

Yard crane and AGV scheduling in automated container terminal: A multi-robot task allocation framework

Xuchao Chen^a, Shiwei He^{a,*}, Yongxiang Zhang^b, Lu (Carol) Tong^{c,*}, Pan Shang^a, Xuesong Zhou^d

^a School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

^b School of Transportation and Logistics, Southwest Jiaotong University, Chengdu 610031, China

^c Research Institute of Frontier Science, Beihang University, Beijing 100091, China

^d School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe, AZ 85281, USA



ARTICLE INFO

Keywords:

Automated container hub
Multi-robot system
Crane scheduling
ADMM
Rolling horizon

ABSTRACT

The efficiency of automated container terminals primarily depends on the synchronization of automated-guided vehicles (AGVs) and automated cranes. Accordingly, we study the integrated rail-mounted yard crane and AGV scheduling problem as a multi-robot coordination and scheduling problem in this paper. Based on a discretized virtualized network, we propose a multi-commodity network flow model with two sets of flow balance constraints for cranes and AGVs. In addition, two side constraints are introduced to deal with inter-robot constraints to reflect the complex interactions among terminal agents accurately. The Alternating Direction Method of Multipliers (ADMM) method is adopted in this study as a market-driven approach to dualize the hard side constraints; therefore, the original problem is decomposed into a set of crane-specific and vehicle-specific subtasks. The cost-effective solutions can be obtained by iteratively adjusting both the primal and dual costs of each subtask. We also compare the computational performance of the proposed solution framework with that of the resource-constrained project scheduling problem (RCPSP) model using commercial solvers. Comparison results indicate that our proposed approach could efficiently find solutions within 2% optimality gaps. Illustrative and real-world instances show that the proposed approach effectively serves the accurate coordination of AGVs and cranes in automated terminals.

1. Introduction

Container hubs, such as railway container terminals and container ports, are critical bottlenecks in global supply networks. This is particularly relevant, as container volume worldwide has continued to grow for decades and is expected to continue to grow in the future (Wu et al., 2015). In the context of Industry 4.0 (BMBF, 2016), freight transportation industries are extremely interested in exploring the use of the emerging technologies, such as the fifth-generation cellular network technology (5G), high-resolution cameras, and connected and automated guided vehicles (AGVs), for next-generation automated container hubs.

Currently, approximately 3% of container hubs are either semi-automated or fully automated (Lysdal, 2018). Automation has brought significant benefits to the hubs. The Port of Rotterdam (Netherlands) is the first port in the world with AGVs and automated

* Corresponding authors.

E-mail addresses: xc_chen@bjtu.edu.cn (X. Chen), shwhe@bjtu.edu.cn (S. He), bk20100249@my.swjtu.edu.cn (Y. Zhang), ltong@buaa.edu.cn (L.C. Tong), shangpan@bjtu.edu.cn (P. Shang), xzhou74@asu.edu (X. Zhou).

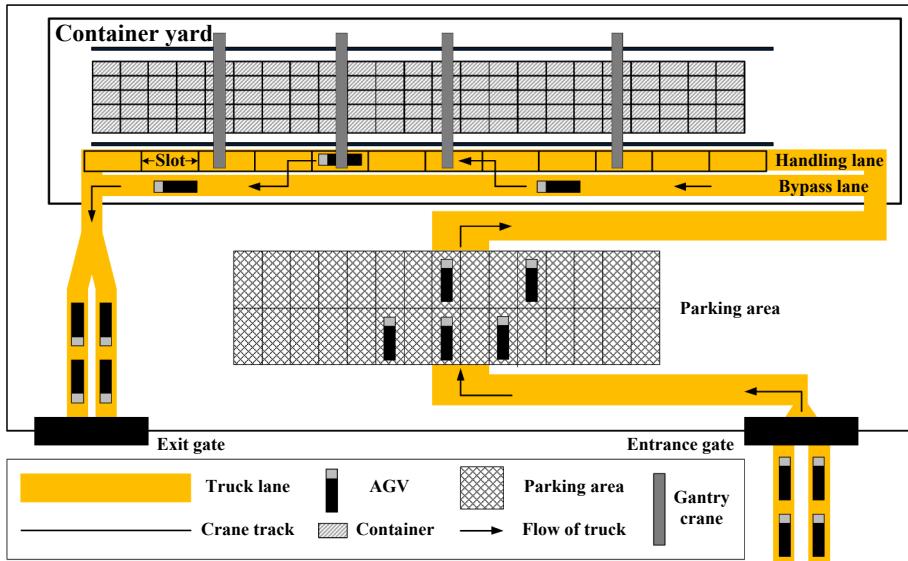


Fig. 1. Typical schematic of a container terminal with cranes and AGVs.

terminals. The Maasvlakte 2 Terminal of the Port of Rotterdam has an annual throughput capacity of 4.5 million twenty-foot equivalent units (TEUs) with the help of automated ship-to-shore cranes and battery-powered lift AGVs. In the Port of Qingdao (China), two fully automated berths equipped with 38 automated stacking cranes and 38 AGVs became operational in 2017. The purpose of bringing automation to container hubs is to reach a new level of productivity, safety, and sustainability. According to survey data provided by McKinsey & Company (2018), a fully automated terminal can increase productivity by 10%–35% while reducing the number of required workers by at least 45% and operating expenses by 25%–55%.

