Optimal planning for container prestaging, discharging, and loading processes at seaport rail terminals with uncertainty

Ying Xie[a], Dong-Ping Song[b],*

[a] *Lord Ashcroft International Business School, Anglia Ruskin University, Bishop Hall Lane, Chelmsford CM1 1SQ, UK*
[b] *School of Management, University of Liverpool, Chatham Street, Liverpool L69 7ZH, UK*

A B S T R A C T

This paper considers the optimal planning problem for container prestaging and dynamic discharging/loading at seaport rail terminals subject to uncertainties. We formulate the problem into a stochastic dynamic programming model to minimise the total logistics cost. Four simplified strategies are proposed. Experiments based on a real case study are performed to compare the optimal strategy with four simplified strategies in terms of cost, computing time and practicality. Results show that the seaport can use prestaging to reduce the cost involved and that an appropriate discharging/loading plan can be created in response to different prestaging situations and uncertain scenarios.

1. Introduction

Global Container port traffic has increased from nearly 85 million TEUs (twenty-foot equivalent unit) in 1990 to 700 million TEUs in 2016. Meanwhile, the size of containerships has increased dramatically, from 5000 TEU in 1990 to 21400 TEU in 2017 (UNCTAD, 2017). The growing traffic volume puts a huge pressure on container ports, which act as critical interface points between seaborne transport and hinterland transport. The deployment of mega-vessels tends to concentrate container loading/unloading activities at hub and gateway ports, adding further pressure to container ports to cope with the surge effect of container flows.

In addition, traffic congestion and emissions (e.g. mainly caused by truck movements) in the areas surrounding seaports have raised serious environmental concerns. Rail transport is regarded as one of the most effective ways to tackle the above challenges due to its high capacity and low emissions. However, the limited handling capacity in seaport rail terminals is not able to match the growing demand; thus there is a compelling need to improve the efficiency of rail terminal operations at seaports in order to achieve sustainability and reduce pollution and congestion in container transport chains.

In a broad context, numerous studies have been conducted on the optimisation of container terminal operations. Readers can refer to the survey papers such as Steenken et al. (2004); Gunther and Kim (2006); Stahlbock and Voß (2008); Bierwirth and Meisel (2010); Carlo et al. (2014); and Kim and Lee (2015). However, it is noted that most of the optimisation models focused on quayside and yardside operations with a relatively limited number of studies on landside operations interfacing with hinterland transport activities. In particular, the operations management issues associated with seaport rail terminals have been under-studied (Caballini et al., 2016; Heggen et al., 2016).

Trains arrive at a seaport rail terminal, carrying export containers which are discharged and transported to storage yards close to ship berth terminals. Then, import containers are loaded onto trains either directly from storage yards or from the rail terminal

---

buffer, where import containers have been prestaged (stored in advance). Both the discharge process and the load process have to be carried out within pre-specified working time windows; in particular each train has a strict departure time due to railway network scheduling. There is often a considerable distance (up to several miles) between rail terminals and storage yards, yard cranes serve multiple tasks, and other factors exist such as weather conditions, equipment breakdown, etc; therefore, a high level of uncertainty is involved in the transportation of containers between the storage yards and the train. In a similar manner to how yard template planning is used to determine the assignment of spaces in a container port prior to a vessel's arrival (Zhen, 2014), a container prestaging mechanism is introduced in this research. Prestaging refers to moving containers from storage yards to the rail terminal buffer in advance, in order to shorten the distance they need to be moved at time critical points so as to mitigate risks imposed by various uncertainties and to better meet the departure deadline. However, prestaging has to be appropriately planned as prestaging activity is costlier and rail terminal buffer space is limited. The discharging and loading processes are performed within planned time windows. A time window may be divided into multiple periods (e.g. hourly periods). Flow rates are defined as container movements between the rail terminal and the storage yards at each period, which may change dynamically over the discharging and loading time windows. In this paper, we propose an integrated model to optimise the decisions of container prestaging and of container flow rates in the presence of uncertainties.

To the best of our knowledge, we are the first to investigate container prestaging issue at rail terminal. The main contributions of this research include: (i) this study investigates the issues of simultaneously optimising container prestaging and container flow rates at seaport rail terminals subject to time windows. (ii) we obtain the optimal strategy using the stochastic dynamic programming method. To overcome the challenge of the computational complexity, we propose four simplified strategies, which are more computationally efficient and readily applicable in practice. In particular, the Bang-bang strategy can achieve a performance very close to the optimal strategy. (iii) The current industrial practice relating to planning prestaging, discharging and loading is manual and time consuming; therefore, this work offers the opportunity for terminal operators to improve the efficiency of rail terminal operations.

The rest of the paper is organised as follows. Section 2 reviews the relevant literature together with an empirical example to motivate the study. In Section 3, an integrated stochastic dynamic programming model is formulated, which is able to optimise the decisions of container prestaging and the decisions of container discharging and loading rates in the presence of uncertainties. Section 4 presents solution strategies including the optimal strategy and four simplified strategies. In Section 5, the model is extended to tackle multiple train situations. In Section 6, numerical examples based on a real case study are provided to compare these strategies and illustrate their effectiveness in a range of scenarios. Finally, we draw conclusions and indicate further research in Section 7.

2. Literature review and an empirical example

This section contains two subsections Section 2.1 Literature review, and Section 2.2 Operations at rail terminal. Section 2.1 provides a summary and critical evaluation on the work in relation to the research topic. A critique of review is also demonstrated to
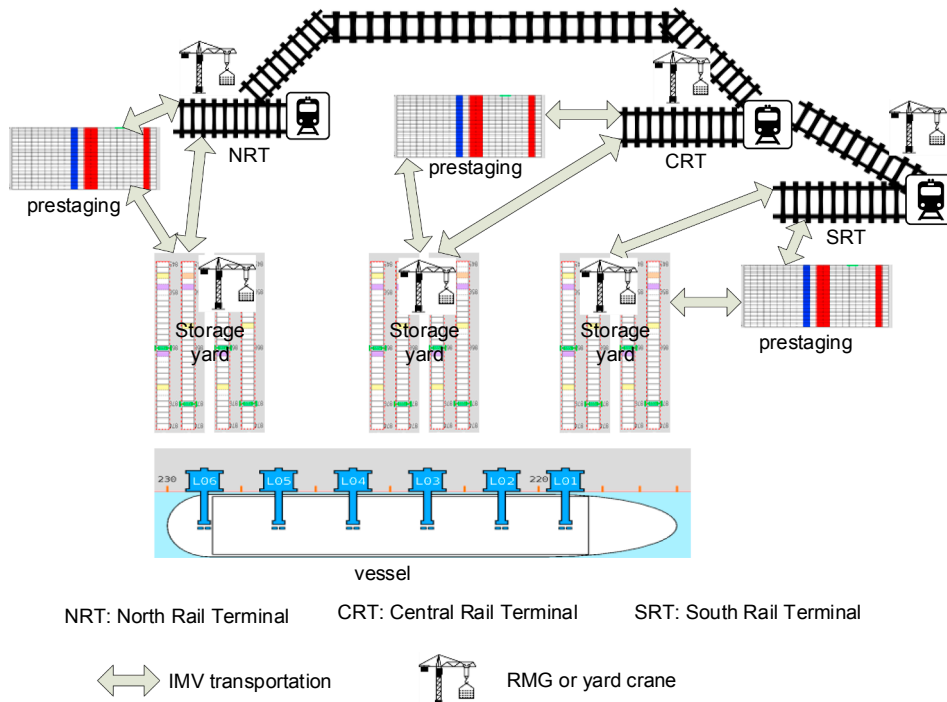

Fig. 1. An illustrative map of rail terminal operations at seaport.

establish need for this research. Section 2.2 provides a clear background of the problem investigated in this research, and also provides a detailed explanation of the operations of rail terminals in a seaport, facilitated by the illustrative map in Fig. 1, and empirical examples in Figs. 2–3.

2.1. Literature review

At a container rail terminal, there are two types of players that are closely related: the rail operator and the terminal operator. The **rail operator** is the operator of trains, who is responsible for transporting containers between seaport rail terminals and inland intermodal terminals via trains. In terms of the importing process, the rail operator will produce a load list specifying the specific containers to be loaded on a particular train. The load list can be communicated to the terminal operator between several hours and one day in advance of the train's arrival at the seaport rail terminal. However, the load list is subject to changes; it is not unusual that the rail operator may require loading of a certain number of containers with short notice. Failing to load such containers may incur a heavy penalty for the terminal operator. Fig. 1 depicts the operational activities among rail terminals in a seaport. The **terminal operator** is responsible for transporting containers between rail terminals and storage yards via internal moving vehicles (IMV) facilitated by other resources such as rail mounted gantry (RMG) cranes and yard cranes. As a seaport normally has a number of storage yards divided into blocks, the distance between storage yards and the rail terminal varies quite significantly. In the following, we provide a brief literature review from two aspects: rail operator and terminal operator.

From the rail operator's perspective, the predominant research focuses on the train load planning problem. Heggen et al. (2016) provided a comprehensive literature review on the models of train load planning problems. This problem mainly concerns determining the optimal assignment of containers to wagon slots in order to maximise train utilisation by taking into account various constraints imposed by train, track, wagons and load unit types. Early work in this research stream concentrated on rail operations at inland intermodal terminals (e.g. Corry and Kozan, 2008; Bruns and Knust, 2012). Ambrosino et al. (2011) is the first to extend the train load planning problem to the context of seaport rail terminals. They integrated the optimisation of train load planning with the optimisation of crane and storage planning. This essentially combines the operations from the rail operator's perspective (maximise train utilisation) with the operations from the terminal operator's perspective (minimise the unproductive operations in the terminal) to some extent. More specifically, Ambrosino et al. (2011) considered the train sequential loading problem at a seaport, which aims to assign containers to the wagon slots of the train by minimising the reshuffles in the storage area. The model was then extended by allowing backward empty crane moves (Ambrosino et al., 2013). Anghinolfi and Paolucci (2014) extended the train sequential loading model presented by Ambrosino et al. (2011) to consider a sequence of trains with different destinations with the aim to minimise the distances between the container locations in the storage area and the assigned wagons. Ambrosino and Siri (2015) examined the train load planning problem that aims to maximise the train utilisation and minimise unproductive operations in the rail terminal (e.g. unproductive movements of crane, reshuffle of containers in the stacking area). The problems in the above studies are generally formulated into mixed-integer programming models and often solved using heuristic or metaheuristic methods due to the computational complexity of the problems. Uncertainty is normally not considered in the research steam.

From the terminal operator's perspective, rail terminals are only part of the complex container terminal system. A large number of studies have been conducted to facilitate container terminal operators to make various strategic and tactical decisions, for example on berth allocation (Zhen, 2015), yard template planning (Zhen, 2014, Zhen et al., 2016), yard capacity, fleet composition, truck substitutability, and terminal scalability issues (Petering, 2011). There is also extensive amount of literature that deals with the quayside and yardside operations at container terminals (c.f. Steenken et al., 2004; Murty et al., 2005; Gunther and Kim, 2006; Stahlbock and Voß, 2008; Bierwirth and Meisel, 2010; Carlo et al., 2014; and Kim and Lee, 2015).

A range of uncertainties are also considered in some of the container terminal literature. For example, Zhen (2014) developed stochastic programming models for yard template planning under uncertain freight transportation demand, traffic congestion and vessel arrival cycle time patterns (Zhen et al., 2016). Zhen (2015) investigated the berth allocation problem related to uncertain dwell times of ships. A mixed integer programming model has been developed to address berth and quay crane scheduling problem simultaneously, with stochastic vessel arrival and container handling times (Han et al., 2010). A stochastic programming model has been proposed to dynamically allocate berths and quay-cranes in real stochastic environments characterised by stochastic vessel arrival and handling times (Zhou and Kang, 2008). However, these uncertainties are associated with the operations at quayside or yardside rather than at the rail terminal.

None of the aforementioned research has addressed the issues in relation to rail terminals at seaports. In this context, research can be classified as: (I) research that investigates the functionality or benefits brought by having a rail terminal at a seaport; and (II) studies that specifically focus on seaport rail terminal operations.

A few studies belong to category (I). Gillen and Hasheminia (2018) revealed that the best way to reduce dwell time in a Canadian seaport is to increase the frequency of train movements. They utilised a simulation model to measure the effect of changes on total dwell time, i.e. the time elapsed between the vessel berthing at the port and the loaded train leaving the port. Schonemann (2010) considered the transhipment process between ships and railways in German seaports and developed an information flow model which aimed to reduce buffer time by enabling better communication between the intermediaries of the transhipment process. However, the information flow model was hypothetical and remained as a conceptual design. Roso (2007), studied the concept of implementing a dry port in Scandinavia. The implementation of a dry port in a seaport's immediate hinterland was shown to increase the utilisation of train services, thus reducing the seaport's lorry congestion, leading to reduced environmental effects along the road network.

However, very few studies investigated the rail terminal operations in category (II). Caballini et al. (2014) presented an aggregate queue-based model representing the movements of containers in the maritime rail terminal area as a set of flows. A mixed-integer

linear mathematical programming problem is formulated in order to optimise the timings of the trains and the use of the handling resources devoted to rail port operations. The authors further extended the deterministic model to deal with unexpected situations by adopting an event-triggered receding-horizon planning approach, in which the deterministic optimisation problem will be re-solved whenever a critical event happens or the real values of some problem data significantly differ from the predicted ones. Caballini et al. (2016) extended the above model to address the capacity design and shunting operations planning problem in a port rail network. The above two papers introduced the concepts of flow rates and queue length to model the container movements/storage between yards, RMGs and the internal rail park. They did not consider the container prestaging issue or the regular uncertainties during container flows.

2.2. Operations at rail terminal- an empirical example

The two container flows via the rail terminal at a sea port (see Fig. 1) can be generally described as follows (see Fig. 2):

(1) Discharging flow (i.e. dashed-line): after a train carrying export containers arrives at a seaport rail terminal, the containers will be discharged from the train by RMGs onto the IMVs, which move them to storage yards to await loading onto vessels. The discharging activities are performed within a pre-specified discharge time window.
(2) Loading flow (i.e. solid lines): the loading process is performed within the load time window but after the discharge process. However, it is possible that the load time window partially overlaps with the discharge window. In the loading process, import containers are transported from storage yards via IMVs to the RMG, or import containers that have been prestaged in the Rail Terminal (RT) buffer are moved from there to the RMG via IMVs. Finally, the RMGs will load the containers onto the train.

