Carbon-efficient deployment of electric rubber-tyred gantry cranes in container terminals with workload uncertainty

Carbon-efficient Deployment of Electric Rubber-tyred Gantry Cranes in Container Terminals with Workload Uncertainty

Dayong Yu[a], Dong Li[b,*], Mei Sha[a], Dali Zhang[c]

[a] *College of Transport and Communications, Shanghai Maritime University, Shanghai, China*
[b] *School of Business and Economics, Loughborough University, Loughborough LE11 3TU, United Kingdom*
[c] *Sino-US Global Logistics Institute Shanghai Jiaotong University, Shanghai, China*

Abstract

Rubber-tyred gantry cranes are one of the major sources of carbon dioxide emissions in container terminals. In a move to green transportation, the traditional diesel powered cranes are being converted to electric ones. In this paper, we study the deployment of electric powered gantry cranes (ERTGs) in container terminal yards. Cranes always move in-between blocks to serve different workload. ERTGs use electricity for most movements but switch to diesel engines to allow inter-block transfers between unaligned blocks. We exploit this feature and propose to consider simultaneously the CO2 emissions and workload delays to develop carbon-efficient deployment strategies. Moreover, unlike previous works we consider the workload uncertainty, and model the problem as a two-stage stochastic program. A sample average approximation framework with Benders decomposition is employed to solve the problem. Multiple acceleration techniques are proposed, including a tailored regularised decomposition approach and valid inequalities. A case study with sample data from a major port in East China show that our proposal could reduce significantly CO2 emissions with only a marginal compromise in workload delays. Our numerical experiments also highlight the significance of the stochastic model and the efficiency of the Benders algorithms.

Keywords: OR in maritime industry, carbon-efficient, crane deployment, regularised decomposition, stochastic programming

1. Introduction

International trade and freight transportation depend heavily on marine transportation with container ships. Seaports and container terminals are becoming more and more important within the water transportation chain (World Shipping Council, a).

There has been a great deal of pressure to handle the constantly increasing number of containers. As it is usually not easy to expand the infrastructure of a container terminal, the efficient deployment and scheduling of its expensive equipment, such as gantry cranes, at the berth and terminal yards has become one of the most critical issues in terminal operations management.

*Corresponding author
Email addresses: dyyu@shmtu.edu.cn (Dayong Yu), d.li@lboro.ac.uk (Dong Li), meisha@shmtu.edu.cn (Mei Sha), zhangdl@sjtu.edu.cn (Dali Zhang)

A typical layout of the container terminals is shown in Figure 1. Containers flow in two opposite directions: inbound and outbound. For inbound flows, the containers carried on a vessel arrive at a pre-allocated berth according to its schedule. One or more quay cranes unload the containers onto internal trucks, which then drive to the designated blocks in the storage yard, where the yard cranes put them into storage, waiting to be picked up to the external trucks to inland destinations. A block is a rectangular shaped heap where containers are stacked on top of each other. A typical block may have 6 lanes of containers side by side, each of which might be 5 containers high and 20 containers long, depending on the shape of the storage yard. The outbound flows are carried out in the opposite direction. All the inbound and outbound containers will need to be stored in the storage yard until they are picked-up again either to internal or external trucks. The yard cranes carry out all the grounding and picking-up works; they serve both internal and external trucks. Their efficiency is critical to the smooth flow of the terminal. Yard cranes are usually the bottlenecks in the entire container handling process. Poor operations of these bulky equipment could lead to extra waiting time of trucks, cause delays to quay cranes, and ultimately push back vessels' departure time with a potential knock-on impact to the subsequent schedules. Therefore it is essential for yard cranes to work effectively.

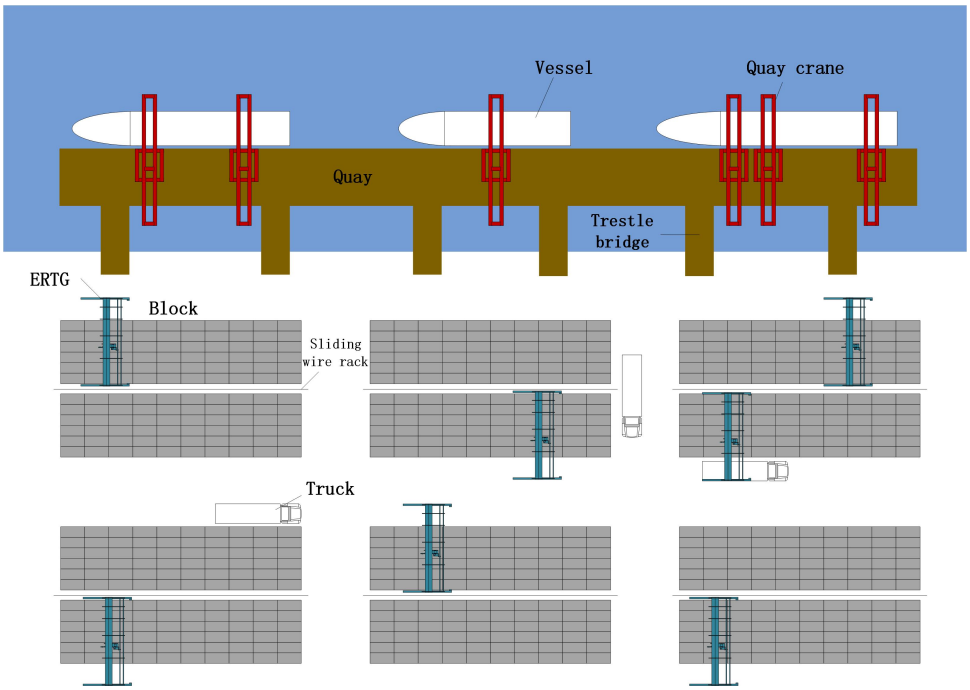

Figure 1: A typical layout of container terminals.

In most container terminal yards, the storage and retrieval of containers is handled by either rubber-tyred gantry cranes (RTGs) or rail-mounted gantry cranes (RMGs). Compared to the latter, RTGs are more flexible, and are widely used in seaports around the world. They can move freely among all blocks in the terminal, i.e., between any two blocks, whether aligned horizontally or not. However, RTGs are powered by diesel and consume more energy and emit much more carbon dioxide than RMGs, which are powered by electricity. Indeed, RTGs are a major source of CO2 emissions in container terminals. For instance in the Ningbo port of East China, the 4th largest port in the world with an annual container throughput of 21.60 millions

TEUs (World Shipping Council (b)), the RTGs in use together consume 30,000 tons diesel every year, causing significant air pollution only next to the exhaust fume from vessels.

Given the increasing demand for sustainable and environment friendly transport, container terminals have been under great pressure to find low carbon alternatives to RTGs. Reducing energy consumption and CO2 emissions of RTGs without a great reduction in their efficiency is a challenging task.

Several Chinese ports, such as Tianjin and Ningbo, whose annual throughputs have ranked in the top 10 in the world since 2012 (World Shipping Council, b), have undertaken some innovative energy-saving projects to tackle this problem. Taking Beilun terminal of the port of Ningbo as an example, a project to save energy and reduce emissions, called "diesel to electric," started in 2007 and finished in 2009, costing 88 million RMB in total. By the end of 2012, Ningbo port has electrified all of their RTGs, with an annual energy saving equivalent to 13,000 tons of standard coals. Since then, many other ports, both within and outside China, have followed the trend (VAHLE; Port Technology).

Diesel to electric means to upgrade the conventional RTGs by installing electric engines in them and providing electricity by either cable reels, the overhead wire, or the low frame sliding wire. The original diesel engines are still retained. The new equipment, called electric powered RTGs (ERTGs), are powered by electricity most of the time and switch to diesel only when they need to move between blocks in different rows or non-adjacent columns. Figure 1 shows the sliding wire racks located in-between two blocks. Both blocks are powered by the same rack, and there are no truck lanes between them. Figure 2 shows an ERTG using the low frame sliding wire. As long as it moves horizontally, either within one block or in-between two adjacent blocks, the electricity is always used. Only when it moves between different rows, the wire is unplugged before the transfer and plugged back afterwards, and in-between the diesel engine is used to turn the crane and travel between the two rows.

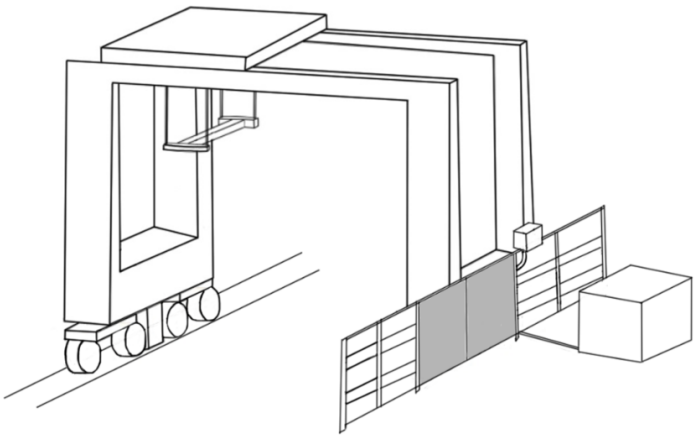

Figure 2: ERTGs with low frame sliding wire.

Therefore ERTGs enjoy the advantages of both RTGs and RMGs. Moreover, unlike RTGs, they do not consume any energy while idle. Although able to reduce CO2 emissions and save energy, ERTGs do need more time to transfer between blocks in different rows or non-adjacent columns, because it takes them additional time to switch between the electric and diesel modes.

Yard cranes are capital intensive assets; most terminals cannot afford to have enough of

them for each block. Moreover, the workload in each block varies and it is not cost effective to have a fixed number of cranes for each block. Therefore cranes constantly move in-between blocks. The main deployment questions are when to transfer them, and to where. As movement of these bulky cranes takes time and causes losses in productivity, traditional approaches for deployment aim to optimise the time related performances (e.g., workload delays), regardless of other objectives such as the environmental impact. Such approaches are appropriate for the conventional RTGs as they always consume energy and rarely switch off while in use. Thus the CO2 emissions are the same regardless of the deployment plan. This is, however, not the case any more for ERTGs. Different routes would lead to very different energy consumption, even though the resulting workload delay could be similar. It is this special feature that offers the opportunity for more energy efficient deployment plans to be made. Intuitively, if most of the inter-block transfers are between aligned blocks, the diesel engine will be rarely used, which however may be at the cost of extra workload delays as such transfers might not be the best moves to reduce the workload delays. The traditional deployment strategies for RTGs are therefore not enough for ERTGs. In this paper, we address the deployment of ERTGs with consideration of both workload delays and CO2 emissions. It is worth mentioning that even though ERTGs do not emit CO2 directly into terminals while in the electric mode, the generation of the electricity itself does and the indirect emissions need to be taken into account.

A successful crane deployment depends heavily on the accuracy of the estimated workload. Previous works usually estimate the workload based on the number of containers that are to be handled in each block (Zhang et al. (2002)), and treat the estimation as deterministic in making next-day plans for the cranes. In practice these plans are usually made a number of hours (typically 6-8 hours) before the actual work starts. The container numbers in each block may still change during this period. As mentioned in Lu and Le (2014), the information on the quantity of containers from vessels/land changes dynamically. Even if the exact numbers were known in advance, the total handling time required for each block would still be uncertain. Therefore, it is essential to consider the workload uncertainty in the crane deployment.

In summary, our contributions are twofold. To the crane deployment literature we exploit the unique features of ERTGs and develop carbon-efficient deployment strategies that consider simultaneously the CO2 emissions and the workload delays. We also fill the research gap of ignoring the workload uncertainty in crane deployment. We model the problem as a two-stage stochastic program, with the first stage determining the routes of crane transfers and the second stage the specific movement time. The problem is solved by the Benders decomposition algorithm embedded in a sample average approximation framework. Our numerical experiments (including a case study using sample data from Ningbo port in East China) show that, in comparison to the traditional and deterministic approaches in the literature, our proposal could lead to a significant reduction in CO2 emissions with none or a marginal compromise in workload delays. The derived deployment plans are also much more robust. Furthermore, we consider a more practical situation where the workload could arrive at a block at any moment during (rather than always at the beginning of) each time period, and the crane transfers do not have to be completed within the same time period. To the Benders decomposition literature, we exploit the special properties of our problem and propose multiple acceleration techniques.

In particular, a tailored regularised decomposition (RD) method is developed to stabilise the solutions during the process. As far as we know, the application of the RD method to problems whose first stage decision variables are integers is few (if any) in the literature. We identify a potential pitfall of this method and address it with a modified penalty adjustment approach. Moreover, two sets of valid inequalities specially designed for the problem concerned are identified. These techniques significantly enhance the performance of the standard Benders algorithm, as confirmed in our numerical experiments.

The remainder of this paper is organised as follows. We give an overview of the existing literature in Section 2. The proposed models are introduced in Section 3. In Section 4 a sample average approximation approach is proposed, along with a Benders decomposition algorithm. Multiple acceleration techniques are developed to speed up the solution process. Extensive numerical experiments, including a case study on a major port in East China, are reported in Section 5. Section 6 concludes this paper.

2. Literature review

The operations management of traditional RTG and RMG yard cranes in container terminals has been well studied. There are mainly two strands to the literature. The first concerns the deployment of cranes during a given planning horizon, usually a day. A common treatment is to divide the planning horizon into smaller intervals: then the cranes can be redeployed in each interval in such a way that a specified performance measurement is optimised. The second strand of the literature concerns the scheduling for each crane once they are allocated to a block. The detailed job arrival times and finish times are considered, with objectives such as to minimise the total waiting time of the trucks, or the total delays. A comprehensive survey of yard operations management in container terminals can be found in Carlo et al. (2014a).