To make use of the emerging applications of automated robot control and communication technologies for container hubs, terminal managements critically need an efficient and reliable planning framework to schedule multiple automated robots within hubs. This study specifically considers the synchronization and path planning problem for two types of robots (automated cranes and AGVs), called the “crane and AGV coordination and scheduling problem” (CACSP) in this study.

1.1. Challenges in CACSP for container hub automation

In the automated container hub, AGVs and automated cranes as heterogeneous robots need to work together to perform container handling tasks. As shown in Fig. 1, a container terminal usually encompasses a container yard, an entrance gate, an exit gate, and a parking area for vehicles. The workflow of the terminal can be summarized as follows:

AGVs first need to be inspected at the entrance gate before entering the terminal. A limited number of truck lanes impose a gate capacity constraint on AGV arrivals during the inspection process. For example, in Fig. 1, two truck lanes at the entrance gate allow up to two AGVs to be inspected simultaneously. If the gate is saturated, AGVs will need to wait at the gate. After inspection, the AGVs will move into the parking area and wait there until being scheduled to the container yard, where several rail-mounted gantry cranes (2–4 cranes in general) move horizontally along the crane track to load/unload containers to/from AGVs. In this paper, we focus on the scheduling problems in the terminals, where the handover operation of containers is executed along the spread of the yard sideways to the pathway of cranes (Boysen et al., 2017). A container yard is usually spatially decomposed into smaller line segments (Boysen and Fliedner, 2010), which are referred to as slots (as shown in Fig. 1). When an AGV enters the container yard, it first travels on the bypass lane to approach the parking slot which is located in the handling lane. Meanwhile, a crane will move to the designated slot to lift the corresponding container. As we assume that each container to be loaded (unloaded) has a known origin (destination) location in the yard, the parking slot of each AGV is predetermined. However, it should be noted that, due to the flexible working range of cranes, the crane-to-vehicle assignment is not fixed. Moreover, cranes cannot overtake each other because they share a common pathway (i.e., rail-track) for their horizontal movement. This is reflected in crane scheduling models by the so-called non-crossing constraints (Boysen et al., 2017). After the handling operation, the AGV again enters the bypass lane and moves to the exit gate.

It should be noted that, in current container ports, AGVs are mainly used inside the terminal to transfer containers between the seaside and landside (Gharehgozli et al., 2017). In the more complicated external transportation environment, manual driving is still needed. However, the automated guided trucks under consideration are assumed to support two driving modes, i.e., automated mode and manual driving mode. Vehicles switch from manual driving mode to the self-driving mode at the entrance gate, where drivers leave the cockpit and wait in the lounge area. When vehicles complete the container pickup/drop-off operations and arrive at the exit gate, they switch back to the manual driving mode. As vehicles are self-driving inside the terminal, for simplicity, AGVs are used to represent automated guided trucks in this paper.

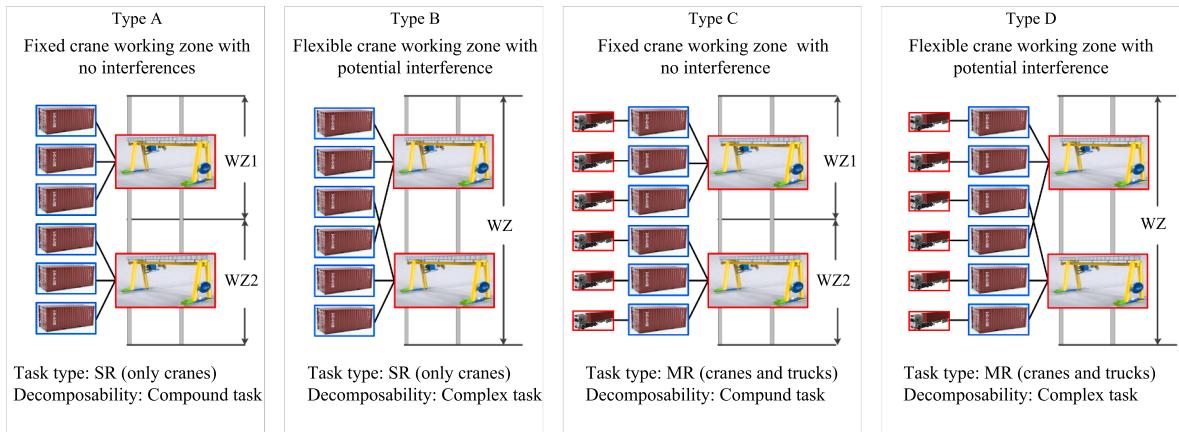


Fig. 2. Illustrations of four types of MRTA for container hub scheduling problems. WZ, WZ1, and WZ2 indicate working zones for cranes and the solid lines indicate potential valid allocations of tasks to robots.

One of the most critical building blocks in CACSP is how to assign a set of robots (AGVs and cranes) to a set of tasks (i.e., container-handling operations) while considering the inter-robot constraints, in a way that optimizes the overall system performance. The problem is called the multi-robot task allocation (MRTA) problem (Khamis et al., 2015). We adopt two taxonomies (Gerkey and Mataric, 2004; Zlot and Stentz, 2006) in the field of MRTA problem to classify the typical container hub scheduling problems illustrated in Fig. 2.