The main job of a seaport rail terminal operator, is to discharge each of the daily inbound train arrivals (7 inbound trains arrive daily in the rail terminal considered in this research), shunt wagons from some inbound trains to form outbound trains, and load containers onto the outbound trains. However, resources, such as RMGs, IMVs and labour resource are limited and each train is assigned a fixed working time window. The terminal operator aims to allocate resources to loading and discharging tasks efficiently within the allowed time window for each train in order to minimise the total logistics costs. Any container that is not discharged or loaded in this time is called a miss. Misses are considered a huge customer service failure for an industrial port, damaging its reputation and potentially resulting in customer loss.

The existence of a variety of uncertain factors makes loading and discharging tasks even more challenging. Firstly, the train load plan could be changed for various reasons, e.g. uncertain events, urgent transport needs, inaccurate information. The terminal operator has to accommodate such later changes and respond in a dynamic way. Secondly, the train may arrive later than the planned arrival time, but it has to depart at the planned departure time. This affects the working time window and implies that the operational plan for discharge and load activities has to be adjusted accordingly. Thirdly, IMV often has to wait at the transfer points at storage yards to collect containers because yard cranes may be not available immediately. Normally, yard cranes are moving among different yard blocks and performing a range of tasks for both import and export containers. According to our empirical data from a real case, the IMV waiting time is uncertain and occupies a significant part of the total container transportation time. Fourthly, the IMV may have to wait at the rail terminal as well due to resource unavailability (e.g. RMGs). Fifthly, there may be uncertainty during the transportation between the rail terminal and storage yards as the distance is up to several kilometres and is subject to traffic congestion.

The fact that discharging happens before loading means that the chance of fully discharging is higher than fully loading. The process of taking the import container from the main storage yard to RMG or train is time consuming for the IMVs involved. As a way to mitigate risks imposed by various uncertainties, short distance local storage areas near RMG or train are used as buffers, to allow containers to be taken to local storage areas prior to the train arrival, so as to ensure service is maintained. The use of buffer is the prestaging measure defined in Introduction section.

Fig. 3 shows the actual container load rates at a rail terminal within a day (every 30 min) based on the empirical data of a real case. These empirical data demonstrate that there is a high level of uncertainty/variability in the process of moving containers between storage yards and rail terminals over the periods.

In current industrial practice, the rail terminal operator at the seaport relies on experience to manually design the operational plan in terms of assigning IMVs to perform the discharge and load tasks. Prestaging decisions are also made subjectively based on
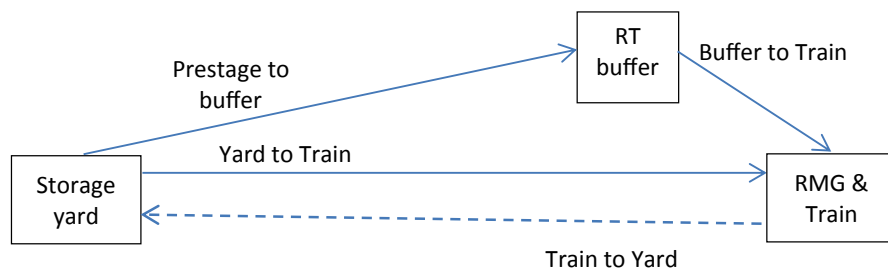

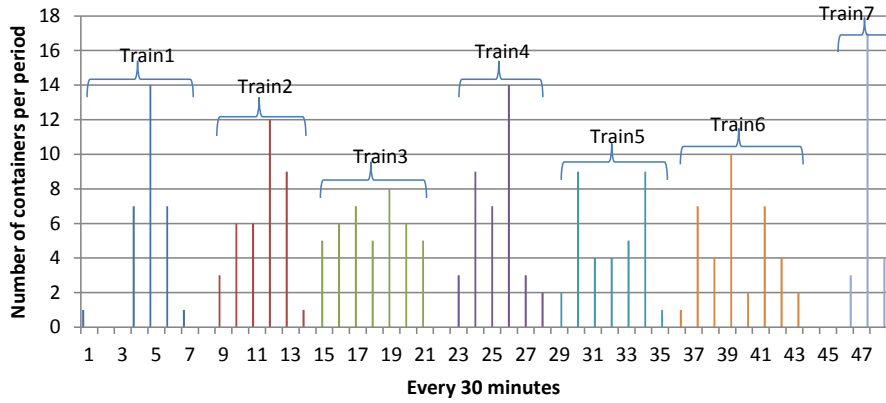**Fig. 2.** The processes of moving containers between yards and train.

Fig. 3. Empirical data of container loading rates at a rail terminal within a day.

experiences. Therefore, a need emerges for a tool that can make static decisions on prestaging and create dynamic plans for container discharging and loading over the time windows by taking into account the uncertainties in the processes.

The purpose of this study is to fill the above research gap by developing such a tool that can be applied to solve real train load/discharge planning problem, and to improve the efficiency of rail terminal operations at a container seaport. We adopt a discrete-time flow rate model. The dynamic plan of container discharging and loading activities is characterised by the discharge flow rates and load flow rates over the working time windows. The static prestaging decisions are represented by the pre-specified number of containers that are stored in the rail terminal buffer before the train arrives. We did not model the schedule of individual containers and resources mainly due to the existence of the aforementioned uncertain factors that may make the detailed schedule infeasible to implement. The flow-based discrete-time model can act as a tool to plan and re-plan the resource assignment in a dynamic way to accommodate the regular uncertainties and respond to the disruptive events. Relevant flow-based discrete-time models can be accessed in Caballini et al. (2014, 2016). However, we formulate the problem into a stochastic dynamic programming model that is able to determine the optimal static container prestaging decision and the optimal dynamic container flow rates simultaneously. More importantly, we take into account the uncertainties appearing in container handling activities, which have been largely ignored in the existing literature.

3. Stochastic dynamic programming model for the single train loading and discharging problem

The rail terminal operator needs to make a range of planning decisions, including: (i) Operational Level Static Planning: the number of containers to be prestaged from storage yards to the RT buffer before the working time window; (ii) Operational Level Dynamic Planning: the container flow rates from train to yard over the discharge time window and the container flow rates from yard to train and/or from RT buffer to train over the load time window. The problem is to seek the optimal decisions by minimising the total logistics cost associated with the prestaging, discharging and loading processes subject to a series of constraints.

3.1. Notation

In order to formulate the optimisation problem mathematically, we introduce the following notation.

Input parameters

t_d^b:	the beginning of the working time window for discharging the train ($t_d^b \geq 0$);
t_d^e:	the end of the working time window for discharging the train ($t_d^b < t_d^e$);
t_l^b:	the beginning of the working time window for loading the train ($t_l^b > t_d^b$);
t_l^e:	the end of the working time window for loading the train ($t_l^b < t_l^e$ and $t_d^e < t_l^e$);
Q_l^B:	the maximum number of containers allowed to be prestaged at the RT buffer;
Q_d^T:	the total number of containers initially on the train to be discharged;
Q_l^T:	the total number of containers planned to be loaded onto the train;
Q^T:	the capacity of the train with $Q^T \geq \max\{Q_d^T, Q_l^T\}$;
$C_{dl}(t)$:	the maximum discharging/loading capacity of RMG cranes in period t;
$U_d(t)$:	the maximum flow rate of transporting discharge containers in period t;
$U_l^Y(t)$:	the maximum flow rate of transporting load containers from yards in period t;
$U_l^B(t)$:	the maximum flow rate of transporting load containers from RT buffer in period t;

f_d^Y, f_l^Y, f_l^B: the parameters to represent uncertainty in container flow rates from train to yard, from yard to train, and from RT buffer to the train respectively;

c^P: the unit cost of prestaging containers at RT buffer (including transport and storage);

c^Y: the unit cost of IMV transporting a container between yard and RMG;

c^B: the unit cost of IMV transporting a container from RT buffer to RMG;

c^D: the unit cost of the RMG discharging a container from the train;

c^L: the unit cost of the RMG loading a container to the train;

c^S: the storage cost at RT buffer per container per period;

c^T: the storage cost on the train per container per period;

c^M: the unit penalty cost for missing discharge containers from the train by the end of discharging time window, or for missing load containers to the train by the end of loading time window;

Dynamic derived variables

$x_d^T(t)$ the number of discharge containers left on the train at the beginning of time period t;

$x_l^B(t)$ the number of containers left in the RT buffer to be loaded on the train at the beginning of time period t;

$x_l^T(t)$ the number of containers loaded on the train at the beginning of time period t;

$\xi_d^Y(t)$ the random variable representing the actual container flow rate from train to yard via RMG in period t;

$\xi_l^Y(t)$ the random variable representing the actual container flow rate from yard to train via RMG in period t;

$\xi_l^B(t)$ the random variable representing the actual container flow rate from RT buffer to train via RMG in period t;

Decision variables

$u_d^Y(t)$ the planned container flow rate from train via RMG to yard in period t;

$u_l^Y(t)$ the planned container flow rate from yard to train via RMG in period t;

$u_l^B(t)$ the planned container flow rate from RT buffer to train via RMG in period t;

q: the number of containers prestaged at RT buffer to be loaded on train initially, which is not greater than Q_l^B;

The rationale of introducing a range of unit costs is explained below:

(1) c^P: the unit cost of prestaging: this is a one-off cost incurred to transport containers from storage yard and store them at RT buffer, and it associates with the prestaged number of containers q.

(2) c^Y: this is the unit cost of IMV transporting one container between yard and RMG, but it is worth noting that IMVs together with drivers, as resources, are deployed ahead of the actual transportation. Therefore, this unit cost is linked with the committed resources, i.e., the planned number of IMVs used to transport containers between yard and RMG, which are measured as planned flow rates $u_d^Y(t)$ and $u_l^Y(t)$.

(3) c^B: this is the unit cost of IMV transporting one container from RT buffer to RMG. For similar reasons to those outlined in point 2) above, this unit cost is associated with the deployed number of IMVs between RT buffer and RMG, measured as $u_l^B(t)$.

(4) c^D and c^L, unit costs of RMG discharging and loading containers: although the planned number of containers to be loaded or charged per train is fixed in our context, the actual number of containers that will be loaded or discharged by RMGs is uncertain due to the uncertainty in the processes. It is not unusual that trains depart from the seaport before completing loading in order to meet the rail network's timetable. Missed loading of containers can incur high penalties and is an important issue for the rail terminal operator to deal with. On the other hand, it is possible that containers on the train may not be completely discharged within the pre-specified discharge time window due to the uncertainty in the discharging process, which incurs additional cost. For these reasons, c^D and c^L are associated with actual loading and discharging activities, measured as $\xi_d^Y(t)$, $\xi_l^Y(t)$ and $\xi_l^B(t)$, which are actual flow rates delivered by the resources available within the working time window.

(5) c^S, unit storage cost at RT buffer: buffer size in a rail terminal is limited and is regarded as a precious resource. A buffer can be shared by a number of trains to store containers. To increase the utilisation of buffer, it is in the terminal operator's interest to free up the buffer as soon as possible and offer it to other trains.

(6) c^T, unit storage cost on the train per container per period: similar to the storage cost adopted in an inventory control problem, the unit storage cost on train is used to model the following practices:

   (a) In a stochastic case, it encourages the operator to discharge containers from the train as soon as possible to avoid storage cost, and possible misses caused by uncertainties;

   (b) In a deterministic situation, it is preferred to load containers onto trains as late as possible to allow customers to make late changes. Our industrial partners advise that customers may occasionally request an urgent change in loading containers.

(7) Unit storage cost in yard is not considered in this research for the following reasons: at the rail terminal the storage yard size is much bigger than the buffer size; therefore, in comparison with other unit costs, the value of unit storage cost at storage yard is small enough to be neglectable. The omission of this cost has no significant impact on the decisions of prestaging, discharging and loading flow rates.

(8) It should be noted that due to the uncertainties involved in the transportation and operation processes at the rail terminal, for example, waiting time at storage yards, waiting time at rail terminals, late arrival of trains, RMG breakdown, etc., the committed

resources may not be fully effective; therefore the planned flow rates may not be fully realised in the actual flow rates. This means it is reasonable to assume that actual flow rates are lower than the planned flow rates, i.e., $\xi_d^Y(t) \leq u_d^Y(t)$, $\xi_l^Y(t) \leq u_l^Y(t)$, and $\xi_l^B(t) \leq u_l^B(t)$.

3.2. Dynamic model

We assume that the actual container flow rates follow uniform probability distributions in intervals with their ranges proportional to the planned flow rates.