Zhang et al. (2002) consider the deployment of RTGs in container terminals where the forecast workload arrives at each block at the beginning of every four-hour time period. The cranes can be redeployed at most once during each time period, the objective is to minimise the total unfinished workload at the end of the day. They require that all crane inter-block movements must be completed within the same time period. The problem is formulated as a mixed integer program and solved by a modified Lagrangian relaxation to cope with the large size of practical problems. Additional constraints are included to tighten up the results.

Cheung et al. (2002) propose an alternative model for the same inter-block crane deployment problems. In their model, the length of the time period can be very small, just a few minutes, so that the travel time of a crane between any two blocks is measured by the number of time periods, and so is the amount of the workload. A mixed integer programming problem (MIP) is formulated and solved by two approaches: one is based on Lagrangian decomposition and the other on a network flow model. A single time period problem is considered in Linn and Zhang (2003). They further divide the time period into smaller deployment time intervals. The cranes can be redeployed at the beginning of each deployment period. An interesting idea is that the number of deployment intervals itself is a decision variable. Their objective is to minimise the accumulated workload overflow from one deployment period to the next. The problem is formulated as an MIP and a least cost heuristic is developed to solve real size problems. He

et al. (2010) consider both the delayed workload and the number of crane movements between blocks. They allow each crane to be redeployed twice within each six-hour time period. A parallel genetic algorithm is developed for the problem and implemented in a rolling horizon manner. As do others in the literature, they assume the workload is always available at the beginning of each time period, and require all crane inter-block transfers to start and finish within the same period.

The research on crane scheduling is much more active. An early work due to Kim and Kim (2003) considers the dispatching of a single crane to minimise the total gantry time. An MIP is formulated and solved by heuristic approaches. Guo et al. (2011) also treat the dispatching of a single crane that serves a number of vehicles with deterministic arrival times, with the objective of minimising the overall vehicle waiting times. A modified A^* search algorithm with an admissible heuristic is proposed to obtain the optimal dispatching solutions. A two-crane dispatching problem is addressed in Jung and Kim (2006) to minimise the total time taken to finish all jobs. Ng (2005) considers the scheduling of multiple cranes to a set of jobs with known arrival times and locations, and aims to minimise the total completion time of all jobs. The interference between cranes is addressed in their model. A two-phase heuristic is developed to generate a tight lower bound for the solutions. However they do not consider the distance requirement between two cranes. Such an issue is addressed in a recent work due to Guo and Huang (2012). More works on crane scheduling can be found in Li et al. (2009), Ng et al. (2006), and Lee et al. (2007).

The majority of the previous literature on yard crane operations management, whether on deployment or scheduling, only treats performance measurements related to time, or in other words, efficiency. The consideration of energy consumption or CO2 emissions is rare. One such is He et al. (2015), who include the energy consumption in a crane scheduling problem. A single time period is considered and all tasks should be able to be finished within that period. Each task has a known arrival time and due time. The tasks are aggregated into different task groups. Their purpose is to schedule multiple cranes to serve all these task groups so that the overall delay is minimised. The problem is reformulated as a vehicle routing problem with time windows. A major limitation of that work is the neglect of blocks, a critical feature of yard management. Their modelling approach cannot differentiate between moves within blocks and moves between blocks, neither could it address the distance requirement between cranes within a block. More recently, Sha et al. (2016) also consider energy consumption as part of crane scheduling. However, they treat a special case where the total number of cranes is always enough to cover the total workload within each time period, and thus they do not need to be concerned with workload delays, just the inter-block crane movements. Both those works consider RTGs only.

We are not aware of anything in the literature which has explicitly considered CO2 emissions in the deployment of RTGs in container terminals, not to mention ERTGs. In fact, as far as we know, the present paper is the first to consider the operations management of ERTGs. The current literature on ERTGs mainly concerns strategical analyses of converting RTGs to ERTGs (Peng et al. (2016) and references therein). All the above mentioned works only consider deterministic workload. In fact, the uncertainty in workload estimation has been rarely

considered in the crane deployment literature. Some researchers have analysed uncertainties in multiple processes of handling containers, however they do not concern the deployment of yard cranes. Lu and Le (2014) consider integrated optimisation of container terminal scheduling with the uncertain travelling speed (of both internal trucks and cranes) and cranes' processing times. They develop a particle swarm optimisation algorithm to minimise the total expected time of the integrated operation involving a single yard crane, a single quay crane and a few trucks. More recently Roy and de Koster (2018) study the performance of a terminal of overlapping loading and unloading operations, taking into account the stochastic interactions among quay-side, automated lifting vehicles, and yard processes. They analyse the uncertainties in multiple stages during loading/unloading containers from the vessel.

Apart from the operations management of yard cranes, many researches on container terminals are concerned with the problems such as berth allocation, quay crane scheduling, assignment of container locations (the grounding policy), and integrated operations. For comprehensive accounts on these problems the readers are directed to a series of review papers (Carlo et al. (2014a,b, 2015)) and references therein.

Before proceeding, we pause to highlight that crane deployment and scheduling are always solved separately in the literature; the integrated optimisation of them is very complicated. The crane deployment concerns the distribution of multiple cranes across a grid of blocks through the entire time horizon. This is the higher level decision in the hierarchy. The lower level decision is the scheduling of individual cranes and concerns the detailed handling sequence of individual containers for each crane. It would be ideal to solve these two problems simultaneously to achieve the real optimum. However, the resulting integrated problem needs to take into account the higher level decisions over all blocks and the entire time horizon, and in the same time consider the very detailed constraints for the movement of each individual crane (such as the interference between two crossing cranes, and the precedence relationships between multiple containers). Even though it could be possible to develop some "comprehensive" models to address them together, the models may well be too complicated to solve exactly within reasonable time, especially when the workload uncertainties are considered. Some approximation or simplification will be inevitable, which would compromise the quality of the solutions. The truly integrated optimisation of both level of decisions with multiple cranes across multiple blocks and time periods is rare (if any at all) in the literature. As far as we know the closest attempt so far is due to Guo and Huang (2012). Yet they assume that the cranes are pre-allocated into different rows and only optimise the deployment and scheduling of those cranes within each single row. No inter-row transfers are allowed, which is a major limitation but allows them to solve the problem (which however is not the true integration).

3. The models

3.1. Problem description

The crane deployment concerns the distribution and movement of cranes among multiple blocks over time. We consider the deployment of ERTGs over a finite planning horizon (e.g. the following day) that is divided into T equal-length time periods each with a length of C. We index these time periods by $t = 1, 2, \cdots, T$. The capacity of a crane is defined by the

available working time during each time period, which is just C. All cranes can be deployed to different blocks in any period. There are overall $K \times L$ blocks arranged in a grid of K rows and L columns, each of which starts with Q_i cranes at the beginning of the horizon. We index the blocks by $i = 1, \cdots, I$, in such a way that they are sequenced firstly in columns and then in rows, with $I = KL$. Due to safety rules, at most two cranes can be deployed in the same block at any point of time. During time period t one batch of workload will arrive at block i, with the first job arriving at time A_i^t. We allow A_i^t to be any point of time within the period, as shown in Figure 3. The total amount of workload of the batch is measured by the required crane time. The workload could be either to store or retrieve containers. The efficiency of each crane is assumed to be the same for either type of workload.

Denote by $\mathcal{T}_i^t = [A_i^t, A_i^{t+1}]$ the time interval in between the first job's arrival times of the workload batches to block i in time period t and $t+1$. For completeness, we define $A_i^{T+1} = TC$, and thus the last period $\mathcal{T}_i^T$ is well defined, where TC is the end of the planning horizon.

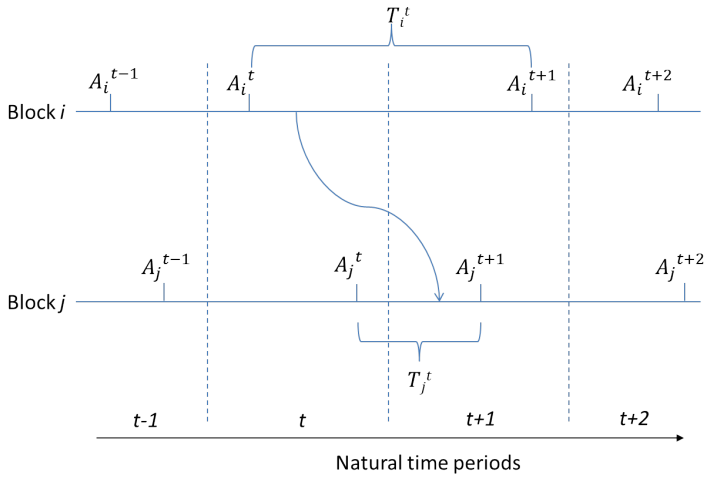

Figure 3: Illustration of the time periods.

If the workload batch that has arrived in the current period is not completed before the arrival of the next batch, the unfinished workload is delayed and carried over to the next time period. Whenever necessary, cranes can be transferred from one block to another. However, within any time window $\mathcal{T}_i^t$, at most one crane can be moved in or out of a block. Such a rule is usually adopted in practice as too often transfers will lead to blocked traffic and lost working hours. The start and end times of a transfer could fall into either the same or two successive natural time periods, as shown in Figure 3. The travel time from block i to j is denoted by t_{ij}. There are three components involved in a transfer: the time to switch between electric mode and diesel mode (T_s); the time to make a 90-degree turn (T_r); and the actual travelling time L_{ij}/v, where L_{ij} is the moving distance from block i to block j and v is the velocity of a crane. We define two functions, $k(i)$ and $l(i)$, to return the row and column of a block i, respectively.

Therefore, we have

$$t_{ij} = \frac{L_{ij}}{v} + t_{ij}^s + t_{ij}^r,$$

where the total switching time

$$t_{ij}^s = \begin{cases} 0, & k(i) = k(j), |l(i) - l(j)| = 1 \text{ (same row and adjacent columns)} \\ 2T_s, & \text{otherwise (different rows or non-adjacent columns)} \end{cases},$$

and the total 90-degree turning time

$$t_{ij}^r = \begin{cases} 0, & k(i) = k(j), |l(i) - l(j)| = 1 \text{ (same row and adjacent columns)} \\ 2T_r, & k(i) \neq k(j), |l(i) - l(j)| = 1 \text{ (different rows and adjacent columns)} \\ 4T_r, & \text{otherwise (non-adjacent columns)} \end{cases}.$$

For ERTGs, the diesel engine is only in use for transfers between different rows or non-adjacent columns (otherwise the electric engine is always used). Switching is needed at both the beginning and end of each transfer. For transfers between adjacent columns (like ERTG 1 in Figure 4), two 90 degree turns are needed, while for non-adjacent ones (ERTG 2), another two turns are required to move to and from the border of the yard to facilitate the transfer.

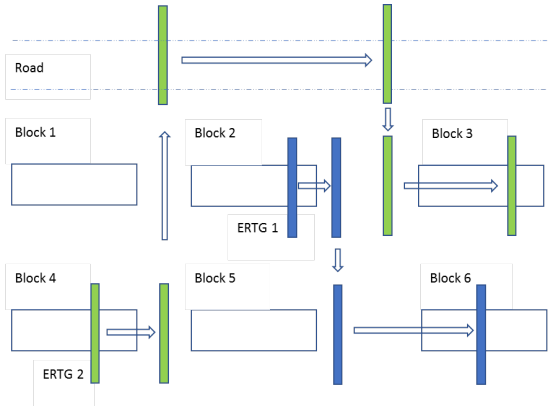

Figure 4: Crane transfer between blocks.

We denote the CO2 emission rate (per unit time) by q_1 kg/min for electric and q_2 kg/min for diesel. Their values are calculated based on the average travelling speed of cranes, their energy consumption rates, and the published CO2 emission indices. More details are included in Section 5.1. Note that even though ERTGs do not emit CO2 directly to terminals in the electric mode, the generation of the electricity does. For simplicity, we define a unified notation q_{ij} for the CO2 emission rate due to the movement from block i to j, as follows.

$$q_{ij} = \begin{cases} 0, & i = j \\ q_1, & k(i) = k(j), |l(i) - l(j)| = 1 \text{ (same row and adjacent columns)} \\ q_2, & \text{otherwise (different rows or non-adjacent columns)} \end{cases}.$$

Since switching between the electric and diesel modes does not consume fuel, we define by $s_{ij} = L_{ij}/v + t_{ij}^r$ the actual time of energy consumption during a transfer. Once the overall time spent on the inter-block transfers are known, the total amount of CO2 emissions can be readily calculated. Note that the energy consumption and CO2 emissions due to the actual handling

of containers are proportional to the total workload and independent of the crane deployment decisions. They are not included in our model.

Each unit of delayed workload incurs a penalty cost of c_d, and each kg of CO2 emissions during crane transfers incurs a cost of c_e. Our objective is to develop a deployment strategy over the planning horizon that minimises the overall cost.

3.2. The deterministic formulation

In this section we consider a perfect situation where the exact amount of workload in each block and in each time period is known at the time of planning, which is denoted by B_i^t for block i in time period $\mathcal{T}_i^t$. We define the following decision variables.

- d_i^t, the amount of workload in block i that is not finished in time in period $\mathcal{T}_i^t$, for $t = 0, 1, \cdots, T$.
- x_{ij}^t, the number of ERTGs that leave block i during time period $\mathcal{T}_i^t$ and arrive at j during $\mathcal{T}_j^t$, for $t = 0, 1, \cdots, T$. These variables determine the movement of cranes among blocks in each time period throughout the planning horizon. Note that a variable with $i = j$ actually means the number of cranes staying in the same block.
- f_{ij}^t, the amount of workload fulfilled by ERTGs in block i before they leave for block j during period $\mathcal{T}_i^t$, for $t = 1, \cdots, T$. These variables, along with the following ones, determine when an inter-block transfer should start or finish.
- g_{ij}^t, the amount of workload fulfilled by ERTGs in block j after they arrive from block i during period $\mathcal{T}_j^t$, for $t = 1, \cdots, T$.