First, according to Gerkey and (2004), the container hub scheduling can be categorized by the type of the task, such as a single-robot (SR)-type task versus a multi-robot (MR)-type task. In particular, each SR task is allocated to a single robot, whereas each MR task requires multiple robots. In Fig. 2(A) and 2(B), there is only one crane robot to be scheduled as an SR task for each container handling task, whereas MR tasks in Fig. 2(C) and (D) involve scheduling of both AGV and crane robots. Furthermore, based on the decomposability of tasks, we can divide the handling resources scheduling problems into *compound task* problems and *complex task* problems (Zlot and Stentz, 2006). There is only one single fixed way of decomposing a task into multiple subtasks for a compound task problem, whereas there are multiple possible ways of decomposing a complex task. Type A and C problems in Fig. 2 belong to the compound task type, where the working area of each crane is fixed. In comparison, the working areas of cranes in Fig. 2(B) and 2(D) are flexible so that the scheduling algorithm needs to handle complex task problems.

In this study, we focus on CACSP with a flexible-crane-working range, as MR-Complex task (Type D) in Fig. 2(D). In this problem, each container handling task belongs to the MR task which needs the cooperation of two types of robots (i.e., automated cranes and AGVs). Besides, the working range of each crane is flexible, thus the vehicle-to-crane assignment becomes a complex task in MRS.

The main difficulties lie in deciding how handling tasks are assigned to cranes and how each crane schedules its tasks and further how AGVs and cranes coordinate each other considering the inter-crane and inter-vehicle interferences. Specifically, first, for MR tasks, we need to schedule AGVs and cranes to synchronize their paths in both space and time dimensions. Second, interactions among multiple robots need to be considered during the task-to-robot assignment, on-line task scheduling, and robot path finding process. In particular, AGVs need to satisfy the gate capacity constraint during the inspection process. Besides, the difficult non-crossing constraints (Boysen et al., 2017) of cranes will be addressed using a discretized space-time network in Section 2.1.2. Similar modeling issues are faced also in unmanned warehouses for e-commerce and logistics companies (He and Chang, 2010; Boysen and Stephan, 2016).

1.2. Literature review

To date, equipment scheduling of container terminals is a well-researched domain and has been extensively studied in the past few decades due to the rapid development of container transport in railway and marine transportation system. Previous research paid a lot of attention to improve the operational efficiency of quay cranes (QC), yard cranes (YC) and container vehicles, but most of it dealt with the resources separately

QC scheduling studies generally deal with the determination for the handling sequence of vessel bays for quay cranes assigned to a vessel. The seminal work by Daganzo (1989) studied both static and dynamic quay crane scheduling problems for multiple container vessels. Peterkofsky and Daganzo (1990) further formulated the crane scheduling problem as a resource-constrained project scheduling problem (RCPSP). Along this research line, a great number of scholars included inter-crane interference constraints in the problem, we refer the studies by Kim and Park (2004), Sammarra et al. (2007), Lee et al. (2008), Bierwirth and Meisel (2009), Han et al. (2010), Kaveshgar et al. (2012), Legato et al. (2012), Meisel and Bierwirth (2013), Unsal and Oguz (2013), Chen et al. (2014), Fu et al. (2014) and Kizilay et al. (2018). YC scheduling focuses on obtaining an optimal plan for yard cranes stacking and retrieving containers in container yards. Related research can be found in Kozan and Preston (1999), Ng (2005), Ng and Mak (2005), Jung and Kim (2006), Lee et al. (2007), Cao et al. (2008), Li et al. (2009), He et al. (2010), Boysen and Fliedner (2010), Park et al. (2010), Yan

et al. (2011), Chang et al. (2011), Gharehgozli et al. (2014), Kellner and Boysen (2015), and Wu et al. (2015). In particular, Vis and Carlo (2010), Carlo and Martínez-Acevedo (2015), and Gharehgozli et al. (2017) addressed the YC scheduling problem in automated container terminals. For container vehicle scheduling problems, most of the scholars focused on seeking the optimized assignment and routing plan for vehicles (Lee et al., 2009; Nguyen and Kim, 2009; Angeloudis and Bell, 2010; He et al., 2013). Interested readers can also find many excellent review papers on these topics, including Bierwirth and Meisel (2010, 2015), Carlo et al. (2014, 2015), Luo et al. (2011), and Boysen et al. (2017). Basically, container handling tasks need the cooperation of multiple resources (e.g., cranes and vehicles). If they are not synchronized effectively, extra wait times may lead to productivity loss. Therefore, scheduling the handling resources separately could lead to sub-optimal solutions (Zhang et al., 2019).

