_{32}$ .

Let  $f_{ij}$  denote the number of cranes moving from position  $p_i$  to  $p_j$ , namely the flow on the arc  $(p'_i, p_j)$  where  $p'_i$  represents a copy node in layer  $l$  and  $p_j$  represents an original node in layer  $l + 1$  of the network given in Figure 2. It is possible to streamline a crossing arc flow by a simple operation and adjust the flows on the corresponding arcs in this network without harming its feasibility. If  $f_{12} \geq f_{34}$ , then add new arcs  $(p'_3, p_2)$  and  $(p'_1, p_4)$  with flows  $f_{34}$ , adjust the flow on arc  $(p'_1, p_2)$  by subtracting  $f_{34}$ , and finally delete arc  $(p'_3, p_4)$ . If  $f_{34} > f_{12}$ , then operate similarly by adding



new arcs  $(p'_3, p_2)$  and  $(p'_1, p_4)$  with flows  $f_{12}$ , adjust the flow on arc  $(p'_3, p_4)$  by subtracting  $f_{12}$ , and finally delete arc  $(p'_1, p_2)$ . These operations are illustrated in Figure 6.

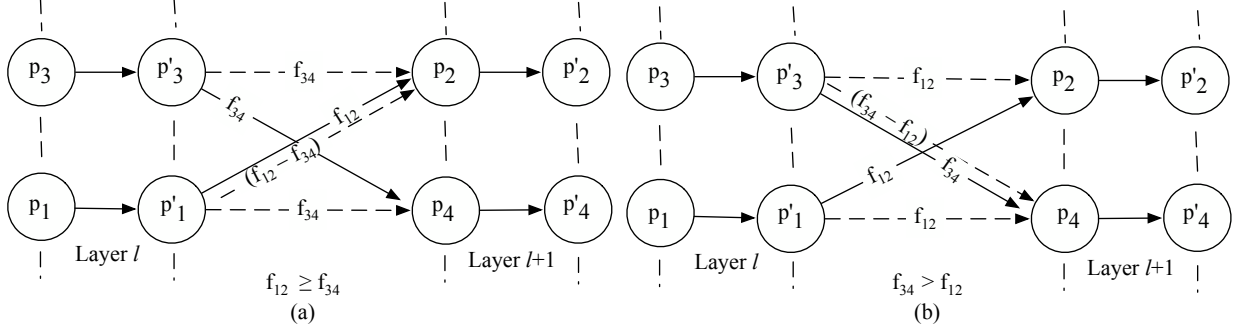


Figure 6: Crossing and non-crossing relocations

The crossing represented with the solid arcs is streamlined by the dashed arcs. Observe that flow balance is preserved at all nodes and capacity restrictions are satisfied on the new arcs since the upper bound is set to the total number  $K$  of cranes. Consequently, only cases (a) and (b) of Figure 5 can occur in an optimal solution of the MCFP since otherwise it is possible to create a new feasible flow with one less crossing and smaller cost after implementing the above operation. Therefore, the elimination of the crossings in an optimal solution of the MCFP results in an alternative solution with no crossings.  $\square$

**Proposition 5.** *An optimal crane schedule that satisfies the proper alignment constraint can be obtained by means of Algorithm 2.*

*Proof.* Any crane schedule with properly aligned cranes is a feasible solution of the MCFP. Arc crossing is not possible for a crane schedule that satisfies the proper alignment constraint. Therefore, the optimal value obtained by the branch-and-bound scheme or the minimum cost crane schedule without any arc crossing is not larger than the cost of any schedule which satisfies the proper alignment condition. Now, consider Algorithm 3 starting with an optimal solution of Algorithm 2.

---

### Algorithm 3

Assign cranes from 1 to  $K$  to vessel and waiting nodes in interval  $t_1$  from left to right on the berth (i.e., bottom-up in the layered network).

Assign cranes to vessels in other time periods by following the arcs in the network.

**if** the cranes in a node separates into multiple nodes in two consecutive layers **then**

    assign the cranes to vessels from left to right

**end if**

---

Assume there exists a misplaced crane pair, that is a crossing, at the end of Algorithm 3. This means that there exist two cranes with ordering  $k_1 < k_2$  with  $v_1$  (vessel to which crane  $k_1$  is assigned) to the right of  $v_2$  (vessel to which crane  $k_2$  is assigned) in interval  $t_l$ . These two cranes must come from two different nodes of the previous layer because otherwise they would be aligned by the algorithm. Since there is no arc crossing due to the branching strategy of the

algorithm, a misplacement should also exist in interval  $t_{l-1}$ . Repeating this argument we conclude that there must be a misplacement in interval  $t_1$  which is a contradiction because the algorithm provides proper alignment in interval  $t_1$ . In other words, Algorithm 2 together with Algorithm 3 gives a crane schedule that satisfies the proper alignment condition (i.e., a schedule with no crossings). We have also shown that the setup cost of this crane schedule is not larger than the cost of any crane schedule which satisfies the proper alignment condition.  $\square$

**Proposition 6.** Let  $|\overline{\mathcal{V}}|$  and  $|\overline{\mathcal{A}}|$  denote the size of  $\overline{\mathcal{V}}$  and  $\overline{\mathcal{A}}$  respectively, and let  $n_l$  denote the size of  $\mathcal{V}_l$ , the set of vessels berthed in interval  $l$ . Then,

$$|\overline{\mathcal{V}}| = \sum_{l=1}^L \binom{K - \sum_{i \in \mathcal{V}_l} g_{il} + n_l}{n_l} \quad (\text{A.1})$$

$$|\overline{\mathcal{A}}| = \sum_{l=0}^L \binom{K - \sum_{i \in \mathcal{V}_l} g_{il} + n_l}{n_l} \binom{K - \sum_{i \in \mathcal{V}_{l+1}} g_{il+1} + n_{l+1}}{n_{l+1}}, \quad (\text{A.2})$$

with the convention that  $\mathcal{V}_0 = \mathcal{V}_{L+1} = \emptyset$ .

*Proof.*  $K - \sum_{i \in \mathcal{V}_l} g_{il}$  is the number of available cranes after allocating  $g_{il}$  cranes to vessels  $i \in \mathcal{V}_l$  for fixed  $l$ . Then, the total number of non-crossing allocations of the cranes becomes equal to the number of selections with repetitions of  $n_l$  objects from  $K - \sum_{i \in \mathcal{V}_l} g_{il} + 1 + n_l$  types of objects in interval  $l$ . This is the number of nodes at layer  $l$  and its summation gives (A.1). Similarly, (A.2) is obtained by multiplying the number of nodes at layer  $l$  with the number of nodes at layer  $l + 1$  and using the convention.  $\square$