Note that we have defined the first two variables for $t = 0$, in order to simplify the presentation of the following model.

We now give the mixed integer program for the deterministic problem, as written below.

$$\text{(DP)} \qquad \min \quad c_e \sum_{t=1}^{T} \left(\sum_{i=1}^{I} \sum_{j=1}^{I} q_{ij} s_{ij} x_{ij}^t \right) + c_d \sum_{t=1}^{T} \sum_{i=1}^{I} d_i^t \tag{1a}$$

$$\text{subject to} \sum_{j=1}^{I} x_{ij}^t = \sum_{j=1}^{I} x_{ji}^{t-1}, \forall 1 \le i \le I, 1 \le t \le T, \tag{1b}$$

$$\sum_{j=1, j\neq i}^{I} x_{ij}^t + \sum_{j=1}^{I} x_{ji}^t \le 2, \forall 1 \le i \le I, 1 \le t \le T, \tag{1c}$$

$$d_i^t - d_i^{t-1} + \sum_{j=1}^{I} f_{ij}^t + \sum_{j=1}^{I} g_{ji}^t = B_i^t, \forall 1 \le i \le I, 1 \le t \le T, \tag{1d}$$

$$f_{ij}^t + g_{ij}^t \le (A_j^{t+1} - A_i^t - t_{ij}) x_{ij}^t, \forall 1 \le i, j \le I, 1 \le t \le T, \tag{1e}$$

$$f_{ij}^t \le A_i^{t+1} - A_i^t, \forall 1 \le i, j \le I, i \neq j, 1 \le t \le T, \tag{1f}$$

$$g_{ij}^t \le A_j^{t+1} - A_j^t, \forall 1 \le i, j \le I, i \neq j, 1 \le t \le T, \tag{1g}$$

$$d_i^0 = 0, \forall 1 \le i \le I, \tag{1h}$$

$$x_{ii}^0 = Q_i, x_{ij}^0 = 0, \forall 1 \le i, j \le I, i \neq j, \tag{1i}$$

$$x_{ii}^t = \{0, 1, 2\}, x_{ij}^t = \{0, 1\}, \forall 1 \le i, j \le I, i \neq j, 1 \le t \le T, \tag{1j}$$

$$f_{ij}^t \ge 0, g_{ij}^t \ge 0, d_i^t \ge 0, \forall 1 \le i, j \le I, 1 \le t \le T. \tag{1k}$$

The objective function (1a) minimises the overall cost incurred for both CO2 emissions during the inter-block transfers (the first term) and delayed workload (the second term).

Constraints (1b) ensure that the number of ERTGs which remain in block i or move out during period $\mathcal{T}_i^t$ is equal to the number of those remaining or being moved into this block during the previous period $\mathcal{T}_i^{t-1}$. See Figure 3 for an illustration. Constraints (1c) ensure that the total number of ERTGs being transferred either in or out or remaining in a block, is not more than two during any time period. Constraints (1d) make sure that the workload flow remains balanced over time. By Constraints (1e), the total completed workload of an ERTGs transferred from block i to j does not exceed the total available working time in both blocks. Constraints (1f) ensure that the amount of workload fulfilled by an ERTG in block i before it transfers to block j may not exceed the time that it can work in block i. Similarly, constraints (1g) ensure that the workload fulfilled by an ERTG after transferring to block j may not exceed the time that it can work in block j. Constraints (1h)–(1i) give the initial situations of the system. Constraints (1j) and (1k) specify variable ranges.

3.3. The stochastic model

In practice the exact amount of workload is never known at the time of planning, and the deterministic model above has limited usage. In this section we propose a stochastic model that takes into account the workload uncertainty. Denote the workload arriving at block i in period t by a random variable $\mathbf{B}_i^t \sim F_i^t$, where F_i^t is the corresponding distribution function that is known to the port operator or can be estimated at the time of planning. Assume all $\mathbf{B}_i^t$ are independent. As we have mentioned, the crane deployment decisions need to be made

well in advance of the planning horizon. After that new and more accurate information will continue to become available right until the actual work starts. For example, the exact number of containers in each block will be known. Detailed scheduling of quay cranes and the number of internal trucks could also have been determined, which will contribute to a much more accurate estimation of the total handling times required (and thus the workload).

In light of these observations, we propose to model the problem as a two-stage stochastic program. The first stage, which takes place at the time of planning, determines the number of cranes that remain in the same block or transfer to another block in each time period over the planning horizon, with the knowledge of the distributions of arriving workload. The second stage, which takes place just at the beginning of the planning horizon when much better understanding of the workload is available, determines the movement time of cranes if they are due to be transferred to another block as prescribed in the first stage decision. It is worth mentioning that in real world problems the exact amount of workload cannot be fully known until all works have been completed. The workload that we use in the second stage can be seen as the most precise estimation that can be obtained just before making the second stage decisions.

Our objective is to find the optimal deployment that minimises the total CO2 emission cost and the expected workload delay cost over the planning horizon. Note that the first stage concerns CO2 emissions while the second stage the workload delays. Our two-stage stochastic programming model is written as follows. For convenience, we have used $\mathbf{B}$ to denote the vector of the random workload for all blocks in each time period, and used B to denote the vector of a realisation (or scenario) of $\mathbf{B}$.

$$\text{(SP)} \quad \min \qquad c_e \sum_{t=1}^{T} \left(\sum_{i=1}^{I} \sum_{j=1}^{I} q_{ij} s_{ij} x_{ij}^t \right) + c_d \mathbb{E}[Q(\mathbf{x}, \mathbf{B})] \tag{2a}$$

$$\text{subject to} \qquad \sum_{j=1}^{I} x_{ij}^t = \sum_{j=1}^{I} x_{ji}^{t-1}, \forall 1 \leq i \leq I, 1 \leq t \leq T, \tag{2b}$$

$$\sum_{j=1, j \neq i}^{I} x_{ij}^t + \sum_{j=1}^{I} x_{ji}^t \leq 2, \forall 1 \leq i \leq I, 1 \leq t \leq T, \tag{2c}$$

$$x_{ii}^0 = Q_i, x_{ij}^0 = 0, \forall 1 \leq i, j \leq I, i \neq j, \tag{2d}$$

$$x_{ii}^t = \{0, 1, 2\}, x_{ij}^t = \{0, 1\}, \forall 1 \leq i, j \leq I, i \neq j, 1 \leq t \leq T, \tag{2e}$$

where $Q(\mathbf{x}, B)$ is the optimal objective function value of the following problem,

$$\min \quad \sum_{t=1}^{T}\sum_{i=1}^{I} d_i^t \tag{3a}$$

$$\text{subject to} \quad d_i^t - d_i^{t-1} + \sum_{j=1}^{I} f_{ij}^t + \sum_{j=1}^{I} g_{ji}^t = B_i^t, \forall 1 \le i \le I, 1 \le t \le T, \tag{3b}$$

$$f_{ij}^t + g_{ij}^t \le (A_j^{t+1} - A_i^t - t_{ij})x_{ij}^t, \forall 1 \le i,j \le I, 1 \le t \le T, \tag{3c}$$

$$f_{ij}^t \le A_i^{t+1} - A_i^t, \forall 1 \le i,j \le I, i \ne j, 1 \le t \le T, \tag{3d}$$

$$g_{ij}^t \le A_j^{t+1} - A_j^t, \forall 1 \le i,j \le I, i \ne j, 1 \le t \le T, \tag{3e}$$

$$d_i^0 = 0, \forall 1 \le i \le I, \tag{3f}$$

$$f_{ij}^t \ge 0, g_{ij}^t \ge 0, d_i^t \ge 0, \forall 1 \le i,j \le I, 1 \le t \le T. \tag{3g}$$

Note that the second stage optimal value $Q(\mathbf{x}, B)$ is the minimum workload delay in the workload scenario B with the given first stage decision $\mathbf{x}$. The decision variables f_{ij}^t and g_{ij}^t determine the optimal departure or arrival time of each inter-block transfer (as prescribed by $\mathbf{x}$) in that scenario.

Remark 1. *The first stage produces the main deployment plan, which determines the number of cranes that stay in the same block or transfer to another one (and which one) in each time period over the planning horizon. Remember these results are optimal with consideration of the uncertain workload scenarios. The first stage decisions can be readily implemented. Note that in practice the deployment plan is made in advance and adjustments are made only if necessary. The second stage determines the departure or arrival time for the cranes which are due to be transferred. These decisions are dependent upon the actual amount of workload, which in turn depends on the scheduling of cranes. The second stage decisions can be viewed as the guidelines. In practice any reasonable scheduling under the given deployment plan can be applied and the actual movement time can then be determined. Nevertheless, the first stage decisions are the main results needed for the crane deployment. As we shall see in the numerical study section, the solution proposed by our approach has strong and robust performance.*

4. Solution algorithms

4.1. Sample average approximation

In many real world problems, including ours, the distribution functions F_i^t are continuous. It is non-trivial to integrate $\mathbb{E}[Q(\mathbf{x}, \mathbf{B})]$ for a given $\mathbf{x}$ even with the simplest distribution functions. Moreover, even if an explicit evaluation of $\mathbb{E}[Q(\mathbf{x}, \mathbf{B})]$ is possible, it is still a non-linear function of $\mathbf{x}$ and its closed form is not available. Minimisation of this function is significantly difficult. When the distribution functions are discrete, the exact evaluation of $\mathbb{E}[Q(\mathbf{x}, \mathbf{B})]$ for a given $\mathbf{x}$ may still be practically impossible when the number of possible realisations is extremely large. To deal with these computational challenges we use a framework called *sample average approximation* (SAA). In the SAA framework, a sample of N realisations of $\mathbf{B}$ is generated and denoted by $B^n, n = 1, \cdots, N$. Using the sample averaged function $\frac{1}{N}\sum_{n=1}^{N} Q(\mathbf{x}, B^n)$ to replace

$\mathbb{E}[Q(\mathbf{x}, \mathbf{B})]$ in equation (2a), and including into problem (SP) a set of constraints (3b-3g), one for each realisation B^n, we obtain a large scale mixed integer program, as written below.

$$\text{(SAA)} \quad \min \quad c_e \sum_{t=1}^{T} \left(\sum_{i=1}^{I} \sum_{j=1}^{I} q_{ij} s_{ij} x_{ij}^t \right) + c_d \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{i=1}^{I} d_i^{t,n} \tag{4a}$$

$$\text{subject to} \quad \sum_{j=1}^{I} x_{ij}^t = \sum_{j=1}^{I} x_{ji}^{t-1}, \forall 1 \leq i \leq I, 1 \leq t \leq T, \tag{4b}$$

$$\sum_{j=1, j \neq i}^{I} x_{ij}^t + \sum_{j=1}^{I} x_{ji}^t \leq 2, \forall 1 \leq i \leq I, 1 \leq t \leq T, \tag{4c}$$

$$d_i^{t,n} - d_i^{t-1,n} + \sum_{j=1}^{I} f_{ij}^{t,n} + \sum_{j=1}^{I} g_{ji}^{t,n} = B_i^{t,n}, \forall 1 \leq i \leq I, 1 \leq t \leq T, 1 \leq n \leq N, \tag{4d}$$

$$f_{ij}^{t,n} + g_{ij}^{t,n} \leq (A_j^{t+1} - A_i^t - t_{ij}) x_{ij}^t, \forall 1 \leq i, j \leq I, 1 \leq t \leq T, 1 \leq n \leq N, \tag{4e}$$

$$f_{ij}^{t,n} \leq A_i^{t+1} - A_i^t, \forall 1 \leq i, j \leq I, i \neq j, 1 \leq t \leq T, 1 \leq n \leq N, \tag{4f}$$

$$g_{ij}^{t,n} \leq A_j^{t+1} - A_j^t, \forall 1 \leq i, j \leq I, i \neq j, 1 \leq t \leq T, 1 \leq n \leq N, \tag{4g}$$

$$d_i^{0,n} = 0, \forall 1 \leq i \leq I, 1 \leq n \leq N, \tag{4h}$$

$$f_{ij}^{t,n} \geq 0, g_{ij}^{t,n} \geq 0, d_i^{t,n} \geq 0, \forall 1 \leq i, j \leq I, 1 \leq t \leq T, 1 \leq n \leq N, \tag{4i}$$

$$x_{ii}^0 = Q_i, x_{ij}^0 = 0, \forall 1 \leq i, j \leq I, i \neq j, \tag{4j}$$

$$x_{ii}^t = \{0, 1, 2\}, x_{ij}^t = \{0, 1\}, \forall 1 \leq i, j \leq I, i \neq j, 1 \leq t \leq T. \tag{4k}$$

We refer the solution $\mathbf{x}^*$ to (SP) as the *true solution* and the solution $\mathbf{x}^N$ to (SAA) as the *SAA solution* with respect to the sample $B^n, n = 1, \cdots, N$. SAA methods have been extensively investigated in the stochastic optimisation community. This type of methods is also known as sample path optimisation (SPO) methods. There is extensive literature on SAA and SPO (see for example Xu and Zhang (2009) and a comprehensive review by Shapiro (2003)).