The coordination of cranes and vehicles is an emerging class of operations research problems arising in automated container terminals. Currently, there are limited papers about the integrated scheduling of cranes and vehicles. To our best knowledge, Bish et al. (2001), Bish (2003), Li and Vairakatarakis (2004), Tang et al. (2014) studied the joint scheduling of QCs and yard vehicles. The synchronization of YCs and yard trucks was considered by Cao et al. (2010). Chen et al. (2007), Chen et al. (2013), He et al. (2015), Zeng and Yang (2009), and Yan et al. (2019) have studied the integrated scheduling of QC, YC and yard vehicles. Existing literature sheds some light on the important research direction in this paper. However, the inter-crane and inter-vehicle constraints have not been carefully exploited. Most of the research assumed that multiple yard cranes do not interfere with each other, while Tang et al. (2014) and He et al. (2015) have introduced the non-crossing and safety distance constraints for QCs.

From the perspective of the automation community (Bragin et al., 2018), two key problems arising in an automated container hub are the crane-to-AGV assignment and crane/AGV scheduling. In order to solve the problems efficiently, some research results in the field of MRTA problems can be adopted. Basically, MRTA scheduling algorithms are based on two different approaches: market-based and optimization-based (Khamis et al., 2015). The efficiency of the optimization-based approach is highly dependent on the nature and complexity of the deterministic or stochastic MRTA problem (Badreldin et al., 2013). In contrast, the market-based approach uses the concept of the auction and has several desirable features, such as high efficiency in satisfying the objective function, robustness, and scalability. Therefore, the latter approach has gained increasing attention within the robotics research community (Tang and Parker, 2007).

We summarize the recent studies related to the crane scheduling problems in Table 1 based on the problem taxonomy mentioned in Section 1.1 and the solution algorithm classification criteria.

1.3. Overview and potential contributions

Based on Table 1, we found the following major observations:

Most studies scheduled cranes and trucks separately and did not pay particular to the coordination of multi-robot for container terminals in both space and time dimensions. The output results from most of the existing optimization models are the sequences of handling tasks. If space-time trajectories of the handling equipment cannot be obtained, it is not sufficient to support the precise multi-robot dispatching in an automated hub. In addition, due to the complexity in the coordination of multiple handling equipment, existing integrated scheduling models are usually solved by heuristics methods (e.g., genetic algorithm, tabu search, simulated annealing, etc.) without providing a systematic way to measure the lower bound and the solution quality. In this study, we propose to address several modeling challenges in CACSP in the context of container hub automation. The potential contributions are as follows:

- (1) A time-discretized space-time network modeling framework is introduced to formulate the CACSP problem with various MR tasks. The coordination between cranes and AGVs is rigorously described using a set of coupled network-based multi-commodity flow formulations. By doing so, the resulting integer programming model could allow us to find an optimal space-time path, as well as the crane-AGV assignment results.
- (2) We use a market-based Alternating Direction Method of Multipliers (ADMM) dual decomposition approach to solve the CACSP with complicated inter-robot constraints (i.e., non-crossing and gate capacity constraints). The original problem can be decomposed into a set of crane-specific and vehicle-specific least-cost path problems that have standard computationally efficient algorithms available on large-scale networks. Based on the unique rolling update scheme of ADMM, the complex coupling constraints can be effectively handled by sequentially updating the space-time path of each robot, with an optimality gap estimation.
- (3) In a rolling horizon based real-time scheduling framework, we capture the real-world impacts of the disturbances caused by stochastic features in the traffic congestion and the probabilistic operation times (e.g. crane handling, gate inspection).

The remainder of this paper is structured as follows. The problem statement is described in Section 2. In Section 3, we establish a mathematical formulation for the CACSP problem and present the ADMM-based decomposition and linearization techniques. Section 4 introduces the detailed ADMM-based solution procedures and the rolling horizon synchronization framework. In Section 5, the connection and potential extensions of the ADMM-based framework are discussed. Section 6 contains a discussion on the results of numerical experiments.

2. Problem statement and crane and vehicle space-time network construction

In this paper, the scheduling problem with the MR-Complex task (which is marked as “Type D” in Fig. 2) is addressed considering the inter-robot constraints. In particular, the inter-robot constraints specifically refer to the non-crossing constraint and the gate

Table 1

Summary of problem characteristics, modeling approaches, and solution algorithms for crane scheduling problems.