In the time periods of $t_d^b \leq t \leq t_l^e$, the discrete-time dynamic equations of container flows (including both discharge and load processes) can be given as follows:

$$x_d^T(t_d^b) = Q_d^T \tag{1}$$

$$x_l^B(t_d^b) = q \tag{2}$$

$$x_l^T(t_d^b) = 0 \tag{3}$$

$$x_d^T(t+1) = x_d^T(t) - \xi_d^Y(t) \tag{4}$$

$$x_l^B(t+1) = x_l^B(t) - \xi_l^B(t) \tag{5}$$

$$x_l^T(t+1) = x_l^T(t) + \xi_l^B(t) + \xi_l^Y(t) \tag{6}$$

$$0 \leq q \leq Q_l^B \tag{7}$$

$$0 \leq u_d^Y(t) \leq \min\{x_d^T(t), C_{dl}(t), U_d(t)\} \cdot I\{t_d^b \leq t \leq t_d^e\} \tag{8}$$

$$0 \leq u_l^B(t) \leq \min\{x_l^B(t), C_{dl}(t), U_l^B(t)\} \cdot I\{t_l^b \leq t \leq t_l^e\} \tag{9}$$

$$0 \leq u_l^Y(t) \leq \min\{C_{dl}(t), U_l^Y(t)\} \cdot I\{t_l^b \leq t \leq t_l^e\} \tag{10}$$

$$u_l^Y(t) + u_l^B(t) \leq \min\{Q_l^T - x_l^T(t), Q^T - x_d^T(t) - x_l^T(t)\} \tag{11}$$

$$u_d^Y(t) + u_l^Y(t) + u_l^B(t) \leq C_{dl}(t) \tag{12}$$

$$\xi_d^Y(t) \sim Uniform(f_d^Y \cdot u_d^Y(t), u_d^Y(t)) \tag{13}$$

$$\xi_l^B(t) \sim Uniform(f_l^B \cdot u_l^B(t), u_l^B(t)) \tag{14}$$

$$\xi_l^Y(t) \sim Uniform(f_l^Y \cdot u_l^Y(t), u_l^Y(t)) \tag{15}$$

The problem is to find the optimal control policy $\{q, u_d^Y(t), u_l^B(t), u_l^Y(t): t_d^b \leq t \leq t_l^e\}$ by minimizing the following objective function:

$$\begin{aligned}
J^* = \min E\,[&c^P \cdot q + c^M \cdot x_d^T(t_d^e + 1) + c^M \cdot (Q_l^T - x_l^T(t_l^e + 1)) + \\
&\sum_t (c^S \cdot x_l^B(t+1) + c^T \cdot (x_d^T(t+1) + x_l^T(t+1)) + c^Y \cdot (u_d^Y(t) + u_l^Y(t)) \\
&+ c^B \cdot u_l^B(t) + c^D \cdot \xi_d^Y(t) + c^L \cdot (\xi_l^B(t) + \xi_l^Y(t)))]
\end{aligned} \tag{16}$$

On the right-hand-side of the above equation, the first term is the container prestaging cost; the second term represents the missed discharge penalty at the end of the discharging time window; the third term represents the missed load penalty at the end of loading time window; the fourth term aggregates the cost for each period that includes the storage costs at the rail terminal buffer, the storage cost for containers on the train, the IMV deployment cost between the yard/buffer, and the handling cost of RMGs. The explanations to the IMV deployment cost and the RMG handling cost can be referred to Section 3.1 points 2) ~ 4).

Note that in the problem (16) the prestaging decision q is a static decision variable, whereas planned flow rates decisions $\{u_d^Y(t), u_l^B(t), u_l^Y(t)\}$ are dynamic decision variables over multiple periods. Specifically, the static prestaging decision can be reflected by appropriately setting the initial value of the state variable $x_l^B(t)$ at $t = t_d^b$.

In practice, we are interested in finding the optimal state-feedback control policies for the container flow rates. Hence, $\{u_d^Y(t), u_l^B(t), u_l^Y(t)\}$ can be interpreted as $\{u_d^Y(x_d^T(t), x_l^B(t), x_l^T(t)), u_l^B(x_d^T(t), x_l^B(t), x_l^T(t)), u_l^Y(x_d^T(t), x_l^B(t), x_l^T(t))\}$. The optimal dynamic container flow rates can be found by applying the stochastic dynamic programming approach. Let $J_t(x_d^T(t), x_l^B(t), x_l^T(t))$ denote the cost-to-go function from period t onward. The dynamic programming optimality equation is given by ($t_d^b \leq t \leq t_l^e$):

$$
\begin{aligned}
J_t(x_d^T(t), x_l^B(t), x_l^T(t)) &= \min E\{c^P \cdot q \cdot I\{t = t_d^b\} + c^M \cdot x_d^T(t) \cdot I\{t = t_d^e + 1\} \\
&\quad + c^M \cdot (Q_l^T - x_l^T(t)) \cdot I\{t = t_l^e + 1\} \\
&\quad + c^S \cdot x_l^B(t) \cdot I\{t > t_d^b\} + c^T \cdot x_d^T(t) \cdot I\{t_d^b < t \leqslant t_d^e\} \\
&\quad + c^T \cdot x_l^T(t) \cdot I\{t_l^b < t \leqslant t_l^e\} + c^Y \cdot (u_d^Y(t) + u_l^Y(t)) \\
&\quad + c^B \cdot u_l^B(t) + c^D \cdot \xi_d^Y(t) + c^L \cdot (\xi_l^B(t) + \xi_l^Y(t)) \\
&\quad + J_{t+1}(x_d^T(t+1), x_l^B(t+1), x_l^T(t+1))\} \\
&= c^P \cdot q \cdot I\{t = t_d^b\} + c^M \cdot x_d^T(t) \cdot I\{t = t_d^e + 1\} \\
&\quad + c^M \cdot (Q_l^T - x_l^T(t)) \cdot I\{t = t_l^e + 1\} + c^S \cdot x_l^B(t) \cdot I\{t > t_d^b\} \\
&\quad + c^T \cdot x_d^T(t) \cdot I\{t_d^b < t \leqslant t_d^e\} + c^T \cdot x_l^T(t) \cdot I\{t_l^b < t \leqslant t_l^e\} \\
&\quad + \min \{c^Y \cdot (u_d^Y(t) + u_l^Y(t)) + c^B \cdot u_l^B(t) \\
&\quad + E[c^D \cdot \xi_d^Y(t) + c^L \cdot (\xi_l^B(t) + \xi_l^Y(t)) \\
&\quad + J_{t+1}(x_d^T(t+1), x_l^B(t+1), x_l^T(t+1))]\}
\end{aligned}
\tag{17}
$$

where $J_{t+1}(x_d^T(t+1), x_l^B(t+1), x_l^T(t+1)) \equiv 0$ for $t = t_l^e$. Here $I\{\text{condition}\}$ is an indicator function. It takes 1 if the condition in {} is true, 0 otherwise.

Note that the minimum operation on the right-hand-side of (17) is taken over the planned flow rates $\{u_d^Y(t), u_l^B(t), u_l^Y(t)\}$; the actual flow rates $\{\xi_d^Y(t), \xi_l^B(t), \xi_l^Y(t)\}$ are random variables that depend on $\{u_d^Y(t), u_l^B(t), u_l^Y(t)\}$; the cost-to-go function $J_{t+1}(x_d^T(t+1), x_l^B(t+1), x_l^T(t+1))$ depends on both $\{u_d^Y(t), u_l^B(t), u_l^Y(t)\}$ and $\{\xi_d^Y(t), \xi_l^B(t), \xi_l^Y(t)\}$ from (4)–(6). At period t, the mathematical expectation is only performed on $c^D \cdot \xi_d^Y(t) + c^L \cdot (\xi_l^B(t) + \xi_l^Y(t)) + J_{t+1}(x_d^T(t+1), x_l^B(t+1), x_l^T(t+1))$ as shown in the second equation of (17). This is because other terms are deterministic at the current period. The calculation of the expectation in the last term on the right-hand-side of (17) can be referred to similar expositions in Bertsekas (1976), Puterman (2005), and Song (2013). For example, under the assumption that the random variables are discrete, let $p_{d,i}{}^Y := Prob\{\xi_d^Y(t) = i\}$; $p_{l,j}{}^Y := Prob\{\xi_l^Y(t) = j\}$; $p_{l,k}{}^B := Prob\{\xi_l^B(t) = k\}$; then we have,

$$
\begin{aligned}
&E[c^D \cdot \xi_d^Y(t) + c^L \cdot (\xi_l^B(t) + \xi_l^Y(t)) + J_{t+1}(x_d^T(t+1), x_l^B(t+1), x_l^T(t+1))] \\
&= \sum_i \sum_j \sum_k p_{d,i}{}^Y \cdot p_{l,j}{}^Y \cdot p_{l,k}{}^B \cdot [c^D \cdot i + c^L \cdot (k+j) + J_{t+1}(x_d^T(t) - i, x_l^B(t) - k, x_l^T(t) + k + j)]
\end{aligned}
$$

From the dynamic programming optimality equation (17), the optimal expected cost defined in (16) is given by $J^* = J_t(Q_d^T, q^*, 0)$ with $t = t_d^b$, where $q^* = \operatorname{argmin}\{J_t(Q_d^T, q, 0) \mid 0 \leqslant q \leqslant Q_l^B, t = t_d^b\}$.

We further analysed the properties of the dynamic model in (1)–(17) to gain an insight into potential decision making on pre-staging and flow rates. The dynamic optimality equation presented in (17) presents a linear relationship between the unit costs and planned flow rate, and a linear relationship between unit costs and derived dynamic variables, such as actual flow rates. The ordinal relationships of the unit cost coefficients (c.f. the explanations in Section 6.1) implies that the dynamic model has several nice properties:

(I) Due to the extremely high penalty of missing containers for discharging and loading, the priority of the decision making should avoid the misses.

(II) Intuitively, the prestaging decision variable q should increase if the train's working time window is narrower and/or the degree of uncertainties between RMGs and yards is higher.

(III) In terms of the flow rate decision variables, the discharging and loading operations should be done as rapidly as possible to avoid misses and penalty costs. However, there exists a trade-off between the penalties and the storage and the prestaging costs. On the other hand, if we can guarantee that all containers will be loaded before the train departure time (e.g. in the deterministic case), then the loading operations should be performed as late as possible due to the storage cost on the train.

(IV) In the uncertain case where there is no guarantee of completing discharge before the train's departure, and if c^M is high enough, the maximum resource capacity should be deployed to discharge containers as much and as early as possible.

(V) There are two factors to consider when loading starts, the timing and the origin of loading. Timing affects the storage cost in the train and in the buffer, while the origin of loading decides the loading sequence, i.e., from yard to train first, or from buffer to train first.

In Section 6 Numerical experiments, a few observations are made to verify the properties above, when parameter values are given.

4. Optimal strategies and solutions

The traditional way to resolve the stochastic dynamic programming problem in (1)-(17) is the backwards value iteration algorithm (c.f. Bertsekas, 1976; Sennott, 1999; Puterman, 2005; Song, 2013), which will be named as **Optimal Strategy** in this research, and explained in detail in Section 4.1.

However, the main disadvantage of seeking the optimal strategy is computational complexity, which is due to the curse of dimensionality and is a common issue for traditional dynamic programming formulated problems. Therefore, four alternative solution strategies are proposed, with the purpose to reduce the curse of dimensionality by simplifying the solution space. In this way the problem can be solved much more efficiently, at the expense of a solution with reduced quality. The proposed four alternative

solution strategies include: a **Decoupled Strategy** where loading and discharging are decoupled; two greedy strategies i.e., **Buffer-first Greedy strategy** and **Yard-first Greedy strategy**; and a **Bang-bang strategy** loading/discharging the whole batch or nothing.

The above five solution strategies are explained in detail in the following sub-sections including the rationales of their adoption in our research context. We will also provide theoretical analysis to compare the computational complexity among five strategies at the end of this section. More comparison among these strategies in terms of effectiveness in achieving optimality and computing efficiency are made in Section 6 using numerical examples.

4.1. Optimal strategy

The stochastic dynamic programming problem in (1)–(17) can be solved using the backwards value iteration algorithm.

Step 1: Let $J_{t+1}(x_d^T(t+1), x_l^B(t+1), x_l^T(t+1)) \equiv 0$ for any state $(x_d^T(t+1), x_l^B(t+1), x_l^T(t+1))$ at $t = t_l^e$.

Step 2: Let $t = t_l^e$.

Step 3: Use (17) to calculate the optimal value function $J_t(x_d^T(t), x_l^B(t), x_l^T(t))$, and the optimal state-feedback control $\{u_d^Y(x_d^T(t), x_l^B(t), x_l^T(t)), u_l^B(x_d^T(t), x_l^B(t), x_l^T(t)), u_l^Y(x_d^T(t), x_l^B(t), x_l^T(t))\}$ subject to constraints in (1)~(15).

Step 4: Let $t = t-1$. If $t \geqslant t_d^b$, go to Step 3.

Step 5: At $t = t_d^b$, find the optimal prestaging decision by:

$$q^* = \text{argmin}\{J_t(Q_d^T, q, 0)|\ 0 \leqslant q \leqslant Q_l^B, t = t_d^b\}$$

Step 6: Return the optimal cost given by $J^* = J_t(Q_d^T, q^*, 0)$ with $t = t_d^b$; the optimal decision variables $q^*, u_d^Y(x_d^T(t), x_l^B(t), x_l^T(t))$, $u_l^B(x_d^T(t), x_l^B(t), x_l^T(t)), u_l^Y(x_d^T(t), x_l^B(t), x_l^T(t))$ for $t_d^b \leqslant t \leqslant t_l^e$.

The advantages of the above algorithm include the ability to obtain the optimal expected cost, J^*, the optimal prestaging decision, q^*, and the optimal dynamic flow rate plan over the working time window, $u_d^Y(x_d^T(t), x_l^B(t), x_l^T(t)), u_l^B(x_d^T(t), x_l^B(t), x_l^T(t))$, $u_l^Y(x_d^T(t), x_l^B(t), x_l^T(t))$ for $t_d^b \leqslant t \leqslant t_l^e$. In addition, although we have assumed that the uncertainties are characterised by uniform distributions, i.e. $\xi_d^Y(t), \xi_l^B(t), \xi_l^Y(t)$ follow uniform distributions, the model and the solution can be similarly applied for other types of probability distributions.

The main disadvantage of the above algorithm is the computational complexity, which is caused by the curse of dimensionality. It is a common issue for most dynamic programming formulated problems. Stochastic dynamic programming has up to three types of curses of dimensionality: the state space, the outcome space and the action space (Powell, 2011). In our problem setting, we have three dimensions of state space, i.e. $(x_d^T(t), x_l^B(t), x_l^T(t))$; three dimensions of outcome space, i.e. $(\xi_d^Y(t), \xi_l^B(t), \xi_l^Y(t))$; and three dimensions of action space, i.e. $(u_d^Y(t), u_l^B(t), u_l^Y(t))$. This gives rise to 9 nested loops at each period in the value iteration algorithm. Note that each dimension takes a range of values. It requires a substantial computational effort to obtain the optimal policy, although it is not intractable. Powell (2011) presented an approximate dynamic programming approach to deal with the curse of dimensionality issue, which is effective for high-dimensional optimisation problems. As our problem is not a very high dimensional problem, we present alternative methods to simplify the computational complexity by focusing on reducing the outcome space and the action space.

4.2. Two greedy strategies

A traditional greedy algorithm is an algorithmic strategy that makes the best optimal choice at each small stage with the aim to find a globally optimum solution. It picks the best immediate solution without considering the big picture, hence it is called greedy. The greedy algorithm or rule has been widely adopted in local search heuristics (Taha, 2011), e.g. the well-known hill climbing algorithm. The philosophy of a greedy algorithm is to achieve the best outcome using the immediate resources at each stage.