We now discuss the convergence of the SAA solution $\mathbf{x}^N$. Denote by $\mathcal{S}$ the feasible set defined by (2b-2e) and consider optimisation problem (SP) in the form of

$$\min_{\mathbf{x} \in \mathcal{S}} \quad z(\mathbf{x}) := c_e \sum_{t=1}^{T} \left(\sum_{i=1}^{I} \sum_{j=1}^{I} q_{ij} s_{ij} x_{ij}^t \right) + c_d \mathbb{E}[Q(\mathbf{x}, \mathbf{B})].$$

Similarly, problem (SAA) can be re-written as

$$\min_{\mathbf{x} \in \mathcal{S}} \quad z^N(\mathbf{x}) := c_e \sum_{t=1}^{T} \left(\sum_{i=1}^{I} \sum_{j=1}^{I} q_{ij} s_{ij} x_{ij}^t \right) + c_d \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{i=1}^{I} d_i^{t,n}.$$

Here we write by z^N the optimal value of the above SAA problem and refer it as the *SAA optimal value* with respect to the sample. We on the other hand write by z^* the optimal value of (SP) and refer it as the *true optimal value*.

It is possible to show that under mild regularity conditions, as the sample size N increases,

z^N and $\mathbf{x}^N$ converge with probability one to their true counterparts. Moreover $\mathbf{x}^N$ converges to an optimal solution of the true problem with probability approaching one exponentially fast (Kelywegt et al. (2002)).

Proposition 1. *Suppose that the variance* $\sigma^2(\mathbf{x}) := \mathrm{Var}\left[Q(\mathbf{x}, \mathbf{B})\right]$ *exists for every* $\mathbf{x} \in \mathcal{S}$. *Then*

$$\sqrt{N}(z^N - z^*) \Rightarrow \min_{\mathbf{x} \in \mathcal{S}^*} N(0, \Sigma(\mathbf{x}))$$

where $\Sigma(\mathbf{x})$ *is the covariance function of* $Q(\mathbf{x}, \mathbf{B})$. *In particular, if* $\mathcal{S}^* = \{\mathbf{x}^*\}$ *is a singleton, then*

$$\sqrt{N}(z^N - z^*) \Rightarrow N(0, \Sigma(\mathbf{x}^*)).$$

We can easily prove the above results by verifying the conditions in Proposition 2.4 in Kelywegt et al. (2002), where the existence of $\mathrm{Var}\left[Q(\mathbf{x}, \mathbf{B})\right]$ and $\Sigma(\mathbf{x})$ directly implies the existence of variances and covariances of $c_e \sum_{t=1}^{T} \left(\sum_{i=1}^{I} \sum_{j=1}^{I} q_{ij} s_{ij} x_{ij}^t \right) + c_d Q(\mathbf{x}, \mathbf{B})$ when c_d is finite. Therefore, we obtain the convergence result in Proposition 1. Note that this convergence result is well-investigated for integer programming problems in Kelywegt et al. (2002) and has been implemented in Santoso et al. (2005). We omit the detailed proof for the proposition.

In practice, the SAA framework involves repeated solutions of the (SAA) problem with independent samples. Statistical confidence intervals are then derived on the quality of the approximate solutions. For more details refer to Kelywegt et al. (2002).

4.2. Benders decomposition

The number of decision variables and constraints in problem (SAA) increases proportionally with the sample size N. For large N it becomes non-realistic to solve the problem within reasonable time using standard packages. In such situations Benders decomposition (Benders (1962)) is usually employed to address the computationally challenges. For the given $\mathbf{x}$, problem (SAA) reduces to a linear program that contains only the continuous variables. Moreover, this linear program is of a L-structured shape and can be readily decomposed into N smaller linear programs, one for each sample B^n. These linear programs are independent to each other and can be solved in parallel, and thus the computational time is constant with regard to the sample size. This is one of the most attractive features of the Benders decomposition approaches.

For a given set of $\bar{x}_{ij}^t$ which satisfy constraints (4b–4c) and (4j–4k), problem (SAA) reduces

to the following *Benders primal subproblem.*

$$\text{(BPS)} \quad \min \qquad c_d \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{i=1}^{I} d_i^{t,n} \tag{5a}$$

$$\text{subject to} \qquad d_i^{t,n} - d_i^{t-1,n} + \sum_{j=1}^{I} f_{ij}^{t,n} + \sum_{j=1}^{I} g_{ji}^{t,n} = B_i^{t,n}, \forall 1 \le i \le I, 1 \le t \le T, 1 \le n \le N, \tag{5b}$$

$$f_{ij}^{t,n} + g_{ij}^{t,n} \le (A_j^{t+1} - A_i^t - t_{ij}) \bar{x}_{ij}^t, \forall 1 \le i, j \le I, 1 \le t \le T, 1 \le n \le N, \tag{5c}$$

$$f_{ij}^{t,n} \le A_i^{t+1} - A_i^t, \forall 1 \le i, j \le I, i \ne j, 1 \le t \le T, 1 \le n \le N, \tag{5d}$$

$$g_{ij}^{t,n} \le A_j^{t+1} - A_j^t, \forall 1 \le i, j \le I, i \ne j, 1 \le t \le T, 1 \le n \le N, \tag{5e}$$

$$d_i^{0,n} = 0, \forall 1 \le i \le I, 1 \le n \le N, \tag{5f}$$

$$f_{ij}^{t,n} \ge 0, g_{ij}^{t,n} \ge 0, d_i^{t,n} \ge 0, \forall 1 \le i, j \le I, 1 \le t \le T, 1 \le n \le N. \tag{5g}$$

The first term in (4a) of problem (SAA) is constant and thus not included. Note that problem (BPS) is always feasible as one can simply let all $f_{ij}^{t,n}$ and $g_{ij}^{t,n}$ be zero and get a feasible solution.

It is clear that BPS can be decomposed into N smaller problems. Denote by BPS^n the subproblem for the n^{th} sample, with an objective to minimise the workload delay cost in that scenario.

$$\text{(BPS}^n\text{)} \quad \min \qquad c_d \sum_{t=1}^{T} \sum_{i=1}^{I} d_i^{t,n} \tag{6a}$$

$$\text{subject to} \qquad d_i^{t,n} - d_i^{t-1,n} + \sum_{j=1}^{I} f_{ij}^{t,n} + \sum_{j=1}^{I} g_{ji}^{t,n} = B_i^{t,n}, \forall 1 \le i \le I, 1 \le t \le T, \tag{6b}$$

$$f_{ij}^{t,n} + g_{ij}^{t,n} \le (A_j^{t+1} - A_i^t - t_{ij}) \bar{x}_{ij}^t, \forall 1 \le i, j \le I, 1 \le t \le T, \tag{6c}$$

$$f_{ij}^{t,n} \le A_i^{t+1} - A_i^t, \forall 1 \le i, j \le I, i \ne j, 1 \le t \le T, \tag{6d}$$

$$g_{ij}^{t,n} \le A_j^{t+1} - A_j^t, \forall 1 \le i, j \le I, i \ne j, 1 \le t \le T, \tag{6e}$$

$$d_i^{0,n} = 0, \forall 1 \le i \le I, \tag{6f}$$

$$f_{ij}^{t,n} \ge 0, g_{ij}^{t,n} \ge 0, d_i^{t,n} \ge 0, \forall 1 \le i, j \le I, 1 \le t \le T. \tag{6g}$$

Define $\pi_i^{t,n}, \varphi_{ij}^{t,n}, \mu_{ij}^{t,n}, \nu_{ij}^{t,n}, \psi_i^n$ to be the dual decision variables for each of the constraints (6b)–(6f), respectively. The *Benders dual subproblem* for the n^{th} sample takes the following

form.

$$\text{(BDS}^n\text{)} \quad \max \quad \sum_{t=1}^{T}\sum_{i=1}^{I} B_i^{t,n}\pi_i^{t,n} + \sum_{t=1}^{T}\sum_{i,j=1}^{I}(A_j^{t+1} - A_i^t - t_{ij})\bar{x}_{ij}^t\varphi_{ij}^{t,n} + \sum_{t=1}^{T}\sum_{i,j=1,i\neq j}^{I}(A_i^{t+1} - A_i^t)\mu_{ij}^{t,n} + \sum_{t=1}^{T}\sum_{i,j=1,i\neq j}^{I}(A_j^{t+1} - A_j^t)\nu_{ij}^{t,n} \tag{7a}$$

$$\text{subject to} \quad \pi_i^{t,n} - \pi_i^{t+1,n} \leq c_d, \forall 1 \leq i \leq I, 1 \leq t \leq T-1, \tag{7b}$$

$$\pi_i^{T,n} \leq c_d, \forall 1 \leq i \leq I, \tag{7c}$$

$$\varphi_{ij}^{t,n} + \mu_{ij}^{t,n} + \pi_i^{t,n} \leq 0, \forall 1 \leq i,j \leq I, 1 \leq t \leq T, \tag{7d}$$

$$\varphi_{ij}^{t,n} + \nu_{ij}^{t,n} + \pi_j^{t,n} \leq 0, \forall 1 \leq i,j \leq I, 1 \leq t \leq T, \tag{7e}$$

$$\psi_i^n - \pi_i^{1,n} \leq 0, \forall 1 \leq i \leq I, \tag{7f}$$

$$\varphi_{ij}^{t,n} \leq 0, \mu_{ij}^{t,n} \leq 0, \nu_{ij}^{t,n} \leq 0, \mu_{ii}^{t,n} = 0, \nu_{ii}^{t,n} = 0. \tag{7g}$$

Note that the feasible region of the above problem is defined by (7b–7g), which is independent of the variables x_{ij}^t. Let Λ be the polyhedron defined by constraints (7b–7g) and P_Λ the set of extreme points of Λ. Both Λ and P_Λ are independent of the sample and thus applied to all BDS problems.

Since the primal problems (BPS^n) are always feasible, their dual problems (BDS^n) are always feasible and bounded. Denoted by $(\bar{\boldsymbol{\pi}}^n, \bar{\boldsymbol{\varphi}}^n, \bar{\boldsymbol{\mu}}^n, \bar{\boldsymbol{\nu}}^n, \bar{\boldsymbol{\psi}}^n) \in P_\Lambda$ an optimal solution to problem (BDS^n). Following the standard results, the solutions across all N scenarios together define a Benders optimality cut in the following form.

$$\eta \geq \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{t=1}^{T}\sum_{i=1}^{I} B_i^{t,n}\bar{\pi}_i^{t,n} + \sum_{t=1}^{T}\sum_{i,j=1}^{I}(A_j^{t+1} - A_i^t - t_{ij})x_{ij}^t\bar{\varphi}_{ij}^{t,n} + \sum_{t=1}^{T}\sum_{i,j=1,i\neq j}^{I}(A_i^{t+1} - A_i^t)\bar{\mu}_{ij}^{t,n} + \sum_{t=1}^{T}\sum_{i,j=1,i\neq j}^{I}(A_j^{t+1} - A_j^t)\bar{\nu}_{ij}^{t,n}\right).$$

Problem (SAA) can then be reformulated as follows, which is called the *Benders master*

problem.

$$\text{(BMP)} \qquad \min: \quad \eta + c_e \sum_{t=1}^{T} \left(\sum_{i=1}^{I} \sum_{j=1}^{I} q_{ij} s_{ij} x_{ij}^t \right) \tag{8a}$$

$$\text{subject to} \sum_{j=1}^{I} x_{ij}^t = \sum_{j=1}^{I} x_{ji}^{t-1}, \forall 1 \le i \le I, 1 \le t \le T, \tag{8b}$$

$$\sum_{j=1, j\neq i}^{I} x_{ij}^t + \sum_{j=1}^{I} x_{ji}^t \le 2, \forall 1 \le i \le I, 1 \le t \le T, \tag{8c}$$

$$\eta \ge \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{t=1}^{T} \sum_{i=1}^{I} B_i^{t,n} \bar{\pi}_i^{t,n} + \sum_{t=1}^{T} \sum_{i,j=1}^{I} (A_j^{t+1} - A_i^t - t_{ij}) x_{ij}^t \bar{\varphi}_{ij}^{t,n} + \sum_{t=1}^{T} \sum_{i,j=1, i\neq j}^{I} (A_i^{t+1} - A_i^t) \bar{\mu}_{ij}^{t,n} + \sum_{t=1}^{T} \sum_{i,j=1, i\neq j}^{I} (A_j^{t+1} - A_j^t) \bar{\nu}_{ij}^{t,n} \right), \forall (\bar{\boldsymbol{\pi}}^n, \bar{\boldsymbol{\varphi}}^n, \bar{\boldsymbol{\mu}}^n, \bar{\boldsymbol{\nu}}^n, \bar{\boldsymbol{\psi}}^n) \in P_\Lambda, \tag{8d}$$

$$x_{ii}^0 = Q_i, x_{ij}^0 = 0, \forall 1 \le i, j \le I, i \neq j, \tag{8e}$$

$$x_{ii}^t = \{0, 1, 2\}, x_{ij}^t = \{0, 1\}, \forall 1 \le i, j \le I, i \neq j, 1 \le t \le T, \tag{8f}$$

$$x_{ij}^t = 0, \text{if } A_j^{t+1} - A_i^t - t_{ij} < 0. \tag{8g}$$

Problem (BMP) is actually a network flow problem with additional constraints (8d). Essentially, we have replaced the decision variables $f_{ij}^{t,n}, g_{ij}^{t,n}, d_i^{t,n}$ in problem (SAA) with these constraints, which are defined for all combinations of the extreme points in P_Λ and could have a very large size. Instead of including all of them, an iterative approach adds only a subset of constraints (8d) each time, and the resulting problem is called the *relaxed* BMP (RBMP). Such iterative approaches are referred to as Benders decomposition algorithms.