Publication	Crane type	Modeling approach	Solution algorithm	Optimization type	Objective	Real-time	SR/MR	Decomposability
Bierwirth and Meisel (2009)	QC	MIP	BBH	Hybrid	Min makespan	No	SR	Complex
Bish et al. (2001)	QC	ILP	H	Deterministic	Min makespan	No	MR	Compound
Bish et al. (2003)	QC	ILP	H	Hybrid	Min makespan	No	MR	Complex
Boysen and Fliedner (2010)	Rail-YC	ILP	DP	Deterministic	Min makespan	No	SR	Complex
Cao et al. (2008)	Port-YC	ILP	H	Hybrid	Min makespan	No	SR	Complex
Cao et al. (2010)	Port -YC	MIP	BD	Deterministic	Min makespan	No	MR	Compound
Carlo and Martínez-Acevedo (2015)	Port -YC	MIP	BB&SIM	Hybrid	Min makespan	No	SR	Complex
Chang et al. (2011)	Port -YC	ILP	GA&SIM	Population-based	Min delay	No	SR	Complex
Chen et al. (2007)	QC & Port-YC	MIP	TS	Hybrid	Min makespan	No	MR	Complex
Chen et al. (2013)	QC & Port-YC	CP	H	Hybrid	Min makespan	No	MR	Compound
Chen et al. (2014)	QC	MIP	SS	Deterministic	Min makespan	No	SR	Complex
Fu et al. (2014)	QC	ILP	GA	Population-based	Min completion time	No	SR	Complex
Gharehgozli et al. (2014)	Port -YC	MIP	LNS	Trajectory-based	Min makespan	No	SR	Complex
Gharehgozli et al. (2017)	Port -YC	SIM	H	Hybrid	Min makespan & waiting time	No	SR	Complex
Han et al. (2010)	QC	MIP	GA	Population-based	Min makespan	No	SR	Complex
He et al. (2010)	Port -YC	OP	GA, H	Hybrid	Min makespan	No	SR	Compound
He et al. (2015)	QC & Port-YC	MIP	GA, PSO	Population-based	Min delay & energy consumption	No	MR	Complex
Jung and Kim (2006)	Port-YC	MIP	GA, SA	Stochastic	Min makespan	No	SR	Complex
Kaveshgar et al. (2012)	QC	MIP	GA	Population-based	Min makespan	No	SR	Complex
Kellner and Boysen (2015)	Rail-YC	MIP	SA	Trajectory based	Min makespan	No	SR	Complex
Kim and Park (2004)	QC	MIP	BBH	Hybrid	Min makespan	No	SR	Complex
Kizilay et al. (2018)	QC	CP	SS	Deterministic	Min makespan	No	SR	Complex
Lee et al. (2007)	Port-YC	ILP	SA	Trajectory-based	Min makespan	No	SR	Complex
Lee et al. (2008)	QC	MIP	GA	Population-based	Min makespan	No	SR	Complex
Legato et al. (2012)	QC	MIP	BBH	Hybrid	Min makespan	No	SR	Complex
Li and Vairaktarakis (2004)	QC	ILP	H	Hybrid	Min makespan	No	MR	Compound
Li et al. (2009)	Port-YC	MIP	SS, H	Hybrid	Min delay	No	SR	Complex
Meisel and Bierwirth (2013)	QC	MIP	BBH	Hybrid	Min makespan	No	SR	Complex
Ng (2005)	Port-YC	ILP	DPH	Hybrid	Min completion time	No	SR	Complex
Ng and Mak (2005)	Port-YC	MIP	BB	Deterministic	Min waiting time	No	SR	Compound
Park et al. (2010)	Port-YC	MIP	H	Stochastic	Min delay	Yes	SR	Complex
Sammarra et al. (2007)	QC	MIP	TS	Hybrid	Min makespan & completion time	No	SR	Complex
Tang et al. (2014)	QC	MIP	PSO	Population-based	Min makespan	No	MR	Complex
Unsal and Ogun (2013)	QC	CP	LNS	Trajectory-based	Min makespan	No	SR	Complex
Vis and Carlo (2010)	Port-YC	MIP	SA	Trajectory based	Min makespan	No	SR	Compound
Wu et al. (2015)	Port-YC	MIP	H	Stochastic	Min delay	Yes	SR	Complex
Yan et al. (2018)	Rail-YC	MIP	GA	Population-based	Min makespan & waiting time	No	MR	Compound
Zeng and Yang (2009)	QC & Port-YC	MIP	GA & SIM	Population-based	Min makespan	No	MR	Compound
This paper	Rail & Port-YC	ILP	ADMM	Market-based	Min total turn time	Yes	MR	Complex

QC—Quay crane, GC—Gantry crane, CP—Constraint programming, GM—Graphical model, ILP—Integer linear programming, MIP—Mixed integer programming, OP—Objective programming, STN—Space-time network, AA—Approximation algorithm, ADMM—Alternating Direction Method of Multipliers-based dual decomposition, BB—Branch-and-bound, BD—Benders' decomposition, BBH—Branch-and-bound based heuristic, DP—Dynamic programming, DPH—Dynamic programming-based heuristic, GA—Genetic algorithm, H—Heuristics, LNS—Large neighborhood search, PSO—Particle swarm optimization, SA—Simulated annealing, SIM—Simulation, SS—Standard solver, TS—Tabu search

capacity constraint. To give a clear description of the CACSP, in this section, we first present the detailed problem statement, assumptions, and notations used in this paper. Subsequently, the construction methods for the crane spacetime network (CSTN) and the vehicle space-time network (VSTN) are presented.

2.1. Problem statement

The CACSP problem in an automated container terminal is formally stated as follows. Given (a) the individual AGV demand with specific origin $o(v)$, destination $d(v)$, and arrival time $AT(v)$ for each vehicle $v(v \in V)$ at a container terminal; (b) the initial and final

Table 2

Sets, subscripts, parameters, and variables for the model formulation.