In our research context, since failing to complete discharging tasks or loading tasks would lead to extremely high penalty, and inspired by philosophy of the traditional greedy algorithm, we introduce two new greedy strategies in this work, to deploy the maximum resources to perform discharging and loading as much and as early as possible.

The advantage of such greedy strategy is the substantial reduction of the action space and the outcome space, which then leads to the reduced solution space. We introduce two greedy strategies by defining the sequence of loading activities (from yard and buffer). The first greedy strategy, called **Buffer-first Greedy strategy**, discharges containers from trains, loads containers from the buffer first and then loads containers from the yard. The second greedy strategy is **Yard-first Greedy strategy**, which discharges containers from trains, and loads containers from the yard as early and as many as possible. It should be noted that the second greedy strategy does not involve prestaging activity.

Mathematically, **Buffer-first Greedy strategy** can be expressed as below in terms of $(u_d^Y(t), u_l^B(t), u_l^Y(t))$:

$$u_d^Y(t) = \min\{x_d^T(t), C_{dl}(t), U_d(t)\}\cdot I\{t_d^b \leqslant t \leqslant t_d^e\} \tag{18}$$

$$u_l^B(t) = \min\{x_l^B(t), Q_l^T - x_l^T(t), Q^T - x_d^T(t) - x_l^T(t), C_{dl}(t) - u_d^Y(t), U_l^B(t)\}\cdot I\{t_l^b \leqslant t \leqslant t_l^e\} \tag{19}$$

$$u_l^Y(t) = \min\{Q_l^T - x_l^T(t) - u_l^B(t), Q^T - x_d^T(t) - x_l^T(t) - u_l^B(t), C_{dl}(t) - u_d^Y(t) - u_l^B(t), U_l^Y(t)\}\cdot I\{t_l^b \leqslant t \leqslant t_l^e\} \tag{20}$$

In mathematical form, **Yard-first Greedy strategy** can be expressed in the following equations:

$$u_d^Y(t) = \min\{x_d^T(t), C_{dl}(t), U_d(t)\}\cdot I\{t_d^b \leqslant t \leqslant t_d^e\} \tag{21}$$

$$u_l^Y(t) = \min\{Q_l^T - x_l^T(t), Q^T - x_d^T(t) - x_l^T(t), C_{dl}(t) - u_d^Y(t), U_l^Y(t)\} \cdot I\{t_l^b \leqslant t \leqslant t_l^e\} \tag{22}$$

$$u_l^B(t) = 0 \tag{23}$$

As shown in formulas (18) and (21), the discharging quantity performed by the two Greedy strategies is the maximum flow rate $u_d^Y(t)$, which is constrained by the number of discharge containers left on the train $x_d^T(t)$, the maximum discharging capacity of RMG crane $C_{dl}(t)$, and the maximum rate of transporting discharge containers $U_d(t)$.

The Buffer-first Greedy strategy first loads the maximum possible number of containers from the buffer, as expressed in (19), and does not start loading from the yard until all the containers in the buffer are loaded (see (20)). As expressed in (19), the maximum flow rate from the buffer, $u_l^B(t)$, is associated with a few parameters, i.e., the number of containers left in the buffer $x_l^B(t)$, the total containers left to be loaded $Q_l^T - x_l^T(t)$, the capacity left on the train $Q^T - x_d^T(t) - x_l^T(t)$, RMG crane's available capacity $C_{dl}(t) - u_d^Y(t)$ and the transportation capacity $U_l^B(t)$. Likewise the maximum flow rate from the yard, $u_l^Y(t)$, is also linked with these parameters, which are further reduced by the flow rate from the buffer $u_l^B(t)$, as presented in (20).

Similarly, the Yard-first Greedy strategy first loads the maximum possible number of containers from the yard, as shown in (22), but does not start loading from buffer at all (see (23)). The maximum flow rate of loading from the yard, $u_l^Y(t)$, is similar to the rate in the Buffer-first Greedy strategy but without the effect of $u_l^B(t)$.

4.3. Decoupled strategy

According to the practice in our industrial partner, discharging and loading processes are often separated in the rail operations at the port, even though their working time windows may overlap. One of the reasons of doing so is wagon shunting at the end of discharging and forming new departure trains. One way to simplify the problem is to decouple the discharging decisions and the loading decisions. This would mitigate two types of curse of dimensionality, i.e. reducing the outcome space and the action space. In our context, the three dimensions of the outcome space $(\xi_d^Y(t), \xi_l^B(t), \xi_l^Y(t))$ can essentially be reduced to be two dimensions, as can the action space $(u_d^Y(t), u_l^B(t), u_l^Y(t))$.

Decoupling may be done by changing the pre-specified working time windows to avoid the overlapping. However, such treatment may significantly reduce the rail terminal handling capacity. Another way is to keep the working time windows unchanged, but impose additional constraints to only allow one way flow at each period during the overlapping interval, and only allow loading action to start after all export containers have been discharged during the overlapping interval.

Mathematically, the additional constraints of the control actions at each period $(u_d^Y(t), u_l^B(t), u_l^Y(t))$ are as follows:

$$u_l^B(t) \cdot I\{x_d^T(t) > 0\} = 0, \text{ for } t_l^b \leqslant t \leqslant t_d^e \tag{24}$$

$$u_l^Y(t) \cdot I\{x_d^T(t) > 0\} = 0, \text{ for } t_l^b \leqslant t \leqslant t_d^e \tag{25}$$

It should be noted that if $t_l^b \leqslant t_d^e$, then the actual start time of the loading process is flexible because the discharge process may be completed earlier or later due to the uncertainties. On the other hand, if $t_l^b > t_d^e$, then the discharge and load processes have already been decoupled and (24) and (25) do not exist. By adding constraints (24) and (25) to (1)–(16), the value iteration algorithm in Section 4.1 can be similarly applied to find the optimal decoupled strategy among all decoupled control policies.

4.4. Bang-bang strategy

Based on the literature on stochastic dynamic programming, we observe that many optimal control policies have good structural properties such as monotonicity (Sennott, 1999; Puterman, 2005; Song, 2013). For example, the optimal action at a given state can infer the optimal action at some neighbouring states.

Bang-bang control restricts the control action to be at either the lower or the upper boundary, but never between the boundaries (Evans, 2010). Bang-bang control has been used in inventory management (Song, 2005; Tsoularis, 2014), reverse logistics (Dobos and Richter, 2004, 2006) and manufacturing systems (Song et al., 1998; Salama, 2000), in the form of a pure bang-bang strategy (Richter, 1997; Song et al., 1998; Dobos and Richter, 2004, 2006; Song, 2005; Tsoularis, 2014), or as the well-known (s, S) policy (Veinott, 1966; Zheng and Federgruen, 1991; Salama, 2000). The (s, S) policy states that an order is placed to increase the inventory position to the upper level S as soon as this inventory position drops below the lower level s, which essentially regulates the system by either committing sufficient effort to bring the inventory position to an upper boundary or doing nothing.

The unique feature of bang-bang type of control policies is restricting the control action into two choices for each decision variable. This could greatly reduce the action space and simplify the optimization of the decision variables. Applying the concept of bang-bang control to our research context, we propose a bang-bang strategy, which performs discharging and loading at either the maximum possible flow rate or the minimum flow rate (i.e. zero). Specifically, the constraints of the control actions at each period $(u_d^Y(t), u_l^B(t), u_l^Y(t))$ can be narrowed down as follows:

$$u_d^Y(t) \in \{0, \min\{x_d^T(t), C_{dl}(t), U_d(t)\} \cdot I\{t_d^b \leqslant t \leqslant t_d^e\}\} \tag{26}$$

$$u_l^B(t) \in \{0, \min\{x_l^B(t), Q_l^T - x_l^T(t), Q^T - x_d^T(t) - x_l^T(t), C_{dl}(t) - u_d^Y(t), U_l^B(t)\} \cdot I\{t_l^b \leqslant t \leqslant t_l^e\}\} \tag{27}$$

$$u_l^Y(t) \in \{0, \min\{Q_l^T - x_l^T(t) - u_l^B(t), Q^T - x_d^T(t) - x_l^T(t) - u_l^B(t), C_{dl}(t) - u_d^Y(t) - u_l^B(t), U_l^Y(t)\} \cdot I\{t_l^b \leqslant t \leqslant t_l^e\}\} \tag{28}$$

Comparing formulas (26)–(28) with (18)–(20) and (21)–(23) respectively, it is clear that the two greedy strategies, the Buffer-first strategy and the Yard-first strategy are special forms of Bang-bang strategy. When loading from the buffer, the Bang-bang strategy either loads no containers, or the maximum flow rate, which is identical to the flow rate performed by the Buffer-first strategy (see (19) and (27)). Likewise, when loading from the yard, the maximum flow rate conducted by the Bang-bang strategy equals the flow rate of the Yard-first strategy (see (22) and (28)). In our context, Bang-bang strategy simplifies the action space, which can also reduce the outcome space, because the outcome space $(\xi_d^Y(t), \xi_l^B(t), \xi_l^Y(t))$ depends on the action space $(u_d^Y(t), u_l^B(t), u_l^Y(t))$.

By replacing the constraints (8)–(10) with (26)–(28), the value iteration algorithm in Section 4.1 can be applied to find the optimal Bang-bang strategy among all Bang-bang types of control policies.

4.5. Comparison of computational time complexity among five strategies

The above five strategies can be obtained and evaluated using the valuate iteration algorithm described in Section 4.1 by appropriately adjusting the constraints of the decision variables. It is helpful to analyse and compare the computational time complexities of these strategies. Note that the value iteration algorithm for each strategy needs to be performed in $t_l^e - t_d^b + 1$ steps to cover both discharging and loading windows. At each step, the objective function will be evaluated $Q_d^T * Q_l^B * Q_l^T$ times, which corresponds to the size of the system state space for $(x_d^T(t), x_l^B(t), x_l^T(t))$. This applies to all five strategies.

Aligned with the discharging window (t_d^b, t_d^e) and the loading window (t_l^b, t_l^e), the time steps over which the optimality equation in (17) is solved can be divided into four possible periods: discharging only period, discharging and loading period, no discharging and no loading period, and loading only period.

In each period of the four possible periods, the minimization in Eq. (17) should be performed over all the possible combinations of control actions $(u_d^Y(t), u_l^B(t), u_l^Y(t))$. As such, a new metric measuring the Computational Time Complexity (CTC) of solution strategies is introduced and defined as the number of mathematical expectation calculations required in order to perform the "min" operation in Eq. (17) at each system state. The CTC metric actually represents the size of the admissible control actions at each system state under a given solution strategy.

We take the discharging and loading period as an example to provide a detailed analysis below to compare the CTC among five strategies.

Firstly, let us consider the Optimal strategy algorithm. In general, $u_d^Y(t)$ can take a value from 0 to $U_d(t)$, $u_l^B(t)$ can take a value from 0 to $U_l^B(t)$, and $u_l^Y(t)$ can take a value from 0 to $U_l^Y(t)$ (note that the maximum possible values of $u_d^Y(t)$, $u_l^B(t)$, $u_l^Y(t)$ may be constrained by other resources, e.g. RMG crane capacity). Therefore, it may require up to $U_d(t) * U_l^B(t) * U_l^Y(t)$ calculation of the mathematical expectation in (17), i.e., CTC in discharging and loading period equals to $U_d(t) * U_l^B(t) * U_l^Y(t)$, as shown in Table 1. It should be pointed out that the real computation time for the mathematical expectation will depend on the values of $u_d^Y(t)$, $u_l^B(t)$, $u_l^Y(t)$ and the uncertainty factors f_d^Y, f_l^B, f_l^Y.

Secondly, we consider the Decoupled strategy. This discharging and loading period can be further divided into two sub-periods according to $x_d^T(t) > 0$ or $x_d^T(t) = 0$. To decouple the discharging and loading activities, we only perform discharging activity when $x_d^T(t) > 0$; and only perform loading activity when $x_d^T(t) = 0$ (i.e. the discharging process has completed). During the sub-period with $x_d^T(t) > 0$, $u_d^Y(t)$ can take a value from 0 to $U_d(t)$, and $u_l^B(t) = u_l^Y(t) = 0$. Therefore, it may require up to $U_d(t)$ calculations of the mathematical expectation in (17), and CTC = $U_d(t)$. On the other hand, during the sub-period with $x_d^T(t) = 0$, $u_l^B(t)$ can take a value from 0 to $U_l^B(t)$, $u_l^Y(t)$ can take a value from 0 to $U_l^Y(t)$, and $u_d^Y(t) = 0$. This implies that up to $U_l^B(t) * U_l^Y(t)$ calculations of the mathematical expectation in (17) are required, i.e., CTC = $U_l^B(t) * U_l^Y(t)$, as presented in Table 1.

Thirdly, we consider the two Greedy strategies: the Buffer-first Greedy strategy and the Yard-first Greedy strategy. From (19)~(20) and (22)~(23), it can be seen that the control actions $(u_d^Y(t), u_l^B(t), u_l^Y(t))$ are implicit, and no "min" operation in (17) is required. In other words, at each time step for each system state, the calculation of the mathematical expectation in Eq. (17) is only performed once, which means CTC = 1.

Fourthly, let us consider the Bang-bang strategy. From (27) and (28), it can be seen that $u_d^Y(t)$, $u_l^B(t)$ and $u_l^Y(t)$ all take at most two values. Therefore, it requires CTC = 2 * 2 * 2 = 8 calculations of the mathematical expectation in Eq. (17).

Table 1 summarizes the comparison of CTC among five strategies. It shows the required CTC for each system state at each time step. Clearly, the two Greedy strategies are the fastest algorithms, followed by the Bang-bang strategy, the Decoupled strategy and the

Table 1
A comparison of Computational Time Complexity (CTC) under five strategies.