Constraints (8g) are added to make sure that no cranes are moved from block i to j if there is not enough time to complete the transfer before the next workload arrival at block j. These constraints were redundant in problem (SAA).

A Benders decomposition algorithm is given in Algorithm 1. Unlike the standard Benders algorithms that solve the RBMP with no Benders cuts and $\eta = 0$ at Step 0, we choose to solve the deterministic formulation (DP) to obtain the initial solution, as shown below.

4.3. Benders acceleration techniques

In principal the standard Benders decomposition algorithm converges after a finite number of iterations. However, in practice the number of iterations could be too large to finish within reasonable time. In this section we exploit the specific properties of the problem concerned and propose multiple acceleration techniques in order to improve the performance of the algorithm outlined above. For a comprehensive account on Benders acceleration techniques, refer to a recent survey paper due to Rahmaniani et al. (2017).

4.3.1. Regularised decomposition

One widely recognised cause of the slow convergence of standard Benders algorithms is the oscillation of the solution (to problem RBMP) from one iteration to the next. The regularised

Algorithm 1: Benders Decomposition Algorithm.

Initialisation: Set $LB \leftarrow -\infty$ and $UB \leftarrow \infty$. Choose a relative optimality tolerance ε. Let $H = 0$ represent the iteration number. Denote by $\mathbf{x}^*$ as the optimal solution.

Step 0: For the first iteration $H = 0$:

- Solve the deterministic formulation of the problem (DP), with the workload being their mean values. Denote the optimal solution by $\mathbf{x}^0$. Let $\mathbf{x}^* \leftarrow \mathbf{x}^0$.
- Solve the Benders dual sub-problem (BDS^n) for each workload realisation $1 \leq n \leq N$. Denote their optimal value by $Z_{\text{BDS}}^{0,n}$.
- Let $UB \leftarrow F(\mathbf{x}^0)$, where $F(\mathbf{x}^H) = \frac{1}{N}\sum_{n=1}^{N} Z_{\text{BDS}}^{H,n} + c_e \sum_{t=1}^{T}\left(\sum_{i=1}^{I}\sum_{j=1}^{I} q_{ij}s_{ij}x_{ij}^{t,H}\right)$,
- Generate from the BDS solutions a Benders optimality cut (8d) for the RBMP.
- Let $H = 1$ and go to Step 1.

Step 1: Re-optimise RBMP and obtain the solution $\mathbf{x}^H, \eta^H$. Let $LB \leftarrow \hat{F}^H$, where $\hat{F}^H = \eta^H + c_e \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{j=1}^{I} q_{ij}s_{ij}x_{ij}^{t,H}$. If $|(UB - LB)/LB| < \varepsilon$, terminate and the optimal solution has been obtained. Otherwise go to Step 2.

Step 2: For the current solution $\mathbf{x}^H$:

- Solve the Benders dual sub-problem (BDS^n) for each workload realisation $1 \leq n \leq N$.
- If $F(\mathbf{x}^H) < UB$, set it to be the new UB and update the optimal solution accordingly $\mathbf{x}^* = \mathbf{x}^H$.
- If $|(UB - LB)/LB| < \varepsilon$, terminate: the optimal solution has been found. Otherwise, generate from the BDS solutions a Benders optimality cut (8d) for the RBMP.
- Let $H := H + 1$ and go to Step 1.

decomposition approach was proposed by Ruszczyński and Świętanowski (1997) to mitigate such a drawback for linear problems. A quadratic term is added to the objective function to penalise the deviation from the reference point, which is the best solution found so far. Denote the reference point by $\boldsymbol{\xi}$. We modify the objective function (8a) to the following.

$$\text{(RBMP-RD)} \qquad \min: \quad \eta + c_e \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{j=1}^{I} q_{ij}s_{ij}x_{ij}^{t} + \sigma \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{j=1}^{I}(x_{ij}^{t} - \xi_{ij}^{t})^2, \tag{9}$$

where σ is the penalty weight. The equation above takes a quadratic form, leading the master problem to a quadratic integer program with linear constraints. The complete algorithm is given in Algorithm 2.

The regularised decomposition algorithm proposed in Ruszczyński and Świętanowski (1997) is designed for two stage stochastic linear problems. Recently it has been extended to multiple stage stochastic linear programs by Asamov and Powell (2018). As far as we know, the direct application of the RD method to problems with integer decision variables (in the first stage) is rarely reported in the literature. One possible hurdle is the computational burden to solve the resulting quadratic integer program (Santoso et al. (2005)). In our implementation, however, we have not found such an issue. The master problems are readily solved by the standard packages, and the solution time is not much longer than the standard Benders algorithm.

During the analysis we have found that the convergence under the RD algorithm could be *false*. In other words, even when the relative gap between $F(\boldsymbol{\xi}^H)$ and $\hat{F}^H$ is within the tolerance,

Algorithm 2: Regularised Benders Decomposition Algorithm.

Initialisation: Choose a relative optimality tolerance ε, an initial value for σ and the termination value σ_s. Let $H = 0$ represent the iteration number and $\hat{F}^0 \leftarrow -\infty$. Denote by $\mathbf{x}^*$ as the optimal solution.

Step 0: For the first iteration $H = 0$:

- Solve the deterministic formulation of the problem (DP), with the workload being their mean values. Denote the optimal solution by $\mathbf{x}^0$. Let $\mathbf{x}^* \leftarrow \mathbf{x}^0$ and $\boldsymbol{\xi}^1 \leftarrow \mathbf{x}^0$.
- Solve the Benders dual sub-problem (BDS^n) for each workload realisation $1 \leq n \leq N$. Denote their optimal value by $Z_{\text{BDS}}^{0,n}$.
- Generate from the BDS solutions a Benders optimality cut (8d) for the RBMP-RD.
- Let $H = 1$ and go to Step 1.

Step 1: Re-optimise RBMP-RD at $\boldsymbol{\xi}^H$ and obtain the solution $\mathbf{x}^H, \eta^H$. Let $\hat{F}^H = \max\{\hat{F}^{H-1}, \eta^H + c_e \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{j=1}^{I} q_{ij} s_{ij} x_{ij}^{t,H}\}$. If $|(F(\boldsymbol{\xi}^H) - \hat{F}^H)/\hat{F}^H| < \varepsilon$ and $\sigma < \sigma_s$, terminate and the optimal solution has been obtained. Otherwise go to Step 2.

Step 2: For the current solution $\mathbf{x}^H$:

- Solve the Benders dual sub-problem (BDS^n) for each workload realisation $1 \leq n \leq N$, and generate from the BDS solutions a Benders optimality cut (8d) for the RBMP-RD.
- If $F(\mathbf{x}^H) < F(\boldsymbol{\xi}^H)$ or $F(\mathbf{x}^H) = \hat{F}^H$, Let $\mathbf{x}^* \leftarrow \mathbf{x}^H$, $\boldsymbol{\xi}^{H+1} \leftarrow \mathbf{x}^H$; otherwise $\boldsymbol{\xi}^{H+1} \leftarrow \boldsymbol{\xi}^H$.
- Update the value of σ accordingly. Let $H := H + 1$ and go to Step 1.

which would mean convergence in standard Benders algorithms, the solution may not be the optimum. This is especially the case when σ is large, in which case the solution of the master problem is forced to be exactly the same as the reference point, leading to the same value of $F(\boldsymbol{\xi}^H)$ and $\hat{F}^H$. To work around this issue, we propose a slightly different approach to adjust σ on-line and only terminate the algorithm when its value is below a pre-defined threshold value σ_s (plus the standard termination criteria), as shown in Algorithm 2.

The value of σ is adjusted as follows. At each iteration H,

- if $F(\mathbf{x}^H) > \gamma^i \cdot F(\boldsymbol{\xi}^H) + (1 - \gamma^i)\hat{F}^H$, set $\sigma \leftarrow \sigma(1 + \tau^z)$,
- if $F(\mathbf{x}^H) \leq (1 - \gamma^d)F(\boldsymbol{\xi}^H) + \gamma^d \cdot \hat{F}^H$, set $\sigma \leftarrow \sigma/4$,
- otherwise keep σ unchanged,

where $\gamma^i, \gamma^d, \tau \in [0, 1]$ are the parameters controlling the updates, and z is the number of false convergences so far. We require that $\gamma^i + \gamma^d \geq 1$. Therefore, if the new solution obtained is worse or only marginally better than the reference point, increase the weight to search a smaller neighbourhood. If the solution is much better than the reference point, decrease the weight to explore a larger neighbourhood around it. In contrast to the constant increase ratio in Ruszczyński and Świętanowski (1997), the amount of increase is dampened with the number of false convergences. We also let the decrease ratio be larger than the increase ratio. Such treatments allow more iterations (and thus more searches) in-between two consecutive false convergences. When the false convergence happens, the second condition is satisfied and the σ value is reduced. Note that when the true convergence actually happens, the second condition

will always be satisfied and thus σ will be repeatedly decreased. After a couple of extra iterations it will be less than the threshold and the algorithm terminates.

4.3.2. Initial solution

In standard Benders algorithms, any feasible solution to RBMP can be used as the initial solution in Step 0, which could be far away from the optimum. A much better initial solution can be easily obtained by solving the deterministic formulation (DP), as we have already implemented in Algorithm 1. However, the subsequent solutions might escape far away from any initial solution and thus the benefit is rather limited. With a regularizing term introduced in Algorithm 2, the deviation from the initial solution and subsequent reference points can be controlled. As we shall see in the numerical experiments, the benefit of a high quality initial solution is fully materialised in the regularised decomposition method.

4.3.3. Working time inequalities

In standard Benders algorithms, the master problem contains little information on the subproblems, only loosely via the Benders cuts. As a result the solution could be of poor-quality, especially in the initial iterations. We exploit our problem structure and identify the following inequalities to be added to the master problem.

In light of the fact that $f_{ij}^{t,n}$ and $g_{ij}^{t,n}$ are all non-negative in constraints (4e), we have

$$f_{ij}^{t,n} \leq (A_j^{t+1} - A_i^t - t_{ij})x_{ij}^t, \forall 1 \leq i,j \leq I, 1 \leq t \leq T, 1 \leq n \leq N,$$
$$g_{ij}^{t,n} \leq (A_j^{t+1} - A_i^t - t_{ij})x_{ij}^t, \forall 1 \leq i,j \leq I, 1 \leq t \leq T, 1 \leq n \leq N.$$

Substituting $f_{ij}^{t,n}$ and $g_{ij}^{t,n}$ into equation (4d), after simplification we have

$$B_i^{t,n} + d_i^{t-1,n} - d_i^{t,n} \leq \sum_{j=1}^{I} \left((A_j^{t+1} - A_i^t - t_{ij})x_{ij}^t + (A_i^{t+1} - A_j^t - t_{ji})x_{ji}^t \right), \\ \forall 1 \leq i \leq I, 1 \leq t \leq T, 1 \leq n \leq N.$$

For each i,t, summarising the above constraints for all N scenarios and taking the average, we obtain the following working time inequalities

$$\bar{B}_i^t + D_i^{t-1} - D_i^t \leq \sum_{j=1}^{I} \left((A_j^{t+1} - A_i^t - t_{ij})x_{ij}^t + (A_i^{t+1} - A_j^t - t_{ji})x_{ji}^t \right), \\ \forall 1 \leq i \leq I, 1 \leq t \leq T, \tag{10}$$

where $\bar{B}_i^t$ is the average workload in block i at time period t and D_i^t is a measure of the average workload delay. So the left-hand-side is the mean workload fulfilled in each time period t in block i, while the right-hand-side is the maximum crane working time for block i during this period. We have introduced new variables D_i^t to the problem, which need to be included into the objective function as $c_d \sum_t \sum_i D_i^t$. These constraints capture the average workload information, which is important to make the crane deployment decisions and thus the solution quality could be significantly improved. A potential pitfall is that the problem size is increased slightly by

$T \cdot I$, which might have impact on the solution time.

4.3.4. Minimum crane inequalities

If there is any workload, it is expected that at least one crane should have stayed or partially served block i in period $\mathcal{T}_i^t$, which is enforced by the following minimum crane inequalities.

$$\sum_{j=1, j\neq i}^{I} x_{ij}^t + \sum_{j=1}^{I} x_{ji}^t \geq 1, \text{if } \mathbb{E}[\mathbf{B}_i^t] > 0, \forall i, t.$$

These cuts are similar to equation (10), but they do not involve any additional decision variables and could be efficient in some situations.

5. Computational results

In this section we describe a series of numerical experiments. We first consider in Section 5.1 a case study problem with real data from a major sea port in East China, and study the effectiveness of our proposal against the method in the literature that only considers workload delays in a deterministic model. In Section 5.2 the performance of the stochastic solutions is subject to comprehensive tests using a number of randomly generated problem instances. Finally, in Section 5.3 the effectiveness of the proposed Benders' acceleration techniques is investigated.

5.1. Case study of BeiLun container terminal in the port of Ningbo, East China

In this section we carry out a case study on Beilun container terminal of the port of Ningbo in East China, the 4th largest port in the world in terms of annual container throughputs. Beilun terminal has converted all their RTGs to ERTGs during the period between 2007 and 2009. The location and layout of the terminal are shown in Figure 5.