Subscripts	Definitions
i, j, i', j'	Index of nodes, $i, j, i', j' \in N$.
(i, j)	Index of links, $(i, j) \in L$.
t, t', τ, τ'	Index of time intervals, $t, t', \tau, \tau' = 1, 2, \dots, T$.
(i, t)	Index of vertices in the space-time network, $(i, t) \in V_{tr}, V_{cr}$.
(i, j, t, t')	Index of arcs in the space-time network, $(i, j, t, t') \in E_{tr}, E_{cr}$.
v	Index of vehicles, $v \in V$.
c, c'	Index of cranes, $c, c' \in C$.
Sets	
T	Length of the planning time horizon.
N	Set of nodes, $N = N_{tr} \cup N_{cr}$.
N_{cr}	Set of crane nodes, $N_{cr} = N_{cr}^{ph} \cup N_{cr}^{pi} \cup N_{cr}^{dr}$, where N_{cr}^{py} , N_{cr}^{pi} , and N_{cr}^{dr} denote the sets of crane physical nodes, dummy pickup nodes, and dummy drop-off nodes, respectively.
N_{tr}	Set of vehicle nodes, $N_{tr} = N_{tr}^{ga} \cup N_{tr}^{pa} \cup N_{tr}^{ha} \cup N_{tr}^{du}$, where N_{tr}^{ga} , N_{tr}^{pa} , N_{tr}^{ha} , and N_{tr}^{du} denote the sets of vehicle gate nodes, parking nodes, handling nodes, and dummy origin and dummy destination nodes, respectively. In addition, $N_{tr}^{ga} = N_{tr}^{ga_en} \cup N_{tr}^{ga_ex}$, where $N_{tr}^{ga_en}$ and $N_{tr}^{ga_ex}$ denote the set of gate entrance nodes and gate exit nodes, respectively.
L	Set of links, $L = L_{tr} \cup L_{cr}$.
L_{cr}	Set of crane links, $L_{cr} = L_{cr}^m \cup L_{cr}^h \cup L_{cr}^r$, where L_{cr}^m , L_{cr}^h , and L_{cr}^r denote the sets of moving links, handling links, and recovery links, respectively.
L_{tr}	Set of vehicle links, $L_{tr} = L_{tr}^m \cup L_{tr}^g \cup L_{tr}^h \cup L_{tr}^d$, where L_{tr}^m , L_{tr}^g , L_{tr}^h , and L_{tr}^d denote the sets of moving links, gate inspection links, handling links, and dummy links, respectively.
V_{tr}	Set of vertices in the VSTN.
V_{cr}	Set of vertices in the CSTN.
E_{tr}	Set of arcs in the VSTN, $E_{tr} = E_{tr}^w \cup E_{tr}^m \cup E_{tr}^g \cup E_{tr}^h \cup E_{tr}^d$, where E_{tr}^w , E_{tr}^m , E_{tr}^g , E_{tr}^h , and E_{tr}^d denote the sets of waiting arcs, moving arcs, gate inspection arcs, handling (loading/unloading) arcs, and dummy arcs, respectively.
E_{cr}	Set of arcs in the CSTN, $E_{cr} = E_{cr}^w \cup E_{cr}^m \cup E_{cr}^h \cup E_{cr}^r$, where E_{cr}^w , E_{cr}^m , E_{cr}^h , and E_{cr}^r denote the sets of waiting, moving, handling (loading/unloading), and recovery arcs, respectively.
V	Set of vehicles.
C	Set of cranes.
$\Phi(i, j, t, t')$	Set of crane moving arcs that are incompatible with space-time arc (i, j, t, t') owing to the crane non-crossing constraint, i.e., if arc (i, j, t, t') is visited by crane c , crane c' is $\varphi(c)$ is not permitted to use any moving arcs in the set $\Phi(i, j, t, t')$.
$\varphi(c)$	Set of cranes that are adjacent to crane c .
$\aleph(i, j, \tau, \tau')$	Set of gate inspection arcs that need to be constrained by the gate capacity, i.e., the total number of vehicles using the gate inspection arcs in this set should be limited by constant $Lnum(i, j)$, within the time range $(\tau - tg + 1, \tau)$ at the gate entrance node $i \in N_{tr}^{ga_en}$ and time range $(\tau - tg + 1, \tau)$ at gate exit node $j \in N_{tr}^{ga_ex}$.
Parameters	
$o(c)$	Origin node of crane c .
$d(c)$	Destination node of crane c .
$o(v)$	Origin node of vehicle v .
$d(v)$	Destination node of vehicle v .
$AT(v)$	Arrival time of vehicle v at the entrance gate.
$cost_{i,j,t,t'}^v$	Travel time cost of vehicle v on arc (i, j, t, t') .
tg	Average gate inspection time of a vehicle.
$Lnum(i, j)$	The total number of the truck lanes at the gates, $(i, j) \in L_{tr}^g$.
ω, ρ, σ	Non-negative penalty parameters in ADMM associated with the vehicle flow and crane flow coupling constraint, gate capacity constraint, and non-crossing constraint for crane moving, respectively.
$\pi_{i,j,t,t'}, \lambda_{i,j,t,t'}, \mu_{i,j,t,t',c}$	Lagrangian multipliers associated with the vehicle flow and crane flow coupling constraint, gate capacity constraint, and non-crossing constraint for crane moving, respectively.
Variables	
$x_{i,j,t,t'}^v$	0–1 space-time arc selection variable for the vehicle; = 1 if vehicle v is assigned on the space-time arc (i, j, t, t') ; = 0 otherwise.
$y_{i,j,t,t'}^c$	0–1 space-time arc selection variable for the crane; = 1 if crane c is assigned on the space-time arc (i, j, t, t') ; = 0 otherwise.

space-time states of each crane with specific origin $o(c)$ at time $t = 0$ and destination $d(c)$ at time $t = T$ within the container yard; and (c) gate capacity $Lnum(i, j)$ associated with the average gate inspection time tg , we aim to optimally assign the AGV loading and unloading tasks to the cranes and coordinate the space-time path of cranes and AGVs with complex inter-robot constraints such that the total AGV turn time in the container terminal is minimized. **Table 2** lists the notations of general sets, subscripts, parameters, and variables.