Time step t	CTC of Optimal	CTC of Decoupled	CTC of Buffer-first or Yard-first	CTC of Bang-bang
Discharging only period $(t_d^b \leqslant t < t_l^b)$	$U_d(t)$	$U_d(t)$	1	2
Discharging and loading period $(t_l^b \leqslant t \leqslant t_d^e)$ with $x_d^T(t) > 0$	$U_d(t) * U_l^B(t) * U_l^Y(t)$	$U_d(t)$	1	8
Discharging and loading period $(t_l^b \leqslant t \leqslant t_d^e)$ with $x_d^T(t) = 0$	$U_l^B(t) * U_l^Y(t)$	$U_l^B(t) * U_l^Y(t)$	1	4
No discharging and no loading period $(t_d^e < t < t_l^b)$	1	1	1	1
Loading only period $(\max\{t_l^b, t_d^e\} < t \leqslant t_l^e)$	$U_l^B(t) * U_l^Y(t)$	$U_l^B(t) * U_l^Y(t)$	1	4

Optimal strategy. In particular, the two Greedy strategies and the Bang-bang strategy could significantly outperform the Optimal strategy in terms of time complexity. The numerical examples in Section 6 will confirm this assertion.

5. Extension to multiple trains loading and discharging problem

In the above sections, we assumed that there is only one train to be discharged and loaded within the working windows. In reality, it is possible that multiple trains are discharged (or loaded) at the same time. In this section, we generalise the model to deal with the N train situation. Suppose the handling capacities $\{C_{dl}(t), U_d(t), U_l^B(t), U_l^Y(t)\}$ are shared by all N trains for any given period, and the cost coefficients and the uncertainty parameters are the same for all trains. We use subscript n to denote the index of a train in the following generalised notation.

5.1. Generalised notation

$t_{d,n}^b$:	the beginning of the working time window for discharging the nth train ($t_{d,n}^b \geq 0$);
$t_{d,n}^e$:	the end of the working time window for discharging the nth train ($t_{d,n}^b < t_{d,n}^e$);
$t_{l,n}^b$:	the beginning of the working time window for loading the nth train ($t_{l,n}^b > t_{d,n}^b$);
$t_{l,n}^e$:	the end of the working time window for loading the nth train ($t_{l,n}^b < t_{l,n}^e$ and $t_{d,n}^e < t_{l,n}^e$);
$Q_{l,n}^B$:	the maximum number of containers allowed to be prestaged at the RT buffer for the nth train;
$Q_{d,n}^T$:	the total number of containers initially on the nth train to be discharged;
$Q_{l,n}^T$:	the total number of containers planned to be loaded onto the nth train;
Q_n^T:	the capacity of the nth train with $Q_n^T \geq \max\{Q_{d,n}^T, Q_{l,n}^T\}$;
$x_{d,n}^T(t)$:	the number of discharge containers left on the nth train at the beginning of time period t;
$x_{l,n}^B(t)$	the number of containers left in the RT buffer to be loaded on the nth train at the beginning of time period t;
$x_{l,n}^T(t)$	the number of containers loaded on the nth train at the beginning of time period t;
$\xi_{d,n}^Y(t)$	the random variable representing the actual container flow rate from the nth train to yard via RMG in period t;
$\xi_{l,n}^Y(t)$	the random variable representing the actual container flow rate from yard to the nth train via RMG in period t;
$\xi_{l,n}^B(t)$	the random variable representing the actual container flow rate from RT buffer to the nth train via RMG in period t;
$u_{d,n}^Y(t)$	the planned container flow rate from the nth train via RMG to yard in period t;
$u_{l,n}^Y(t)$	the planned container flow rate from yard to the nth train via RMG in period t;
$u_{l,n}^B(t)$	the planned container flow rate from RT buffer to the nth train via RMG in period t;
q_n	the number of containers prestaged at RT buffer to be loaded on the nth train initially, which is not greater than $Q_{l,n}^B$;

5.2. Generalised dynamic model

Similarly, we assume that the actual container flow rates follow uniform probability distributions in intervals with their ranges proportional to the planned flow rates. The discrete-time dynamic equations of container flows (including both discharge and load processes) can be given as follows:

$$x_{d,n}^T(t_{d,n}^b) = Q_{d,n}^T \tag{29}$$

$$x_{l,n}^B(t_{d,n}^b) = q_n \tag{30}$$

$$x_{l,n}^T(t_{d,n}^b) = 0 \tag{31}$$

$$x_{d,n}^T(t+1) = x_{d,n}^T(t) - \xi_{d,n}^Y(t) \tag{32}$$

$$x_{l,n}^B(t+1) = x_{l,n}^B(t) - \xi_{l,n}^B(t) \tag{33}$$

$$x_{l,n}^T(t+1) = x_{l,n}^T(t) + \xi_{l,n}^B(t) + \xi_{l,n}^Y(t) \tag{34}$$

$$0 \leqslant \sum q_n \leqslant Q_l^B \tag{35}$$

$$0 \leqslant u_{d,n}^Y(t) \leqslant \min\{x_{d,n}^T(t), C_{dl}(t), U_d(t)\} \cdot I\{t_{d,n}^b \leqslant t \leqslant t_{d,n}^e\} \tag{36}$$

$$0 \leqslant u_{l,n}^B(t) \leqslant \min\{x_{l,n}^B(t), C_{dl}(t), U_l^B(t)\} \cdot I\{t_{l,n}^b \leqslant t \leqslant t_{l,n}^e\} \tag{37}$$

$$0 \leqslant u_{l,n}^Y(t) \leqslant \min\{C_{dl}(t), U_l^Y(t)\} \cdot I\{t_{l,n}^b \leqslant t \leqslant t_{l,n}^e\} \tag{38}$$

$$u_{l,n}{}^Y(t) + u_{l,n}{}^B(t) \leqslant \min\{C_{dl}(t), Q_{l,n}{}^T - x_{l,n}{}^T(t), Q_n{}^T - x_{d,n}{}^T(t) - x_{l,n}{}^T(t)\} \tag{39}$$

$$\sum_n u_{d,n}{}^Y(t) \leqslant U_d(t) \tag{40}$$

$$\sum_n u_{l,n}{}^B(t) \leqslant U_l{}^B(t) \tag{41}$$

$$\sum_n u_{l,n}{}^Y(t) \leqslant U_l{}^Y(t) \tag{42}$$

$$\sum_n (u_{d,n}{}^Y(t) + u_{l,n}{}^Y(t) + u_{l,n}{}^B(t)) \leqslant C_{dl}(t) \tag{43}$$

$$\xi_{d,n}{}^Y(t) \sim Uniform(f_d{}^Y \cdot u_{d,n}{}^Y(t), u_{d,n}{}^Y(t)) \tag{44}$$

$$\xi_{l,n}{}^B(t) \sim Uniform(f_l{}^B \cdot u_{l,n}{}^B(t), u_{l,n}{}^B(t)) \tag{45}$$

$$\xi_{l,n}{}^Y(t) \sim Uniform(f_l{}^Y \cdot u_{l,n}{}^Y(t), u_{l,n}{}^Y(t)) \tag{46}$$

Let $\mathbf{x}_d{}^T(t) := (x_{d,1}{}^T(t), x_{d,2}{}^T(t), ..., x_{d,N}{}^T(t))$, $\mathbf{x}_l{}^B(t) := (x_{l,1}{}^B(t), x_{l,2}{}^B(t), ..., x_{l,N}{}^B(t))$. The dynamic programming optimality equation is given by (for $t = 0, 1, ..., T$):

$$
\begin{aligned}
J_t(\mathbf{x}_d{}^T(t), \mathbf{x}_l{}^B(t), \mathbf{x}_l{}^T(t)) = \ & \min E \sum_n \{c^P \cdot q_n \cdot I\{t = t_{d,n}{}^b\} \\
& + c^M \cdot x_{d,n}{}^T(t) \cdot I\{t = t_{d,n}{}^e + 1\} + c^M \cdot (Q_{l,n}{}^T - x_{l,n}{}^T(t)) \cdot I\{t = t_{l,n}{}^e + 1\} \\
& + c^S \cdot x_{l,n}{}^B(t) \cdot I\{t > t_{d,n}{}^b\} + c^T \cdot x_{d,n}{}^T(t) \cdot I\{t_{d,n}{}^b < t \leqslant t_{d,n}{}^e\} + c^T \cdot x_{l,n}{}^T(t) \cdot I\{t_{l,n}{}^b < t \leqslant t_{l,n}{}^e\} \\
& + c^Y \cdot (u_{d,n}{}^Y(t) + u_{l,n}{}^Y(t)) + c^B \cdot u_{l,n}{}^B(t) \\
& + c^D \cdot \xi_{d,n}{}^Y(t) + c^L \cdot (\xi_{l,n}{}^B(t) + \xi_{l,n}{}^Y(t)) \\
& + J_{t+1}(\mathbf{x}_d{}^T(t+1), \mathbf{x}_l{}^B(t+1), \mathbf{x}_l{}^T(t+1))\}
\end{aligned}
\tag{47}
$$

where $J_{t+1}(\mathbf{x}_d{}^T(t+1), \mathbf{x}_l{}^B(t+1), \mathbf{x}_l{}^T(t+1)) \equiv 0$ for $t = t_{l,n}{}^e$ and $1 \leqslant n \leqslant N$.

In principle, the optimal cost function, the optimal prestaging decision, and the optimal planned flow rates policy can be obtained using the value iteration algorithm similar to Section 4. However, due to the curse of dimensionality, it is computationally difficult to obtain the optimal solution to (29)–(47). Note that the empirical data in Fig. 3 shows that the load processes for different trains are completely decoupled. This implies that we can divide the integrated optimisation problem (29)–(47) into a series of sub-optimisation problems for individual trains. That is,

$$J_t(\mathbf{x}_d{}^T(t), \mathbf{x}_l{}^B(t), \mathbf{x}_l{}^T(t)) = \sum_n J_t(x_{d,n}{}^T(t), x_{l,n}{}^B(t), x_{l,n}{}^T(t)) \tag{48}$$

Where $J_t(x_{d,n}^T(t), x_{l,n}^B(t), x_{l,n}^T(t))$ is given in (17), and the handling capacities $\{C_{dl}(t), U_d(t), U_l{}^B(t), U_l{}^Y(t)\}$ apply to individual trains. Following the solution methods in Section 4, we can then solve the problem (48) in a sequential approach.

6. Numerical experiments

In this section, numerical experiments based on a real case study are performed to test the stochastic dynamic model, and to compare five different solution strategies. The data used in this study is retrieved from real data occurred on a weekday in one of the three rail terminals at the industrial port (see Fig. 3). The case port is one of the largest container ports in Northern Europe with an annual throughput in excess of 4 million TEUs. Nearly 30% of the containers are handled by rail at the port. The case terminal handles 7 inbound and 7 outbound trains per day; the number of containers loaded onto 7 trains are presented in Fig. 3.

The experimental results are discussed in three subsections:

In Section 6.1, we focus on a single train with loading process only. The optimal prestaging and loading rate policy will be illustrated in detail.

In Section 6.2, we consider a single train with both discharging and loading processes. Comparison of the optimal strategy with four simplified strategies is made in a range of scenarios in terms of solution quality and computational performance.

In Section 6.3, we apply four simplified strategies in multiple train situations to illustrate their applications.

The experiments are coded in Matlab R2015b running on a Windows 10 PC with CPU 3.00 GHz with RAM 4 GB.

6.1. Single train loading problem

We select the second train from Fig. 3 as a **Reference Case** to optimise its prestaging and loading process. Based on the empirical data from the real case, we calibrate the input data of the **Reference Case** as follows, by keeping the ordinal relationship among them:

- the time period is 30 min;
- the load time window is $t_l{}^b = 10$; $t_l{}^e = 15$;
- the number of containers to be loaded are $Q_l{}^T = 37$ containers;

Table 2
Optimal prestaging and cost with varying uncertainty degrees.

f_l^B, f_l^Y	q^*	$J_0(q^*, 0)$	$J_0(0, 0)$	%above
1, 1.0	0	236.50	236.50	0.00%
1, 0.8	6	279.21	279.81	0.21%
1, 0.6	23	300.99	315.45	4.80%
1, 0.4	30	311.06	358.90	15.38%
1, 0.2	30	321.77	414.16	28.71%

- the RMG cranes' loading capacity is $C_{dl}(t) \equiv 15$ containers in period t;
- the maximum flow rate from storage yards to RMG cranes in period t is $U_l^Y(t) = 15$
- the maximum flow rate from rail terminal buffer to RMG cranes in period t is $U_l^B(t) = 15$;
- the maximum capacity of prestaging $Q_l^B = 30$ containers;
- cost parameters are $c^P = 4$; $c^Y = 5$; $c^B = 2$; $c^D = c^L = 1$; $c^S = 0.1$; $c^T = 0.5$; $c^M = 20$.

It should be noted that the above cost coefficients are largely hypothetical and only the relative values of these cost elements are meaningful. The values of the cost parameters are based on real data that reflect the ordinal relationships among them. As illustrated in Fig. 1, the buffer is situated close to the rail track, and RMG is located next to the rail track. Therefore, unit cost of IMV transporting from yard to RMG, c^Y, is higher than pre-staging cost c^P (from yard to buffer); the pre-staging cost c^P is higher than the cost from buffer to RMG, c^B; and the penalty of missing discharging or loading a container is much higher than other costs. While unit costs of RMG loading to and discharging from trains, c^D and c^L, are usually lower than IMV's transportation costs c^Y and c^B. Storage costs on train or in buffer are introduced to influence decision making in the dynamic model, and they are much smaller than IMV transportation or RMG operation costs, with storage cost in buffer cheaper than storage cost in train. So, the order of unit costs in terms of their values is: $c^M > c^Y > c^P > c^B > c^D (\text{or } c^L) > c^T > c^S$.

Table 2 gives the results of the single train with loading process only under different scenarios of uncertainties. The Optimal Strategy is applied based on the stochastic dynamic approach. Column 2 shows the optimal prestaging decisions; column 3 and 4 show the optimal costs with and without prestaging, respectively. The percentage above (noted as %above) is introduced here, to indicate the percentage of $J_0(0, 0)$, i.e., the cost achieved under no prestaging scenario, above $J_0(q^*, 0)$ the cost achieved in optimal prestaging case. It can be observed that higher level of uncertainty leads to higher cost, and the cost saving from prestaging is increasing as the degree of uncertainty increases.

6.1.1. Optimal flow rate policies

To describe the optimal planned flow rate policies for loading activities, we take two examples from Table 2, including Example 1-the deterministic case $(f_l^B, f_l^Y) = (1, 1)$, and Example 2-the stochastic case $(f_l^B, f_l^Y) = (1, 0.6)$. In each case, an analysis is made for the flow rate policies derived under three different prestaging levels, with $q = 15$, 10 and 0. In Example 3, we compared the flow rate polices obtained under different levels of uncertainties, i.e. $(f_l^B, f_l^Y) = (1, 0.8/0.6/0.4/0.2)$.