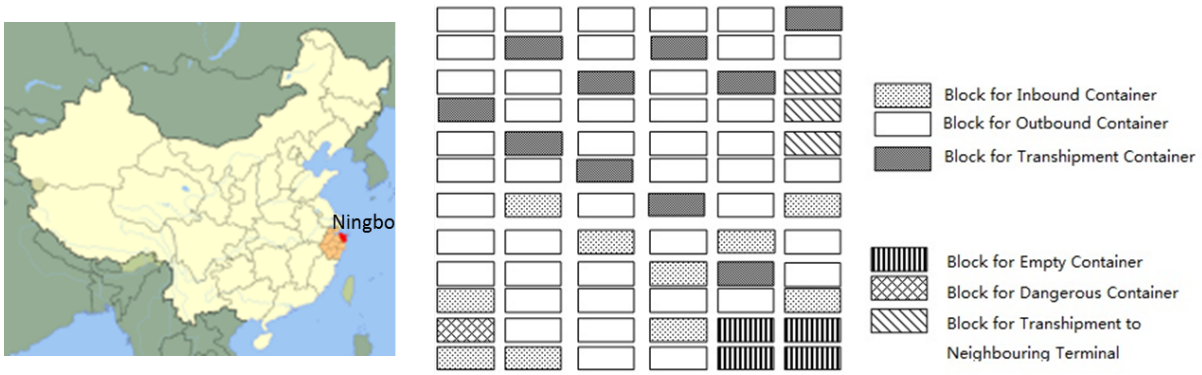

Figure 5: Beilun terminal location and layout. (Source: Ningbo Port Company Limited.)

In practical yard operations, the blocks are usually split into zones and most crane movements take place within the same zone. During each planning period, a couple of quay cranes, multiple inner trucks, and a number of ERTGs are usually grouped to work together. The containers to be loaded/unloaded to/from a vessel by the same quay cranes are usually stored in blocks located in two or three adjacent columns, so as to reduce the travelling distance of the

inner trucks. In light of this, we choose to focus on a zone of 3 rows and 3 columns with overall 9 blocks. Problems of such a scale can be solved exactly by standard solvers, which enables meaningful comparisons between multiple runs. The total number of cranes available for the zone is 9 (one crane per block on average), which are non-evenly distributed at the beginning of the planning horizon, as shown in Figure 6.

To calculate the transfer time of cranes between any two blocks, we collected data from the terminal, which includes the actual distances between blocks, the transfer routes (and thus the moving distances), and the average speed of a crane's movement (v=130 meter/min). Note that there are three truck lanes between the second and third row and thus the distance is wider. It takes an ERTG 10 minute to switch between the electricity and diesel modes ($T_s = 10$min), and 1 min to make a 90-degree turn ($T_r = 1$min). The transfer times are listed in Table 1, which are derived according to the approach presented in Section 3.1. Other data collected for ERTGs include the average diesel consumption rate (0.02 litre/meter, and thus 2.6 litre/min), and the average electricity consumption rate (1.5 kwh/min).

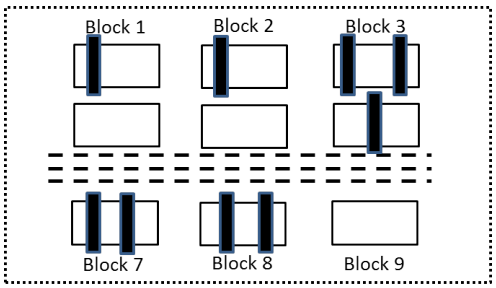

Figure 6: Layout and the initial position of ERTGs in the case study.

Blocks	1	2	3	4	5	6	7	8	9
1	0	2	29	25	25	29	25	25	30
2		0	2	25	25	25	25	25	25
3			0	29	25	25	30	25	25
4				0	2	30	25	25	30
5					0	2	25	25	25
6						0	30	25	25
7							0	2	30
8								0	2
9									0

Table 1: Inter-block transfer time (in minutes).

Based on the literature (The U.S. Energy Information Administration), each litre of diesel consumed could produce 2.5 kg CO2, while each kwh of electricity consumed could produce 0.778 kg CO2 if generated by oil. Therefore we have $q_1 = 0.778 * 1.5 = 1.2$kg/min and $q_2 = 2.5 * 2.6 = 6.5$kg/min. It is clear that the oil generated electricity produces just one-fifth of the CO2 that diesel engines produce. The emission will be even less if the electricity is generated by renewable energy sources.

The daily work in Beilun terminal is organised into 3 shifts. During each shift the cranes can be redeployed twice. Therefore the planning horizon comprises 6 time periods, of 4 hours each ($T = 6, C = 4$ h). Without loss of generality, we let $c_d = 1$ always and vary $c_e \in \{0.0, 0.2, \cdots, 2.0\}$. If $c_e = 0$ the model reduces to the traditional approach that is only concerned with workload delays.

The arrival time A_i^t for block i in time period t is randomly generated according to the uniform distribution $U[(t-1)C, tC], \forall 1 \leq t \leq T$. The amount of workload $\mathbf{B}_i^t$ follows a symmetric triangular distribution with parameters $(0.25b_i^t, 1.75b_i^t)$, where $b_i^t = \mathbb{E}[\mathbf{B}_i^t]$ is the mean workload. Its value is set as $b_i^t = 0.8(A_i^{t+1} - A_i^t)$. Therefore it would be necessary to have two cranes work in those blocks where the amount of workload is too high to be handled without delay by a single crane. Note that the proposed solution framework is not restricted to specific

distributions. Indeed, normal distributions with different degrees of variation have been used in the next section.

We generate a sample of 20 realisations of $\mathbf{B}$ based on their distributions. This sample is included in the SAA approach to develop a stochastic deployment plan. We also solve a deterministic (DP) problem with the mean workload b_i^t. Both problems are solved with CPLEX12.6 directly; a time limit of 3 hours is imposed. In order to evaluate the performance of the solutions, another sample of 100 workload realisations is generated. For each realisation the resulting workload delay is calculated by solving the problem (3a-3g) with the given $\mathbf{x}$ values under each deployment plan. CO2 emissions are calculated via the formula $\sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{j=1}^{I} q_{ij}s_{ij}x_{ij}^t$. Note that CO2 emissions are completely determined by the deployment plan and do not change with the sample. The average amount of workload delays over the sample and CO2 emissions are plotted in Figure 7 for different c_e values. Some interesting and important findings can be observed from the results.

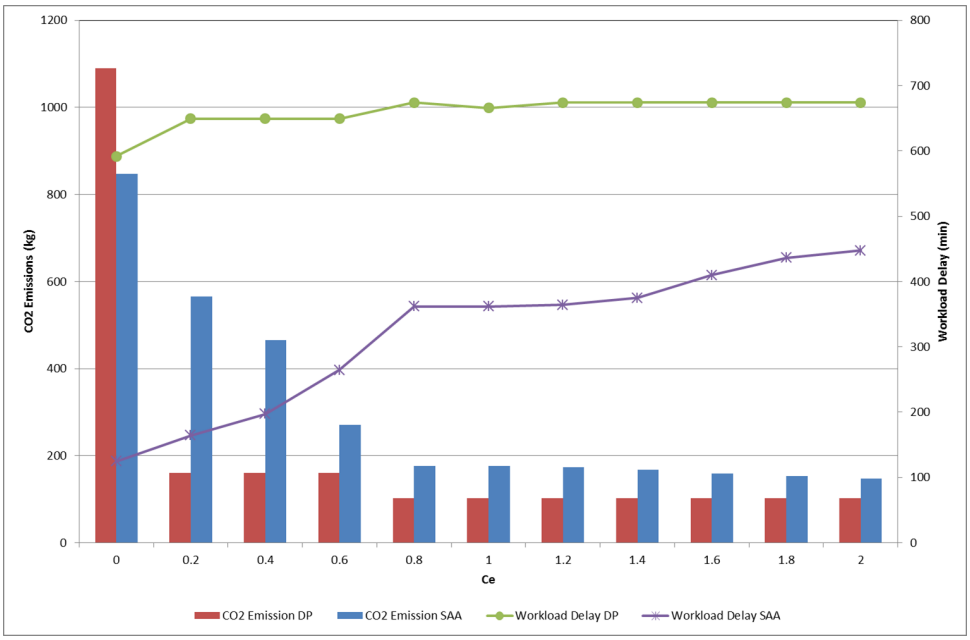

Figure 7: Performance of SAA and DP solutions with various c_e values.

Firstly, the results show that the traditional approach that is only concerned with workload delays could lead to significant amount of CO2 emissions (see both bars for $c_e = 0$). When the workload is deemed as deterministic, the introduction of the penalty term c_e for CO2 emissions, even as small as 0.2, immediately reduces the CO2 emissions by remarkably 85% (930kg). This is at a cost of increased workload delays, but only by a small amount of 9%. Further increases of c_e values continue to reduce CO2 emissions, and the resulting increase of workload delays is marginal. Similar patterns are also observed in the SAA results where the workload uncertainty is considered.

Secondly, the SAA solutions significantly reduce the workload delays compared to their deterministic counterparts. The former achieves much smaller mean values for all c_e, although in most cases this is at an expense of CO2 emissions. The SAA approach recommends more crane movements (and thus leads to more CO2 emissions) to deal with the workload uncertainty.

The extra amount is however limited and reduces continuously with c_e. The only exception is when $c_e = 0$, in which case the SAA solution cuts both workload delays (by 79%) and CO2 emissions (by 22%). These results further highlight the importance of the consideration of workload uncertainties in the crane deployment.

One might suggest to decrease the c_e value to allow more CO2 emissions and reduce the workload delays for the DP approach. However, the results show that even when $c_e = 0$ it still leads to much higher amount of workload delays than SAA, and the corresponding CO2 emissions will increase drastically, as we have mentioned above. Therefore, to achieve a similar amount of expected workload delays, the deterministic solution emits much more CO2 than the stochastic one. Similarly, with a similar amount of CO2 emissions, the former results in much higher workload delays than the latter.

These results also provide important information for port operators to find the necessary trade-off between delayed workload and CO2 emissions, through the appropriate selection of c_d and c_e values. There is however not a universal choice that fits all scenarios in practice. More discussion around this matter can be found in Section 5.2.

Finally, it is worth mentioning that this case study considers a heavy workload scenario, and a workload distribution with rather large variation of coefficient. In the next section, we test the performance of our proposed approach comprehensively in various situations.

5.2. Quality of stochastic solutions

We consider a number of scenarios with various values of the key problem features, which include the layout of the yard (number of columns K and rows L), the initial location of cranes Q_i, workload arrival times A_i^t, and the distribution for the workload $\mathbf{B}_i^t$. These values are sampled as below.

$$(K, L) \in \{(2, 5), (3, 3), (5, 2)\}, \tag{11a}$$

$$Q_i \begin{cases} = 1, & \text{(Evenly distributed)} \\ \sim DU[0, 2], & \text{(Non-evenly distributed)} \end{cases}, \forall 1 \leq i \leq I, \tag{11b}$$

$$A_i^t \sim U[(t-1)C, tC], \forall 1 \leq i \leq I, 1 \leq t \leq T, \tag{11c}$$

$$\mathbf{B}_i^t \sim \mathcal{N}(b_i^t, (b_i^t \delta)^2), \forall 1 \leq i \leq I, 1 \leq t \leq T, \text{where:} \tag{11d}$$

$$b_i^t = \begin{cases} 0.6(A_i^{t+1} - A_i^t), & \text{(Low workload)} \\ 0.8(A_i^{t+1} - A_i^t), & \text{(High workload)} \end{cases}, \text{and} \tag{11e}$$

$$\delta = \begin{cases} 0.1, & \text{(Low uncertainty)} \\ 0.2, & \text{(Moderate uncertainty)} \\ 0.3, & \text{(High uncertainty)} \end{cases}. \tag{11f}$$

We consider 3 different layouts of the blocks, which have different shapes and together cover a wide range of scenarios of the terminal yard zones. For example, a $(5, 2)$ zone is typical for unloading a vessel and a $(2, 5)$ zone is typical for loading a vessel. The total number of cranes is set to be the same as the number of blocks. For each layout, we consider two scenarios of the initial location of the cranes, i.e., evenly and non-evenly distributed. Therefore in the first

scenario, one crane is allocated to each block at the beginning of the time horizon. In the second scenario, cranes are assigned to a random block. Since the maximum number of cranes of any block is two, whenever a block has been allocated two cranes, it is removed from further crane allocations. The workload arrival times are generated according to uniform distributions (11c). The amount of workload is assumed to follow normal distributions. We consider two levels of the workload, i.e., high and low. The standard deviations are set at three levels, with the coefficient of variation δ ranging from 0.1 to 0.3. Therefore for each layout we have generated 12 testing instances in total.

The energy consumption parameters for ERTGs and other crane specific coefficients are as specified in Section 5.1. The transfer times between blocks for the other two layouts are listed in Table 2. Without loss of generality, we have chosen $c_d = 1$ and $c_e = 1.2$ for all experiments. Still we have $T = 6$ and $C = 4$ h.

Block	1	2	3	4	5	6	7	8	9	10
1	0	2	29	25	32	34	25	25	29	34
2		0	2	29	32	25	25	25	29	32
3			0	2	29	29	25	25	25	29
4				0	2	32	29	25	25	25
5					0	34	32	29	25	25
6						0	2	29	32	34
7							0	2	29	32
8								0	2	32
9									0	32
10										0

(a) (2,5)

Block	1	2	3	4	5	6	7	8	9	10
1	0	2	25	25	25	29	25	25	26	26
2		0	25	25	25	25	25	25	26	26
3			0	2	25	25	25	25	26	26
4				0	25	25	25	25	36	26
5					0	2	25	25	25	25
6						0	25	25	25	25
7							0	2	25	25
8								0	25	25
9									0	2
10										0

(b) (5,2)

Table 2: Inter-block transfer times in minutes for different layouts.

All experiments are carried out on high performance computing clusters. Each node in the cluster has 256GB RAM and two Intel Xeon E5 processors, and each processor has multiple cores with a CPU speed of 2.50GHz. Unless specified otherwise, a 3 hour solution time limit has been specified for all models.