2.2. Assumptions and notations

Without loss of generality, some key assumptions are made for CACSP to facilitate the modeling process.

- (1) A gantry crane can only serve one AGV at any time.
- (2) Each container to be handled has a known origin (destination) location in the yard. As the cranes and AGVs need to stay at the slot where the target container is located during the handling operation, the parking slot for each AGV is predetermined.
- (3) The reshuffling operation of containers is not considered in this study. If a reshuffling scheduling is needed, the time cost of the reshuffling operation can be viewed as a component of the total handling time.
- (4) As we focus on the scheduling level of the crane and AGV operations, the potential collisions among vehicle movements are ignored. This is a reasonable assumption in typical crane scheduling studies (Bierwirth and Meisel, 2015; Boysen et al., 2017). A similar method for handling the non-crossing constraint in section 2.3.1 can be applied to deal with the collision avoidance issue at a finer resolution if needed.

2.3. Space-time network construction

2.3.1. Space-time network for cranes

As described above, to ease the orientation for crane and vehicle operations in a practical situation, a container yard is usually spatially decomposed into several slots along its horizontal spread. The size of a slot needs to consider the length of a container truck or a railway wagon. For instance, according to Boysen et al. (2010), the length of a typical work section is 14 m in the German transshipment yards, which covers the size of a railway car of type “Lgs580” carrying one 40-foot container (FEU) or two 20-foot containers (TEU).

Based on this practice, we first discretize the crane-working space (as a virtualized resource set) into several working sections represented by a set of crane physical nodes N_{cr}^{ph} . Then, for each node $i \in N_{cr}^{ph}$, we design two related dummy pickup nodes $(i + 1) \in N_{cr}^{pi}$ and drop-off nodes $(i + 2) \in N_{cr}^{dr}$, each with different process time costs. Next, we construct CSTN, where V_{cr} is the set of crane vertices and E_{cr} is the set of crane arcs. The set of arcs E_{cr} can be classified into the following five types:

- (1) Waiting arcs $(i, i, t, t + 1)$ are constructed from vertex (i, t) to vertex $(i, t + 1)$ at node $i \in N_{cr}^{ph}$, which represent the crane-waiting process at a physical node i .
- (2) Moving arcs (i, j, t, t') are extended from link (i, j) where $i, j \in N_{cr}^{ph}$. The length of the time interval $(t' - t)$ indicates the deterministic moving time for a crane on link (i, j) .
- (3) Pickup arcs (i, j, t, t') are extended from link (i, j) where $i \in N_{cr}^{ph}$ and $j \in N_{cr}^{pi}$, which represent the container-lifting process from one physical node i at time t to one dummy pickup node j at time t' .
- (4) Drop-off arcs (i, j, t, t') are similar to the pickup arcs, except that $j \in N_{cr}^{dr}$.
- (5) Recovery arcs (i, j, t, t') are constructed from vertex (i, t) to vertex (j, t') , where $i \in N_{cr}^{pi} \cup N_{cr}^{dr}$ and $j \in N_{cr}^{ph}$, which represent the spreader position reset process of the crane after one handling job. The length of the time interval $(t' - t)$ can be regarded as the average time cost of the spreader reset, $(i, j, t, t') \in E_{cr}^r$.

An illustrative example with two crane space-time trajectories is presented in Fig. 3. In particular, crane 1 departs from slot 1 and arrives at slot 2 by traveling through moving arc (110, 120, 0, 2). After waiting at slot 2 for 2 min, crane 1 performs the drop-off (i.e., AGV loading) task through drop-off arc (120, 121, 4, 8) and returns to node 120 at slot 2 using recovery arc (121, 120, 8, 9). Moreover, crane 2 departs from slot 4 and moves to slot 3 to perform the pickup (i.e., AGV unloading) task. Note that the proposed CSTN is applicable to the terminals where the container handover position is along the pathway of cranes. However, it can be easily modified to describe the terminal with other layout types (e.g., the layout with handling points located in front of container blocks) by adjusting positions of pickup and drop-off arcs.

To model the non-crossing constraints for cranes in CSTN, the horizontal movements of cranes are described by moving arcs. A set of incompatible moving arcs for each type of arc has been constructed, as shown in Fig. 4(a) and (b). When any space-time arc is occupied by a crane, all the potential conflicting arcs (i.e., incompatible moving arcs) of this arc need to be prohibited for adjacent cranes in set $\varphi(c)$.