- *Example 1 Deterministic case*

Although load window opens at $t = 10$, actual loading does not start until $t = 13$. The optimal flow rate policy $(u_l^Y(t), u_l^B(t))$ at period $t = 13$–15 with varying prestaging levels is shown in Table 3. It is clear that the flow rate policy $(u_l^Y(t), u_l^B(t))$ varies with time period t, the number of containers already loaded to the train $x_l^T(t)$, and the number of prestaged containers q. To simplify the narrative, we use u_l^Y to represent $u_l^Y(t)$, and the same applies to u_l^B and x_l^T.

In line with the analytical properties of the dynamic model summarised under points I)-V), a few observations can be made from Table 3 in terms of loading activities:

(i) Under the deterministic case, completing the loading task is guaranteed; as such the activity does not start until $t = 13$;
(ii) Loading always starts from prestaging buffer if prestaging level is positive; the loading amount increases with time period t and the prestaging level q, but decreases with the number of loaded containers x_l^T; when the prestaged containers q are exhausted, the rest of the containers are loaded from the yard in order to complete loading the planned number of containers Q_l^T.
- *Example 2 Stochastic case*

The second example is the stochastic case with $(f_l^B, f_l^Y) = (1, 0.6)$. The optimal flow rate policies (u_l^Y, u_l^B) with varying number of prestaged containers ($q = 0$, 10, and 15) are presented in Table 4 for periods $t = 12$–15.

Unlike the flow rate policy in the deterministic case, the flow rate cannot be fixed or represented in simple mathematical formula for certain periods. In deterministic situation, it is preferable to plan loading form rail buffer first to reduce the storage cost without incurring any penalty. While in stochastic situation with uncertain movement from yard to train but certain movement from rail buffer to train, it is preferable to plan loading from yard first and then from rail buffer, to reduce the possible heavy penalty. In

Table 3
Optimal loading flow rate policies with varying prestaged containers ($f_l^Y = 1.0$).

	x_l^T	u_l^B		x_l^T	u_l^B		x_l^T	u_l^B
$t = 15$ $q=15$	[0, 22]	15	$t = 15$ $q=10$	[0, 27]	10	$t = 15$ $q=0$	[0, 37]	0
	[23, 37]	$37 - x_l^T$		[28, 37]	$37 - x_l^T$			
	x_l^T	u_l^Y		x_l^T	u_l^Y		x_l^T	u_l^Y
	[0, 37]	0		[0, 22]	5		[0, 22]	15
				[23, 26]	$27 - x_l^T$		[23, 37]	$37 - x_l^T$
				[27, 37]	0			
	x_l^T	u_l^B		x_l^T	u_l^B		x_l^T	u_l^B
$t = 14$ $q = 15$	[0, 7]	15	$t = 14$ $q=10$	[0, 12]	10	$t = 14$ $q=0$	[0, 37]	0
	[8, 21]	$22 - x_l^T$		[13, 21]	$22 - x_l^T$			
	[22, 37]	0		[22, 37]	0			
	x_l^T	u_l^Y		x_l^T	u_l^Y		x_l^T	u_l^Y
	[0, 37]	0		[0, 7]	5		[0, 7]	15
				[8, 11]	$12 - x_l^T$		[8, 21]	$22 - x_l^T$
				[12, 37]	0		[22, 37]	0
	x_l^T	u_l^B		x_l^T	u_l^B		x_l^T	u_l^B
$t = 13$ $q=15$	[0, 6]	$7 - x_l^T$	$t = 13$ $q=10$	[0, 6]	$7 - x_l^T$	$t = 13$ $q=0$	[0, 37]	0
	[7, 37]	0		[7, 37]	0			
	x_l^T	u_l^Y		x_l^T	u_l^Y		x_l^T	u_l^Y
	[0, 37]	0		[0, 37]	0		[0, 6]	$7 - x_l^T$
							[7, 37]	0

comparison with the deterministic case, the planned flow rate becomes more irregular where uncertainty exists. However, some observations can still be drawn from Table 4:

(i) Loading starts earlier than the deterministic case, with a small flow rate u_l^Y from the yard, but reserving the prestaged containers until later working periods. As shown in Table 4, at $t = 12$, loading starts but with a small rate from the yard only.

(ii) Towards the last working period, the flow rate u_l^Y decreases, while u_l^B increases up to the full size of prestaging, with the changes dependent on x_l^T. To fully load Q_l^T and clear prestaging buffer, the maximum number of prestaged containers are loaded to the train, with the rest being loaded from the yard. In Table 4, towards the end of working window, u_l^B increases to be the full size of the prestaged number of containers, i.e., 15 and 10 respectively in the two prestaging scenarios $q = 15$ and 10.

(iii) Loading from yard is prioritised in the stochastic case, and loading is not made from prestaging until the last one or two working periods.

It can be seen that if the system is deterministic, the containers will be loaded onto the train in the final three periods in the working window (note that $C_{dl}(t) = 15$). However, when there exists uncertainty, the optimal policy tends to allocate more flow rates at early working periods. This would ensure that the train could be fully loaded under the uncertainty of container flow processes.

- *Example 3 A comparison of stochastic cases with varying uncertainties*

The findings above can be further illustrated in Fig. 4. Setting the prestaging $q = 15$, the flow rate policy varies with the level of uncertainty f_l^Y which ranges from 0.2 to 1.0, following the trend below:

(i) As the level of uncertainty increases, the flow rate policy becomes more conservative, as reflected by a) earlier loading and b) more loading.

(ii) As the level of uncertainty increases, more loading is made from the yard to the train.

Here are some examples for trend (i): as shown in Fig. 4, when $f_l^Y = 0.4$ and 0.6, loading starts at $t = 12$, with the flow rates being [0, 10] and [0, 5], respectively, but when uncertainty increases to be $f_l^Y = 0.2$, loading starts at $t = 11$, and the flow rate increases to be [0, 15]. The flow rate increases when uncertainty increases. With the decreasing uncertainty, for example, in the cases of $f_l^Y = 0.8$ and 1.0, loading does not start until $t = 13$. Comparing the two cases when $f_l^Y = 0.8$ and 1.0, a higher flow rate [0, 12] is executed in more uncertain situation when $f_l^Y = 0.8$, while a lower flow rate [0, 7] is made in deterministic case when $f_l^Y = 1.0$.

The flow rate above is the combined flow rate from both the yard and the RT buffer to the train. The dynamic programming model can also provide guidance on the specific flow rates from the yard and the buffer, as shown in Fig. 5. The impacts of the uncertainty level on the flow rate policy can be explained as: an increment in the level of uncertainty f_l^Y results in a decrement of average actual

Table 4
Optimal loading flow rate policies with varying prestaged containers ($f_l^Y = 0.6$).

t = 15

	x_l^T	u_l^B		x_l^T	u_l^B		x_l^T	u_l^B
	[0, 22]	15		[0, 27]	10		[0, 37]	0
$t = 15$	[23, 37]	$37 - x_l^T$	$t = 15$	[28, 37]	$37 - x_l^T$	$t = 15$		
$q=15$	x_l^T	u_l^Y	$q=10$	x_l^T	u_l^Y	$q=0$	x_l^T	u_l^Y
	[0, 37]	0		[0, 22]	5		[0, 22]	15
				[23, 26]	$27 - x_l^T$		[23, 37]	$37 - x_l^T$
				[27, 37]	0			

t = 14

	x_l^T	u_l^B		x_l^T	u_l^B		x_l^T	u_l^B
	[0, 10]	0		(1, 6)	(10, 5)		[0, 37]	0
	[11,18]	$18 - x_l^T$		others	0			
	[19, 37]	0						
$t = 14$	x_l^T	u_l^Y	$t = 14$	x_l^T	u_l^Y	$t = 14$	x_l^T	u_l^Y
$q=15$	[0, 10]	15	$q=10$	[0,2-5,7-15]	15	$q=0$	[0, 22]	15
	[11, 16]	10		(1, 6)	(5, 10)		[23, 37]	$37 - x_l^T$
	[17, 21]	5		[16-26]	$28 - x_l^T$		[22, 37]	0
	[22, 37]	0		[27, 37]	0			

t = 13

	x_l^T	u_l^B		x_l^T	u_l^B		x_l^T	u_l^B
	[0, 37]	0		[0, 37]	0		[0, 37]	0
	x_l^T	u_l^Y		x_l^T	u_l^Y		x_l^T	u_l^Y
	[0, 8]	15		[0, 5]	15		[0, 13]	15
$t = 13$	[9, 12]	10	$t = 13$	[6-8]	10	$t = 13$	[14-17]	10
$q=15$	(13, 15, 17, 20)	(5, 7, 5,2)	$q=10$	(9-11, 12, 16, 17, 20)	(5, 2, 5, 10, 2)	$q=0$	[18-27]	5
	others	0		others	0		[28, 37]	$35-x_l^T$

t = 12

	x_l^T	u_l^B		x_l^T	u_l^B		x_l^T	u_l^B
	[0, 37]	0		[0, 37]	0		[0, 37]	0
$t = 12$	x_l^T	u_l^Y	$t = 12$	x_l^T	u_l^Y	$t = 12$	x_l^T	u_l^Y
$q =15$	0	5	$q=10$	(0, 1)	(5, 2)	$q =0$	0	15
	[1, 37]	0		others	0		[1, 4]	10
							[5, 8]	5
							[9, 37]	0

loading rate, i.e., $E[\xi_l^Y(t)]$ and $E[\xi_l^B(t)]$, as such the optimal planned flow rates $u_l^B(t)$ and $u_l^Y(t)$ would naturally increase to accomplish loading and discharging the planned number of containers.

Trend (ii) is observed when the degree of uncertainty increases. This is due to the facts that advanced loading is conducted from the yard when sufficient time is allowed, and that the prestaged containers are reserved for more certain loading when the time window is going to end. Loading from RT buffer to train is deterministic, whereas loading from yard to train is uncertain. The dynamic model reserves the deterministic loading until the last few periods, to minimise the likelihood of paying penalty fines for any missed loading.
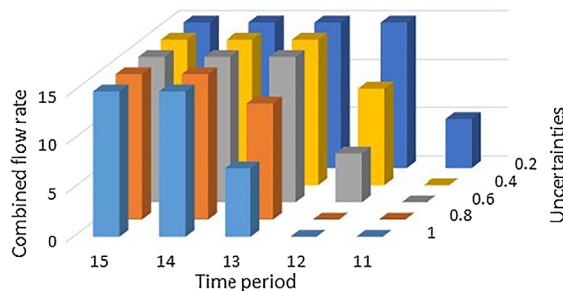
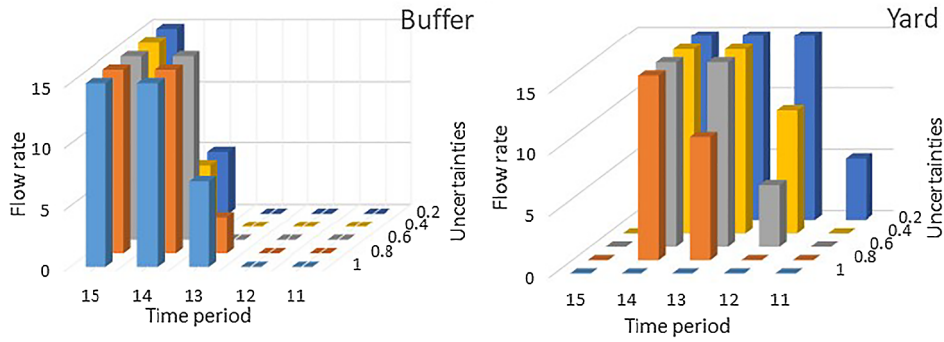**Fig. 4.** Combined flow rate (loading) with varying uncertainty degrees.

Fig. 5. Loading flow rates from yard or buffer with varying uncertainty degrees.

Table 5
Comparison of different strategies for single train loading problem.

f_l^B, f_l^Y	Optimal		Buffer-first		Yard-first		Bang-bang	
	$J_0(q^*,0)$	CPU	%above	CPU	%above	CPU	%above	CPU
1.0, 1.0	236.50	0.50	26.85%	0.05	26.85%	0.05	2.37%	0.08
1.0, 0.8	279.21	1.08	15.36%	0.03	15.36%	0.06	0.04%	0.06
1.0, 0.6	298.90	1.45	15.39%	0.05	15.39%	0.05	0.93%	0.08
1.0, 0.4	308.58	1.84	20.90%	0.05	21.91%	0.06	2.44%	0.08
1.0, 0.2	318.97	2.23	21.35%	0.05	31.78%	0.06	3.96%	0.13
0.5, 1.0	236.50	1.27	26.85%	0.06	26.85%	0.03	3.38%	0.09
0.5, 0.8	279.81	3.16	15.12%	0.06	15.12%	0.03	0.00%	0.11
0.5, 0.6	315.45	4.55	9.34%	0.09	9.34%	0.05	0.05%	0.13
0.5, 0.4	353.92	5.94	6.29%	0.11	6.29%	0.05	0.18%	0.16
0.5, 0.2	369.13	7.30	9.37%	0.14	13.87%	0.06	1.29%	0.19

6.1.2. Comparison of different solution strategies for single train loading problem

In this section, we compare the performance of different solution strategies applied to solve the single train loading problem. Note that as there are no discharging operations, the decoupled strategy become the same as the optimal strategy and is, therefore, omitted. Table 5 presents the cost and computational time (computer CPU time in seconds) of the four strategies (the **Optimal strategy**, the **Buffer-first strategy**, the **Yard-first strategy**, and the **Bang-bang strategy**), when they are applied to the Reference Case. The percentage above (noted as %above) is introduced to measure the percentage by which costs derived under alternative strategies are above the baseline cost (optimal strategy).

It is clear that Bang-bang significantly outperforms Buffer-first and Yard-first strategies in terms of total logistics cost only with slightly more computation time. Comparing with the optimal strategy, the Bang-bang strategy achieves 0%~3.96% above the optimal strategy with substantially less computation time. While the results obtained under Buffer-first resemble the results obtained from Yard-first, it outperforms Yard-first when there is a high level of uncertainty, for example when (f_l^B, f_l^Y) = (1.0, 0.2) and (f_l^B, f_l^Y) = (0.5, 0.2). These findings are in line with the rationale of using buffer to mitigate the risks imposed by uncertainties.