As in Section 5.1, for each testing problem, a sample of $N = 20$ workload realisations is generated according to (11d). If a sampled value is out of the range $[b_i^t - 3\delta b_i^t, b_i^t + 3\delta b_i^t]$, it is discarded and re-sampled. The resulting sample represents 99.7% of the data and excludes negative values. This sample is used in the SAA framework to derive a stochastic deployment plan. This procedure is repeated for 10 times and the one with the minimum objective function value is selected as the final solution. Again we solve a (DP) model for each problem and develop a deterministic solution. Moreover a benchmark approach which is deterministic and does not consider CO2 emissions ($c_e = 0$) is also included. All models are solved with CPLEX12.6 directly. A much larger sample of $N' = 100$ is generated to evaluate the workload delays, in the same way as specified in Section 5.1.

The results of CO2 emissions and the average workload delays are summarised in Table 3, where the top (bottom) table lists the results for the problems where the cranes are initially evenly (non-evenly) distributed. It is shown in Table 3a that, when cranes are evenly distributed, the difference between the SAA and DP solutions is negligible when the workload is low, regardless the level of workload uncertainties. No inter-block crane movements are suggested by either solution. When the workload increases, the DP solution still keeps all cranes

in their initial locations. This is not surprising as the average workload is only 80% of their availability time. In contrast, the SAA solution suggests to move cranes around to reduce the workload delays. The results show that with a small amount of CO2 emissions, the average workload delays are significantly reduced, especially when the uncertainty is high (up to 51% reduction). Such results are true for all the three layouts.

Layout	Measure	Low Workload									High Workload								
		$\delta = 0.1$			$\delta = 0.2$			$\delta = 0.3$			$\delta = 0.1$			$\delta = 0.2$			$\delta = 0.3$		
		Bench	DP	SAA	Bench	DP	SAA	Bench	DP	SAA	Bench	DP	SAA	Bench	DP	SAA	Bench	DP	SAA
(2, 5)	CO2 (kg)	708	0	0	708	0	0	708	0	0	1178	0	0	1178	0	31	1178	0	99
	Average Delay (m)	0	0	0	2	0	0	17	12	12	25	3	3	161	142	106	587	680	336
(3, 3)	CO2 (kg)	588	0	0	588	0	0	588	0	0	907	0	0	907	0	17	907	0	73
	Average Delay (m)	6	0	0	28	0	0	78	14	14	17	2	2	132	128	110	377	529	318
(5, 2)	CO2 (kg)	451	0	0	451	0	0	451	0	0	435	0	0	435	0	23	435	0	93
	Average Delay (m)	2	0	0	33	0	0	57	13	13	25	2	2	197	159	131	586	641	361

(a) Cranes initially evenly distributed.

Layout	Measure	Low Workload									High Workload								
		$\delta = 0.1$			$\delta = 0.2$			$\delta = 0.3$			$\delta = 0.1$			$\delta = 0.2$			$\delta = 0.3$		
		Bench	DP	SAA	Bench	DP	SAA	Bench	DP	SAA	Bench	DP	SAA	Bench	DP	SAA	Bench	DP	SAA
(2, 5)	CO2 (kg)	2279	83	83	2279	83	83	2279	83	88	2113	**200**	**159**	1709	**200**	**165**	1709	200	213
	Average Delay (m)	10	67	67	39	93	93	99	142	124	146	**209**	**235**	295	**439**	**414**	626	897	605
(3, 3)	CO2 (kg)	644	71	71	644	71	71	644	71	74	1090	103	106	1090	103	125	1090	103	170
	Average Delay (m)	13	16	16	27	23	23	77	51	48	71	60	50	229	249	168	503	590	321
(5, 2)	CO2 (kg)	381	63	63	381	63	63	381	63	63	690	66	66	690	66	85	690	66	151
	Average Delay (m)	0	0	0	1	3	0	15	18	14	41	5	5	215	145	114	673	622	304

(b) Cranes initially non-evenly distributed.

Table 3: Performance of the deployment solutions due to alternative models.

When the cranes are not initially located evenly across the blocks, inter-block crane transfers become always necessary in both solutions. Similar to the results in Table 3a, the difference between these two solutions increases with the amount of workload and the level of uncertainties, as shown in Table 3b. For the high uncertainty scenarios in particular, a few extra transfers (and thus extra CO2 emissions) would significantly reduce the workload delays for the SAA solutions.

Compared with the other two alternatives, the benchmark approach produces the worst performance in all situations. It suggests frequent inter-block transfers that emit a large amount of CO2, but is still unable to reduce the average workload delays most of the time. These results highlight the effectiveness of the consideration of CO2 emissions in the crane deployment.

It is worth mentioning that in some situations the DP solution suggests more moves than the SAA. Usually these extra moves are unnecessary and do not reduce at all the expected workload delays, as shown in $\delta = 0.2$ cases for the (2,5) layout in Table 3b. In some other cases, these moves may lead to less amount of delays, such as those $\delta = 0.1$ cases. These behaviours are due to the unbalance between the workload and cranes' initial locations, which results in frequent inter-block transfers. For the layouts with a lot of columns such as (2,5), many of these transfers are between non-adjacent columns. Such moves are the most time consuming and emit the most amount of CO2. As a result SAA rather proposes to move less, at an expense of some longer delays. Note that no matter which situation SAA always produces lower overall objective function values than DP. If necessary, one could always reduce the value of c_e to encourage SAA to suggest more moves in order to reduce the workload delays, especially for the yard zones with

multiple columns.

To understand the robustness of SAA and DP solutions, we plot the workload delays over the 100 realisations in each scenario. For brevity we only present the boxplot results for the layout of (3,3) in Figure 8. It is shown that for both solutions the variation of workload delays increases with the amount of workload and their uncertainties. There is no significant difference between them in the low workload scenarios. However, when the workload is high, the delays due to DP have much larger variation than those due to SAA. Moreover, the latter achieves smaller amount of delays in all the quartiles and the worst cases.

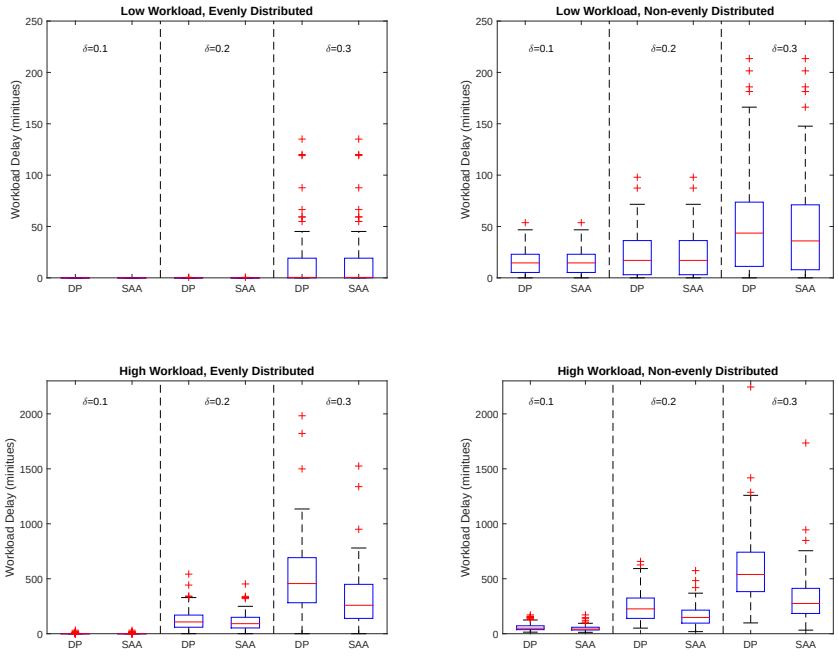

Figure 8: Workload delays of the DP and SAA solutions for the (3,3) layout.

It is therefore clear that SAA is stronger in terms of the overall performance (CO2 emissions and workload delays). It is also more robust compared to DP, especially for high workload scenarios with also high uncertainties. It is worth mentioning that, with cleaner electricity sources such as wind or solar, the advantage of SAA will become even more obvious as less CO2 will be emitted to achieve the similar results of workload delays.

5.3. Efficiency of the Benders decomposition algorithms

We first study the performance of the proposed Benders acceleration techniques and their combinations in Section 5.3.1. Then in Section 5.3.2 the strongest acceleration techniques are compared against the direct SAA approach, with a range of sample sizes.

5.3.1. Performance of Benders acceleration techniques

For convenience we denote each of acceleration techniques as follows: regularised decomposition (RD), working time inequalities (WT), and minimum crane number inequalities (MC). For all the acceleration methods the initial solution is obtained by solving the (DP) problem. Its benefit has been clearly reported in Ruszczyński and Świętanowski (1997).

We focus our attention on selected (3,3) problems with moderate workload uncertainties and a sample size of $N = 40$. Both initially evenly and non-evenly located cranes, and both low and high workload scenarios are considered, leading to four instances all together. Each method is applied 30 times to each instance. In our experiments the parameters are set as $\gamma^i = 0.9, \gamma^d = 0.1, \tau = 0.8$, the initial value of $\sigma = 2$ and $\sigma_s = 0.5$. A time limit of 3600s is imposed.

Figure 9 plots the trajectory of the solution error over time for the instance with low workload and non-evenly located cranes. For the methods without the regularising term the solution error is calculated as $100(UB - LB)/UB$, otherwise it is calculated as $100(F(\boldsymbol{\xi}^H) - \hat{F}^H)/F(\boldsymbol{\xi}^H)$.

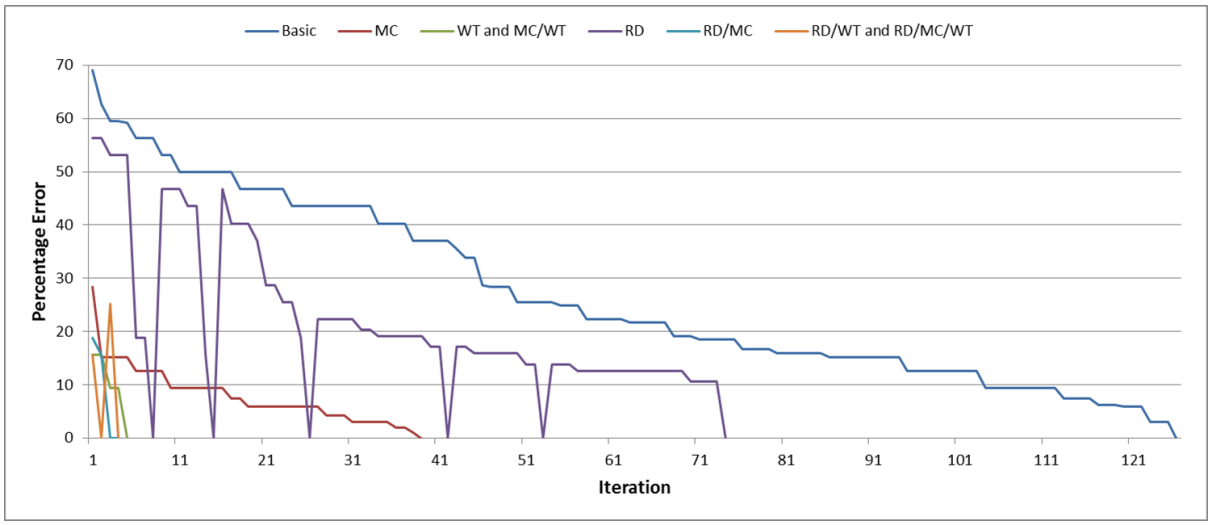

Figure 9: Solution error over time for the instance with low workload and non-evenly distributed cranes.

It is clear that all the acceleration techniques contribute to the reduction of convergence time. The effectiveness of the inequalities is particularly significant. Compared to the basic algorithm that converges after 126 iterations and 3200s, MC takes only a third of iterations to converge, cutting the solution time by 67%. WT is even more efficient; it takes only 5 iterations and merely 114s. The RD method alone is not as impressive as the inequalities; yet it takes only half of iterations to converge. Its performance is hugely improved along with the inequalities. Indeed, the method with RD and WT together achieves the fastest convergence in this instance, taking only 99s. It is worth noting that for the RD methods, the solution error drops to zero multiple times. These are the points where the false convergences happen. They terminate only when the σ value drops to below the threshold.

The full results are summarized in Table 4. For the low workload problems all methods converge. For the high workload ones, however, none of them converges within the time limit. Therefore the CPU time is reported for the former instances (Table 4a), while the percentage sub-optimality is reported for the latter ones (Table 4b). Note that for the sample size of 40 the optimal solution of the SAA problem can be obtained, and thus the sub-optimality of the final solution from each Benders method can be evaluated. We also include in the table the average number of iterations and the average CPU time of the master problems per iteration.

For the low workload scenario, the problems with evenly distributed cranes are trivial. The results for the non-evenly distributed problems are similar to what is shown in Figure 9. The

	Evenly			Non-evenly		
Acceleration method	CPUs	Iterations	CPUs/iteration	CPUs	Iterations	CPUs/iteration
Basic	**2**	1	0.2	2886	139	20
MC	3	1	0.3	875	40	20
WT	3	1	0.2	113	5	13
MC/WT	**2**	1	0.2	**107**	5	13
RD	3	2	0.1	2005	83	24
RD/MC	3	2	0.1	459	21	19
RD/WT	3	2	0.1	109	4	16
RD/MC/WT	3	2	0.1	108	4	15

(a) Low Workload.