6.1.3. Discussions of solutions for single train loading problem

The loading plans discussed in Examples 1–3 show that dynamic programming model allows us to decide the prestaging number and flow rates (u_l^Y, u_l^B) to minimise the total cost. In the deterministic case, this is realised by retaining containers in the yard, loading from the RT buffer, and postponing loading to the last a few working periods. This yields two insights. In certain circumstances, containers should be kept in the yard and buffer for as long as possible, to minimise storage cost in the train. Loading is not executed until the late working periods, to avoid storage cost at train (and increase flexibility to customers) and free up resources such as RMG, IMVs and workforce for other duties. The flow rates identified by the dynamic model can facilitate creating the rail loading plan, planning and deploying required resources. For example, the flow rate from the buffer u_l^B decides the number of IMVs and RMGs required to perform the loading.

When compared to the deterministic case, the flow rates in the stochastic case are less predictable. We note from Table 4 and Fig. 5 that loading starts earlier than the deterministic case, but only from the yard. Loading from the buffer is pushed back to the last one or two working periods. In these cases, exploiting buffer holding to retain containers until late working periods may be more economically desirable. The prestaged containers can be used to address urgent needs arising from uncertain events, such as late arrival trains, or IMV breakdown, etc. This would ensure that the train could be fully loaded under the uncertainty of container flow processes. So this guidance is perfectly reasonable recommendations.

Whilst in the stochastic case prestaging is useful in reducing the total cost incurred (see Table 2), the number of prestaged containers has little influence on the loading plan. We note from Table 4 that the loading window remains as 4 periods (t = 12–15) despite of the

Table 6
Single train cases with varying time windows, uncertainty and penalty cost.

Case	$t_d{}^b$; $t_d{}^e$; $t_l{}^b$; $t_l{}^e$	$f_d{}^Y$; $f_l{}^B$; $f_l{}^Y$	c^M
1	10; 13; 12; 15	0.90; 0.90; 0.90	20
2	8; 13; 10; 15	0.90; 0.90; 0.90	20
3	8; 15; 10; 17	0.90; 0.90; 0.90	20
4	10; 13; 12; 15	0.80; 0.90; 0.60	20
5	8; 13; 10; 15	0.80; 0.90; 0.60	20
6	8; 15; 10; 17	0.80; 0.90; 0.60	20
7	10; 13; 12; 15	0.40; 0.90; 0.20	20
8	8; 13; 10; 15	0.40; 0.90; 0.20	20
9	8; 15; 10; 17	0.40; 0.90; 0.20	20
10	10; 13; 12; 15	0.80; 0.90; 0.60	40
11	8; 13; 10; 15	0.80; 0.90; 0.60	40
12	8; 15; 10; 17	0.80; 0.90; 0.60	40
13	10; 13; 12; 15	0.80; 0.90; 0.60	10
14	8; 13; 10; 15	0.80; 0.90; 0.60	10
15	8; 15; 10; 17	0.80; 0.90; 0.60	10

prestaging number, and that loading does not start from the buffer until the last one or two periods. This phenomenon is owing to the facts that the unit storage cost on the train is higher than the unit storage cost in the buffer, and that the objective is to minimise the total cost. In addition to minimising the overall cost of loading, it would be interesting to set multiple objectives for the dynamic programming model, such as maximising customer service level, minimising total operation time or minimising total carbon emissions. It is therefore worth analysing the compromised solutions for these conflicting objectives and revisiting the resulting loading plan.

We have shown how the loading plan could be created in response to different prestaging situations and uncertain scenarios. Clearly a properly defined loading plan can help a seaport rail terminal to handle uncertain events more effectively, so as to minimise missed containers and ensure customer satisfaction while minimising the total cost. However, in practice, uncertain circumstances may be complex and hard to model, so we do not think a loading plan can be created by the dynamic programming model alone without investigation into uncertainties and resource availability, and with subjective judgement.

6.2. Single train discharging and loading problem

We choose the second train in Fig. 3 as the **Reference Case** to optimise its discharging and loading processes. The input data of the **Reference Case** is as follows:

- $t_d{}^b = 8$; $t_d{}^e = 13$; $t_l{}^b = 10$; $t_l{}^e = 15$;
- containers to be discharged are $Q_d{}^T = 43$, and containers to be loaded are $Q_l{}^T = 37$;
- the RMG cranes' discharging and loading capacities are $C_{dl}(t) \equiv 30$ containers per period;
- $U_d(t) \equiv 15$; $U_l{}^Y(t) = 15$; $U_l{}^B(t) = 15$;
- uncertainty parameters are $f_d{}^Y = 0.8$; $f_l{}^B = 0.9$ and $f_l{}^Y = 0.6$;
- cost parameters are $c^P = 4$; $c^Y = 5$; $c^B = 2$; $c^D = c^L = 1$; $c^S = 0.1$; $c^T = 0.5$; $c^M = 20$.

As working time windows and the degree of uncertainties are important factors that affect the planning of the flow rates and the prestaging decision, we will vary these two sets of parameters in the **Reference Case**. In addition, the penalty cost coefficient c^M is selected to examine its impact on the decisions and the performance. Fifteen cases are examined and presented in Table 6, in which Case 1 is the **Reference Case**.

6.2.1. Results without prestaging

Firstly, let us exclude the prestaging decision by setting $Q_l{}^B = 0$. The results of fifteen cases achieved under five strategies, i.e., the optimal strategy, the Decoupled strategy, two greedy strategies, and Bang-bang strategy, are shown in Table 7. Using the cost achieved under the optimal strategy as a baseline, the percentage above (noted as %above) is introduced to measure the percentage by which costs derived under alternative strategies are above the baseline cost, as shown in the "%above" columns. The CPU running times (in seconds) required by the three strategies are also compared, as shown "CPU" columns.

Based on the results in Table 7, the impacts of time window size, the degree of uncertainties, and the penalty cost can be summarised as below:

(i) As the length of the working time window increases (e.g. from Case 1 to Case 3), the expected cost is decreasing, while the computational time is increasing;
(ii) As the degree of uncertainty increases (e.g. from Case 1 to Case 4 to Case 7), the expected cost is increasing, and the computational time is also going up;

Table 7
Comparison of five strategies without prestaging decision.

Case	Optimal		Decoupled		Buffer-first		Yard-first		Bang-bang	
	Cost	CPU	%above	CPU	%above	CPU	%above	CPU	%above	CPU
1	566.4	0.55	13.95%	0.08	2.76%	0.05	2.76%	0.02	0.04%	0.06
2	564.1	0.98	3.50%	0.09	8.98%	0.03	8.98%	0.06	0.04%	0.06
3	563.5	1.44	0.88%	0.13	15.47%	0.08	15.47%	0.05	0.04%	0.09
4	629.3	1.02	18.10%	0.11	0.03%	0.03	0.03%	0.03	0.03%	0.08
5	626.2	1.92	5.02%	0.14	4.70%	0.08	4.70%	0.08	0.03%	0.13
6	625.5	2.88	1.26%	0.17	10.46%	0.11	10.46%	0.08	0.03%	0.14
7	863.9	2.89	14.03%	0.17	0.00%	0.08	0.00%	0.08	0.00%	0.09
8	810.5	5.59	12.61%	0.19	0.80%	0.13	0.80%	0.14	0.01%	0.20
9	806.5	8.64	5.33%	0.25	4.44%	0.19	4.44%	0.20	0.04%	0.25
10	655.2	1.00	51.74%	0.11	0.00%	0.03	0.00%	0.03	0.00%	0.05
11	644.4	1.92	15.34%	0.13	2.55%	0.08	2.55%	0.08	0.07%	0.09
12	641.9	2.86	3.85%	0.17	7.83%	0.08	7.83%	0.08	0.07%	0.16
13	604.4	1.05	2.17%	0.09	2.01%	0.03	2.01%	0.08	0.03%	0.08
14	604.9	1.95	0.58%	0.13	7.97%	0.06	7.97%	0.06	0.03%	0.09
15	605.3	2.88	0.15%	0.14	14.05%	0.08	14.05%	0.08	0.03%	0.13

(iii) As the penalty cost coefficient C^M increases (e.g. from Cases 13 to Case 4 to Case 10), the expected cost increases while the computational times are quite similar;

(iv) These observations are generally in agreement with intuition;

(v) The expected cost under the decoupled strategy is 0.15%–51.74% above that under the optimal strategy; the expected costs under both Buffer-first and Yard-first strategies range within 0.00%–15.74%; and the expected cost under the Bang-bang strategy is 0.00–0.07% above the optimal cost. It is obvious that the Bang-bang strategy outperforms all other alternative strategies in both solution quality and computation time. In particular, the Bang-bang strategy can achieve a performance extremely close to the optimal strategy.

6.2.2. Results with prestaging

Secondly, let us include the prestaging decision by setting the maximum allowed prestaged containers $Q_l^B = 10$. The results of fifteen cases under five strategies are shown in Table 8. Similar patterns as those in Table 7 can be observed in Table 8. But the computation time under the optimal strategy increases substantially, which is a result of the increased curse of dimensionality. Particularly, the computation time increases from less than ten seconds to be over ten minutes in Case 9. This can be explained by the fact that high degree of uncertainty significantly increases the action space and the output space which imposes computational burden substantially. Tables 7 demonstrate that the decoupled strategy's cost is 0–12.40% above the optimal cost; the Buffer-first strategy's cost is 0.82–15.47% above the optimal cost, the Yard-first strategy's cost is 0.51–15.74% above the optimal cost; and the Bang-bang strategy's cost is 0.00–0.19% above the optimal cost. In terms of computational complexity, all four simplified strategies require substantially less CPU time (over 90% less) than the optimal strategy, with the Yard-first being the most computation

Table 8
Comparison of five strategies with prestaging decision.

Case	Optimal		Decoupled		Buffer-first		Yard-first		Bang-bang	
	Cost	CPU	%above	CPU	%above	CPU	%above	CPU	%above	CPU
1	566.4	38.8	1.85%	2.63	2.76%	0.48	2.76%	0.47	0.04%	0.89
2	564.1	75.7	1.92%	2.80	8.98%	0.48	8.98%	0.45	0.04%	1.17
3	563.5	111.3	0.88%	3.11	15.47%	0.70	15.47%	0.61	0.04%	1.58
4	624.3	78.5	3.60%	4.00	0.82%	0.48	0.82%	0.42	0.00%	1.05
5	623.3	154.0	0.92%	4.47	5.20%	0.77	5.20%	0.64	0.00%	1.67
6	624.5	229.5	0.23%	4.67	10.63%	1.03	10.63%	0.89	0.00%	2.36
7	820.8	228.4	6.84%	5.98	1.79%	1.03	3.76%	0.89	0.19%	2.02
8	779.9	446.6	5.29%	6.69	2.92%	1.80	4.75%	1.52	0.16%	3.63
9	777.9	684.8	2.03%	7.41	6.73%	2.55	8.27%	2.17	0.19%	5.34
10	649.8	77.7	12.40%	3.81	0.84%	0.47	0.51%	0.44	0.03%	1.00
11	641.3	153.0	3.40%	4.25	3.03%	0.73	3.03%	0.66	0.03%	1.66
12	640.9	227.8	0.85%	4.72	8.00%	1.02	8.00%	0.86	0.02%	2.33
13	602.2	78.4	0.00%	3.80	2.39%	0.47	2.39%	0.42	0.01%	1.00
14	604.6	152.3	0.00%	4.23	8.02%	0.78	8.02%	0.64	0.02%	1.63
15	605.3	228.2	0.15%	4.69	14.05%	1.03	14.05%	0.89	0.03%	2.31

Table 9
Application of decoupled, bang-bang and greedy strategies to multi-train cases.

$Q_{l,n}{}^B$	$f_d{}^Y$; $f_l{}^B$; $f_l{}^Y$	Decoupled		Buffer-first		Yard-first		Bang-bang	
		%above	CPU	%above	CPU	%above	CPU	Cost	CPU
0	0.9; 0.9; 0.9	0.86%	0.47	12.36%	0.16	12.36%	0.28	2615.9	0.31
0	0.8; 0.9; 0.6	1.54%	0.66	7.37%	0.31	7.37%	0.30	2921.5	0.42
0	0.6; 0.9; 0.4	3.10%	0.89	4.51%	0.42	4.51%	0.44	3294.6	0.55
0	0.4; 0.9; 0.2	5.24%	0.95	2.19%	0.59	2.19%	0.55	3787.6	0.75
0	0.2; 0.9; 0.0	6.17%	1.08	0.95%	0.80	0.95%	0.78	4308.8	0.92
5	0.9; 0.9; 0.9	0.65%	5.72	12.36%	1.36	12.36%	1.13	2615.9	2.78
5	0.8; 0.9; 0.6	0.90%	8.92	7.41%	1.92	7.41%	1.58	2920.4	3.91
5	0.6; 0.9; 0.4	1.82%	11.33	5.37%	2.98	5.38%	2.36	3267.5	5.64
5	0.4; 0.9; 0.2	3.23%	13.92	3.35%	4.44	3.85%	3.59	3726.7	7.94
5	0.2; 0.9; 0.0	4.10%	15.50	1.80%	5.92	3.13%	4.69	4217.7	10.45
10	0.9; 0.9; 0.9	0.52%	14.72	12.36%	2.25	12.36%	2.03	2615.9	4.86
10	0.8; 0.9; 0.6	0.22%	23.30	7.73%	3.33	7.73%	2.80	2911.9	6.92
10	0.6; 0.9; 0.4	0.67%	30.48	6.48%	5.08	6.62%	4.34	3229.3	10.19
10	0.4; 0.9; 0.2	1.73%	36.95	4.76%	7.28	6.17%	6.39	3645.4	14.06
10	0.2; 0.9; 0.0	2.20%	40.47	3.05%	9.41	5.77%	8.16	4112.3	18.02

efficient, followed by Buffer-first strategy, Bang-bang strategy and Decoupled strategy. Overall, the Bang-bang strategy is the best choice due to its high solution quality and efficiency in computation effort.

6.2.3. Discussions of solutions for single train discharging and loading problem

Comparing the single train results achieved with and without prestaging, we found that the total costs incurred under the prestaging scenario are smaller than the total costs derived without prestaging, regardless of the choice of solution strategy. This shows that the seaport can use prestaging to reduce the cost involved.

Analysing the single train results under varying factors, such as window sizes, uncertainties and penalty cost, we observed the same pattern of relationship from scenarios without and with prestaging, i.e., the expected cost and computation time increase along with the length of the working time window, the degree of uncertainty and the penalty cost. A lower degree of uncertainty will reduce the total cost significantly, with a higher degree of uncertainty having the opposite impact. This urges seaports to handle uncertainties effectively and minimise disturbance imposed by uncertainties. In contrast, window sizes have less impacts on the total cost. Train arrival and departure times are usually considered fixed, but subject to periodic change. Our results show that a more open working window would not bring significant economic gains, although it might release pressure on the workforce or equipment. The penalty cost must be addressed in reality by minimising the missed containers and avoiding late departures.

6.3. Multiple trains with discharging and loading

The above sections showed that the simplified strategies are computationally efficient and more readily applicable in practice. In particular, the Bang-bang strategy is able to achieve near-optimal solutions. Therefore, we focus on these four simplified strategies in the multiple train situations. Note that Fig. 3 displays seven trains handled within a day where the first train and the last train are only partially handled (either partially discharged or partially loaded). We select the middle five trains by excluding the first and the last trains in our experiments, which specifies $t_{d,n}{}^b$; $t_{d,n}{}^e$; $t_{l,n}{}^b$; $t_{l,n}{}^e$;$Q_{d,n}{}^T$ and $Q_{l,n}{}^T$ for train $n = 1,2, …,5$.

6.3.1. Results under two optimal strategies

Let the maximum number of allowed prestaged containers ($Q_{l,n}{}^B$) take three values: 0, 5 and 10; let the degree of uncertainties ($f_d{}^Y$; $f_l{}^B$; $f_l{}^Y$) take four levels as shown in Table 5. Other system parameters are the same as the **Reference Case** described in Section 6.2. The results achieved under the decoupled strategy and the Bang-bang strategy are shown in Table 9. The percentage above (% above) is calculated as the percentage by which the expected cost achieved under other alternative strategies is above the cost achieved under the Bang-bang strategy.

Observations made from Table 9 can be summarised as:

(i) Compared with the other alternative strategies, the Bang-bang strategy presents superior results in terms of both solution quality and computational time;

(ii) As the degree of uncertainty increases, the extent by which the Bang-bang strategy outperforms the decoupled strategy also increases, denoted by the percentage above (%above); however, results produced by Buffer-first and Yard-first strategies get closer to results achieved under Bang-bang strategy; This reflects the intuition that greedy strategies tend to work well as the degree of uncertainty increases.

(iii) As the maximum allowed prestaged containers increases, the total cost decreases;

(iv) The computation time increases significantly as the degree of uncertainty increases, or as the maximum number of allowed prestaged containers increases.

The above observations are in line with the results in Section 6.2.

The percentage above (% above) and the computation time have an opposite trend in relation to the degree of uncertainties and the maximum number of allowed prestaged number. When the degree of uncertainties increases, the percentage above and computation time increase, as reported in Table 9. This means that, when many uncertain factors are imposed, which can be weather condition, train timetable, or machine breakdown, a greater total cost is necessary, while it takes longer to produce effective loading or discharging plans.

6.3.2. Discussions of solutions for multiple train loading and discharging problem

Whilst the observations in the single train context provide insights as to how total cost is affected by different factors, in practice it is possible that two or more trains are handled at the same time. We note from Table 9 that total costs in the multiple train scenario change with the degree of uncertainty in the same pattern as the single train scenario. A higher total cost is necessary to address higher uncertainty. Of interest is the conclusion that the cost saving made by the Bang-bang strategy is significant when measured against other three simplified strategies, and the level of saving increases when the degree of uncertainty increases. While results produced by Buffer-first and Yard-first greedy strategies more resemble the results obtained from Bang-bang strategy as level of uncertainty increases.

The maximum number of allowed prestaged containers $Q_{l,n}^B$ can reduce total cost, although it takes longer to produce a loading/discharging plan. Our results show that the higher cost incurred by the higher degree of uncertainty can be alleviated (offsetting some cost) by a bigger prestaging buffer. We have shown how a seaport can "tune" a prestaging buffer size to address a variety of uncertainty levels. The buffer size that will be the best in a given situation will depend on the real situation. For example, a seaport that encounters very uncertain train arrival time may increase the prestaging buffer size to offset disturbances related to the arrival time and working window.

The quantitative results prove that the Bang-bang strategy can achieve a performance extremely close to the optimal strategy, making the Bang-bang strategy a more applicable strategy in practice.

In the case company, train loading/discharging plan is created manually and relies heavily on user's expertise and judgement. There is little consideration of uncertainties or operational disturbances, such as equipment breakdown, variations in workload or in customer demands; as such the plan is very static. Our industrial partners have carefully examined and evaluated the results presented by the dynamic programming model in terms of the loading/discharging plan, associated cost, customer service and flexibility in cost adjustment in their business. They feel that the proposed model can be used in conjunction with the already present train planning tool to reduce the reliance on subjective judgement, therefore creating a more dynamic plan that is updated by responding to operational changes and uncertainties.

The optimal strategies have different features and implications for practice; it is an intention of this research that comparison of the results may be useful for the company's experts under different circumstances. Results achieved under different strategies may offer meaningful managerial insights for the decision analysts. In comparison, the optimal strategy provides the most optimal results in terms of total cost, but at the expense of long computation time, so a win lose situation. While the Decoupled, Buffer-first, Yard-first and Bang-bang strategies are able to reduce the computation time substantially, the solutions are characterised by higher costs than the solutions provided by the optimal strategy. However, a slight increment in cost can be endured and compensated by a much-reduced computation time, making the overall solution presented by the Bang-bang strategy better, to be a win–win situation.

7. Conclusions

This paper presents an integrated model for optimising the decisions of container prestaging and the decisions of container discharging and loading rates at seaport rail terminals, in the presence of uncertainties. We formulate the problem into a stochastic dynamic programming model to minimise the expected cost incurred during the logistics processes of transporting and storing containers between the storage yards and the trains. Five solution strategies, i.e., the optimal strategy, the two greedy strategies (i.e., Buffer-first and Yard-first strategies), the decoupled strategy and the Bang-bang strategy, are presented. Numerical experiments based on a real case study are conducted to compare these strategies and illustrate their effectiveness and sensitivity to the key system parameters.

It should be noted that the four simplified strategies are obtained via the value iteration algorithms rather than heuristically. This implies that the simplified strategies are optimal within the specified type of control policies (or action space). The value iteration algorithms also help to identify the optimal static prestaging decisions under each proposed strategy. This work has the potential to be integrated with the existing train loading and discharging operations, at least complementing the current manual operation and producing closer to optimal results.

However, there are limitations on the proposed model. In this research, we pursue a macroscopic approach to derive the aggregated container flow rate, without considering detailed container attributes (such as size, weight, commercial value etc.), train attributes (wagon types, shunting time, etc.), resource management (RMG allocation, labour availability and allocation, etc.), or yard management. The schedules of individual containers and resources are not modelled, mainly due to the existence of uncertain factors. The proposed model is tested using the calibration dataset through total cost indicators, but other operational performance indicators such as turnaround time, service time, or financial indicators are not considered.

Further research could be done in the following directions: (i) considering the dual-cycle IMV movements by appropriately linking the discharging and loading processes; (ii) considering the headcount and flexible resource management at seaport rail terminals; (iii) considering more detailed operations such as individual IMV movements on the real-time basis; (iv) considering more container attributes, train attributes and the differences in problem setting in different countries; (v) assigning buffer capacity to multiple trains dynamically; and (vi) including multiple rail terminals in the analysis.

Acknowledgements

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.tre.2018.09.008.

References

Ambrosino, D., Bramardi, A., Pucciano, M., Sacone, S., Siri, S., 2011. In: In: Proc. of the 7th annual IEEE Conference on Automation Science and Engineering, pp. 208–213.

Ambrosino, D., Caballini, C., Siri, S., 2013. A mathematical model to evaluate different train loading and stacking policies in a container terminal. Maritime Econ. Logistics 15, 292–308.

Ambrosino, D., Siri, S., 2015. Comparison of solution approaches for the train load planning problem in seaport terminals. Transp. Res. Part E 79, 65–82.

Anghinolfi, D., Paolucci, M., 2014. A general purpose Lagrangian heuristic applied to the train loading problem. Procedia – Social Behav. Sci. 108, 37–46.

Bertsekas, D.P., 1976. Dynamic programming and stochastic control. Academic Press, New York.

Bierwirth, C., Meisel, F., 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. Eur. J. Oper. Res. 202, 615–627.

Bruns, F., Knust, S., 2012. Optimized load planning of trains in intermodal transportation. OR Spectrum 34 (3), 511–533.

Caballini, C., Pasquale, C., Sacone, S., Siri, S., 2014. An event-triggered receding-horizon scheme for planning rail operations in maritime terminals. IEEE Trans. Intell. Transp. Syst. 15 (1), 365–375.

Caballini, C., Fioribello, S., Sacone, S., Siri, S., 2016. An MILP optimization problem for sizing port rail networks and planning shunting operations in container terminals. IEEE Trans. Autom. Sci. Eng. 13 (4), 1492–1503.

Carlo, H.J., Vis, I.F.A., Roodbergen, K.J., 2014. Transport operations in container terminals: literature overview, trends, research directions and classification scheme. Eur. J. Oper. Res. 236, 1–13.

Corry, P., Kozan, E., 2008. Optimised loading patterns for intermodal trains. OR Spectrum 30 (4), 721–750.

Dobos, I., Richter, K., 2004. An extended production/recycling model with stationary demand and return rates. Int. J. Prod. Econ. 90 (3), 311–323.

Dobos, I., Richter, K., 2006. A production/recycling model with quality considerations. Int. J. Prod. Econ. 104 (2), 571–579.

Evans, L.C., 2010. An Introduction to Mathematical Optimal Control Theory. University of California, Berkeley.

Gillen, D., Hasheminia, H., 2018. Empirical analysis and simulation modeling of a canadian seaport transportation network. J. Supply Chain Oper. Manage. 16 (1), 17–35.

Gunther, H.O., Kim, K.H., 2006. Container terminals and terminal operations. OR Spectrum 28 (4), 437–445.

Han, X., Lu, Z., Xi, L., 2010. A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. Eur. J. Oper. Res. 207 (3), 1327–1340.

Heggen, H., Braekers, K., Caris, A., 2016. Optimizing train load planning: review and decision support for train planners, international conference on computational logistics. Lecture Notes Comput. Sci. (LNCS) 9855, 193–208.

Kim, K.H., Lee, H., 2015. Container Terminal Operation: Current Trends and Future Challenges. In: Lee, C.-.Y., Meng, Q. (Eds.), Handbook of Ocean Container Transport Logistics– Making Global Supply Chain Effective. Springer, New York, pp. 43–74.

Murty, K.G., Wan, Y.W., Liu, J., 2005. Hongkong International Terminals gains elastic capacity using a data-intensive decision-support system. Interfaces 35 (1), 61–75.

Petering, M.E.H., 2011. Decision support for yard capacity, fleet composition, truck substitutability, and scalability issues at seaport container terminals. Transp. Res. Part E 47, 85–103.

Powell, W.B., 2011. Approximate dynamic programming: solving the curse of dimensionality, second ed. Wiley, New Jersey.

Puterman, M.L., 2005. Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York.

Richter, K., 1997. Pure and mixed strategies for the EOQ repair and waste disposal problem. OR Spectrum 19 (2), 123–129.

Roso, V., 2007. Evaluation of the dry port concept from an environmental perspective: A note. Transp. Res. Part D 12, 523–527.

Salama, Y., 2000. Optimal control of a simple manufacturing system with restarting costs. Oper. Res. Lett. 26 (1), 9–16.

Schonemann, R., 2010. Integrating Railway Services into the Supply Chain at the Last Mile of the Transhipment Interface Seaport-Rail. In: Conference Proceeding: international scientific conference, Białowieża, 21-24 September 2010.

Sennott, L.I., 1999. Stochastic Dynamic Programming and the Control of Queueing Systems. John Wiley & Sons, New York.

Song, D.P., 2005. Optimal threshold control of empty vehicle redistribution in two depot service systems. IEEE Trans. Autom. Control 50 (1), 87–90.

Song, D.P., 2013. Optimal Control and Optimization in Stochastic Supply Chain Systems. Springer, London.

Song, D.P., Sun, Y.X., Xing, W., 1998. Optimal control of a stochastic assembly production line. J. Optim. Theory Appl. 98 (3), 681–700.

Stahlbock, R., Voß, S., 2008. Operation research at container terminals - a literature update. OR Spectrum 30 (1), 1–52.

Steenken, D., Voß, S., Stahlbock, R., 2004. Container terminal operations and operations research—a classification and literature review. OR Spectrum 26, 3–49.

Taha, H.A., 2011. Operations Research, an introduction, 9th Ed. Pearson, New Jersey.

Tsoularis, A., 2014. Deterministic and stochastic optimal inventory control with logistic stock-dependent demand rate. Int. J. Mathem. Oper. Res. 6 (1), 41–69.

UNCTAD, 2017. Review of Maritime Transport 2017. Development, Geneva.

Veinott, A., 1966. On the optimality of (s, S) inventory policies: new condition and a new proof. SIAM J. Appl. Math. 14 (5), 1067–1083.

Zheng, Y.S., Federgruen, A., 1991. Finding optimal (s, S) policies is about as simple as evaluating a single policy. Oper. Res. 39 (4), 654–665.

Zhen, L., 2014. Container yard template planning under uncertain maritime market. Transp. Res. Part E 69, 199–217.

Zhen, L., 2015. Tactical berth allocation under uncertainty. Eur. J. Oper. Res. 247, 928–944.

Zhen, L., Xu, Z., Wang, K., Ding, Y., 2016. Multi-period yard template planning in container terminals. Transport. Res. Part B: Methodol. 93 (B), 700–719.

Zhou, P., Kang, H., 2008. Study on berth and quay-crane allocation under stochastic environments in container terminal. Syst. Eng.-Theory Pract. 28 (1), 161–169.