	Evenly			Non-evenly		
Acceleration method	Sub-optimality(%)	Iterations	CPUs/iteration	Sub-optimality(%)	Iterations	CPUs/iteration
Basic	3.00	140	25	14.04	113	31
MC	2.56	111	32	11.82	154	23
WT	1.68	133	27	6.94	167	21
MC/WT	1.73	128	28	7.23	189	19
RD	**0.67**	100	36	1.44	98	36
RD/MC	0.71	107	34	2.49	99	36
RD/WT	**0.67**	103	35	1.51	110	32
RD/MC/WT	0.70	109	33	**1.38**	119	30

(b) High Workload.

Table 4: Performance of acceleration techniques for the (3,3) layout instances with moderate workload uncertainty.

strongest methods are MC/WT and RD/MC/WT (or RD/WT). Again the RD itself provides some but not much enhancement. The solution time for the master problems does not increase much for the RD methods, only a few more seconds per iteration. The valid inequalities clearly help cut the solution time.

The significance of the RD method is much more obvious in the high workload scenario. For problems with evenly distributed cranes, RD and RD/WT are the strongest methods, achieving a sub-optimality of 0.67%. The other two RD methods follow closely behind. In contrast, those without RD show much weaker performance. Such results are even more distinct in the non-evenly distributed crane problems. All the RD methods comfortably beat their counterparts without RD. In particular, the gap between RD and Basic is as large as 13%. Again, there is no significant difference in the solution time for the master problems between these techniques.

To summarise, the inequalities contribute to shorter solution time of each iteration and tighter lower bounds, while the RD method contributes to the identification of higher quality solutions. In the following experiments we focus on the combination of these techniques and compare their performance against the direct SAA solution for larger sample sizes.

5.3.2. Comparison between the Benders algorithms and the direct SAA solution

We compare the performance of the strongest Benders decomposition algorithm (RD/MC/WT) identified in the previous section against the direct SAA solution by CPLEX for different sample sizes, which vary between 40 and 200 at an increment of 40. The problem instances are still generated according to (11b-11f), and each instance is solved repeatedly for 30 times with both algorithms. All the results reported below are the average of the 30 solutions. We still set a time limit of 3600s for all experiments.

The total CPU time (in seconds) is plotted in Figure 10. It is shown that for both algorithms their solution time increases with the workload level and the uncertainties. They also spend longer time to solve the problems in which the cranes are initially non-evenly distributed. For the low workload instances Benders is very efficient and beats SAA in most cases. It finds the optimal solution much quicker than SAA, especially when the sample size is large. For the high workload scenarios the Benders' solution time is longer than SAA for smaller sample sizes (except the low uncertainty instance with evenly distributed cranes). For larger sample sizes both algorithms reach the time limit for problems with moderate and high uncertainties.

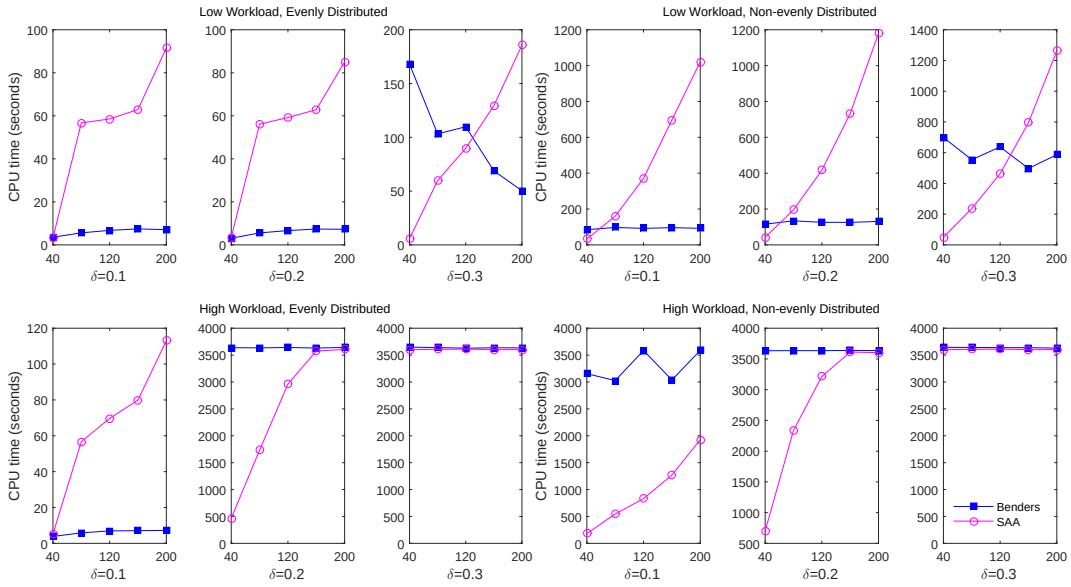

Figure 10: The CPU time with sample size N for layout (3,3) for Benders and SAA algorithms, with a time limit of 3600s.

The most distinct feature of the Benders algorithm is that its solution time does not increase much with the sample size. Remember that we have not yet implemented parallel computation for the Benders sub-problems. On the contrary, the SAA solution time increases rapidly with N. The clear contrast between these two algorithms is evident in every scenario in Figure 10. It is worth mentioning that the Benders' solution time obviously decreases with the sample size in one particular scenario (low workload, evenly distributed cranes, and high uncertainty). We have observed that in this scenario less iterations are used to prove convergence thanks to tighter lower bounds for the larger sample sizes.

Table 5 lists the solution difference between these two algorithms in terms of the relative percentage gap of SAA's solution over Benders'. Therefore, a positive gap means that SAA results in a larger objective function value than Benders, and thus produces weaker performance, and vice versa. For the low workload instances both algorithms are able to find the optimal solutions and the gap is always zero. Therefore only the results for the high workload instances are included in the table.

It is shown that there is no difference for problems with low uncertainties. With the increase of the workload uncertainty their solutions become more apart and the gap increases quickly with the sample size. SAA initially produces marginally better solutions than Benders for small

	Evenly			Non-evenly		
N	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.3$	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.3$
40	0	-1	-1	0	-2	-1
80	0	-1	-1	0	-2	1
120	0	0	4	0	-1	2
160	0	0	40	0	3	35
200	0	2	1401	0	1947	1196

Table 5: Relative solution gaps (in percentage) between SAA and Benders for high workload instances with the (3,3) layout.

sample sizes. With the increase of the sample size the former's solution quickly deteriorates, resulting in larger gaps between them. This is not surprising as SAA's solution time increases quickly with N. For larger sample sizes, Benders comfortably beats SAA in all situations.

6. Conclusions

In this paper we study the deployment of electric powered rubber-tyred gantry cranes in container terminals. Compared to conventional RTGs, the upgrade to electric powered gantry cranes saves energy and produces less CO2 emissions, even though their efficiency could be reduced due to the extra time required to switch between the electric and diesel modes. Considering the distinctive features of ERTGs, we have proposed to include both workload delays and CO2 emissions into the deployment decisions. In light of the uncertainty in the workload estimation, a two-stage stochastic program has been formulated and a sample average approximation solution framework has been developed. Due to the computational challenges for large sample sizes, we have employed a Benders decomposition algorithm within the SAA framework. Multiple acceleration techniques have been proposed to speed up the solution process, which includes a tailored regularised decomposition method and valid inequalities.

The case study with sample data from a major sea port in East China show that, compared to the approaches in the literature that ignore CO2 emissions all together, our proposal could significantly reduce CO2 emissions with only a marginal compromise in workload delays. Moreover, the stochastic deployment solutions due to SAA are much more robust than its deterministic counterpart, leading to much lower average workload delays and slightly higher CO2 emissions. Our numerical results confirm that the solution time of the SAA approach increases rapidly with the sample size. The accelerated Benders decomposition algorithms effectively address this difficulty, with much improved solutions within the same time limit. This is especially the case for busy storage yards with the moderate to high level of workload uncertainties. Our results also show that the regularised decomposition method does help reduce the convergence time for combinatorial first stage problems. Moreover, if combined with other techniques, such as the valid inequalities, its effectiveness could be significantly enhanced.

Future research could take into consideration the actual arrival times of individual containers, and address the yard crane deployment and subsequent crane scheduling simultaneously. Novel modelling and solution approaches will be required to address the complexity of the integrated problem. Another area of interest is the collaborative scheduling of quay cranes and yard cranes considering multiple uncertainties.

Acknowledgements

This work was supported by the National Natural Science Foundation of China [grant number 71172076].

References

Asamov, T., Powell, W.B., 2018. Regularized decomposition of high-dimensional multistage stochastic programs with Markov uncertainty. SIAM Journal on Optimization 28, 575–595.

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

Carlo, H.J., Vis, I.F., Roodbergen, K.J., 2014a. Storage yard operations in container terminals: Literature overview, trends, and research directions. European Journal of Operational Research 235, 412–430.

Carlo, H.J., Vis, I.F., Roodbergen, K.J., 2014b. Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. European Journal of Operational Research 236, 1–13.

Carlo, H.J., Vis, I.F., Roodbergen, K.J., 2015. Seaside operations in container terminals: literature overview, trends, and research directions. Flexible Services and Manufacturing Journal 27, 224–262.

Cheung, R.K., Li, C.L., Lin, W., 2002. Interblock crane deployment in container terminals. Transportation Science 36, 79–93.

Guo, X., Huang, S.Y., 2012. Dynamic space and time partitioning for yard crane workload management in container terminals. Transportation Science 46, 134–148.

Guo, X., Huang, S.Y., Hsu, W.J., Low, M.Y.H., 2011. Dynamic yard crane dispatching in container terminals with predicted vehicle arrival information. Advanced Engineering Informatics 25, 472–484.

He, J., Chang, D., Mi, W., Yan, W., 2010. A hybrid parallel genetic algorithm for yard crane scheduling. Transportation Research Part E: Logistics and Transportation Review 46, 136–155.

He, J., Huang, Y., Yan, W., 2015. Yard crane scheduling in a container terminal for the trade-off between efficiency and energy consumption. Advanced Engineering Informatics 29, 59–75.

Jung, S.H., Kim, K.H., 2006. Load scheduling for multiple quay cranes in port container terminals. Journal of Intelligent manufacturing 17, 479–492.

Kelywegt, A., Shapiro, A., Homem-De-Mello, T., 2002. The sample average approximation method for stochastic discrete optimization. SIAM Journal of Optimization 12, 479–502.

Kim, K.Y., Kim, K.H., 2003. Heuristic algorithms for routing yardside equipment for minimizing loading times in container terminals. Naval Research Logistics 50, 498–514.

Lee, D.H., Cao, Z., Meng, Q., 2007. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. International Journal of Production Economics 107, 115–124.

Li, W., Wu, Y., Petering, M.E., Goh, M., de Souza, R., 2009. Discrete time model and algorithms for container yard crane scheduling. European Journal of Operational Research 198, 165–172.

Linn, R.J., Zhang, C.Q., 2003. A heuristic for dynamic yard crane deployment in a container terminal. IIE Transactions 35, 161–174.

Lu, Y., Le, M., 2014. The integrated optimization of container terminal scheduling with uncertain factors. Computers & Industrial Engineering 75, 209–216.

Ng, W., 2005. Crane scheduling in container yards with inter-crane interference. European Journal of Operational Research 164, 64–78.

Ng, W., Mak, K., Tsang, W., 2006. Scheduling yard crane in a port container terminal using genetic algorithm. International Journal of Industrial Engineering: Theory Applications and Practice .

Peng, Y., Wang, W., Song, X., Zhang, Q., 2016. Optimal allocation of resources for yard crane network management to minimize carbon dioxide emissions. Journal of Cleaner Production 131, 649–658.

Port Technology, 2014. RTG conversion finishes ahead of schedule. URL: `https://www.porttechnology.org/news/rtg\_to\_ertg\_conversion\_finishes\_ahead\_of\_schedule`. accessed: 01/2017.

Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2017. The Benders decomposition algorithm: A literature review. European Journal of Operational Research 259, 801–817.

Roy, D., de Koster, M., 2018. Stochastic modeling of unloading and loading operations at a container terminal using automated lifting vehicles. European Journal of Operational Research 266, 895 – 910.

Ruszczyński, A., Świętanowski, A., 1997. Accelerating the regularized decomposition method for two stage stochastic linear problems. European Journal of Operational Research 101, 328–342.

Santoso, T., Ahmed, S., Goetschalckx, M., Shapiro, A., 2005. A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operational Research 167, 96–115.

Sha, M., Zhang, T., Lan, Y., Zhou, X., Qin, T., Yu, D., Chen, K., 2016. Scheduling optimization of yard cranes with minimal energy consumption at container terminals. Computers & Industrial Engineering .

Shapiro, A., 2003. Monte carlo sampling methods. Handbooks in operations research and management science 10, 353–425.

The U.S. Energy Information Administration, 2016. URL: http://www.eia.gov. accessed: 01/2017.

VAHLE, 2011. VAHLE receives ERTG conversion project order from modern terminals limited in Hong Kong. URL: http://www.vahleinc.com/vahle-receives-ertg-conversion-project.html. accessed: 01/2017.

World Shipping Council, 2018a. Trade statistics. URL: http://www.worldshipping.org/about-the-industry/global-trade/trade-statistics. accessed: 07/2018.

World Shipping Council, 2018b. Top 50 world contrainer ports. URL: http://www.worldshipping.org/about-the-industry/global-trade/top-50-world-container-ports. accessed: 07/2018.

Xu, H., Zhang, D., 2009. Smooth sample average approximation of stationary points in nonsmooth stochastic optimization and applications. Mathematical Programming 119, 371–401.

Zhang, C., Wan, Y.w., Liu, J., Linn, R.J., 2002. Dynamic crane deployment in container storage yards. Transportation Research Part B: Methodological 36, 537–555.