2.3.2. Space-time network for AGVs

The AGV operations can also be described using the time-discretized space-time network. The sets of gate nodes N_{tr}^{ga} , parking nodes N_{tr}^{pa} , handling nodes N_{tr}^{ha} , and dummy origin and dummy destination nodes N_{tr}^{du} are constructed. It should be noted that we borrowed the method of Caprara et al. (2002) and Shang et al. (2019), where each train station is split into two dummy stations, two sets of gate nodes are constructed in VSTN, i.e., $N_{tr}^{ga} = N_{tr}^{ga_en} \cup N_{tr}^{ga_ex}$, where $N_{tr}^{ga_en}$ and $N_{tr}^{ga_ex}$ denote the set of gate entrance nodes and the set of gate exit nodes, respectively. Moreover, the set of gate inspection links (i.e., L_{tr}^g) is naturally formed between the gate entrance nodes and the gate exit nodes to describe the vehicle inspection times. In addition, each vehicle $v \in V$ is assumed to have a given origin dummy node $o(v) \in N_{tr}^{du}$ at time $t = 0$ and a destination dummy node $d(v) \in N_{tr}^{du}$ at the end time of the planning time horizon $t = T$.

Based on the sets of nodes N_{tr} and links L_{tr} , the VSTN can be constructed with sets of vertices V_{tr} and arcs E_{tr} . The arcs set can also

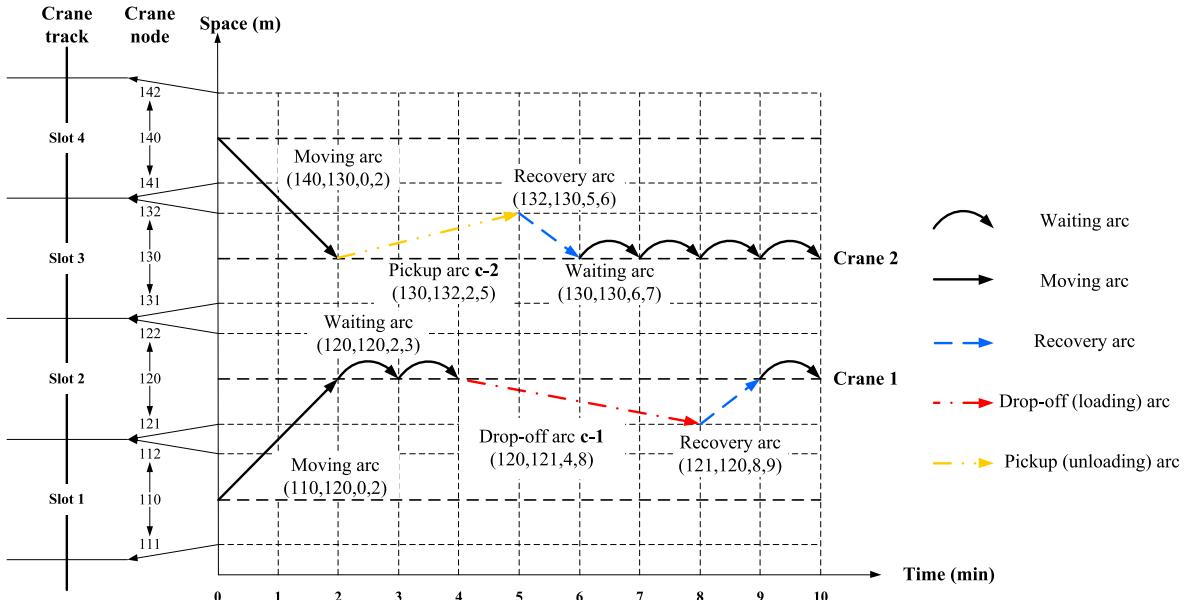


Fig. 3. An illustrative example of the crane space-time trajectories in CSTN.

be divided into five types: the waiting arcs, moving arcs, gate inspection arcs, pickup arcs, and drop-off arcs. An illustrative example involving two AGVs' space-time trajectories is presented in Fig. 5. Note that the drop-off arc **t-1** and pickup arc **t-2** are coupled with the drop-off arc **c-1** and pickup arc **c-2** in Fig. 3, respectively. It represents that AGV 1 is assigned to crane 1 for the loading task, and AGV 2 is assigned to crane 2 for the unloading task.

3. Model reformulation and decomposition for applying ADMM in CACSP

3.1. Multi-robot coordination and scheduling model

CACSP can be formulated as a coupled network-based multi-commodity flow model in VSTN and CSTN with several embedded constraints. The proposed integer linear programming (ILP) model for the crane–AGV coordination and scheduling problem is stated as follows.

■ Original formulation M1:

Objective function:

$$\text{MinZ} = \sum_{v \in V} \sum_{(i,j,t,t') \in E_{tr}} (\text{cost}_{i,j,t,t'}^v \times x_{i,j,t,t'}^v) \quad (1)$$

Subject to:

Vehicle flow balance constraint:

$$\sum_{i,t:(i,j,t,t') \in E_{tr}} x_{i,j,t,t'}^v - \sum_{i,t:(j,i,t,t') \in E_{tr}} x_{j,i,t,t'}^v = \begin{cases} 1 & i = o(v), t = AT(v) \\ -1 & j = d(v), t = T \\ 0 & \text{otherwise} \end{cases}, \forall v \in V \quad (2)$$

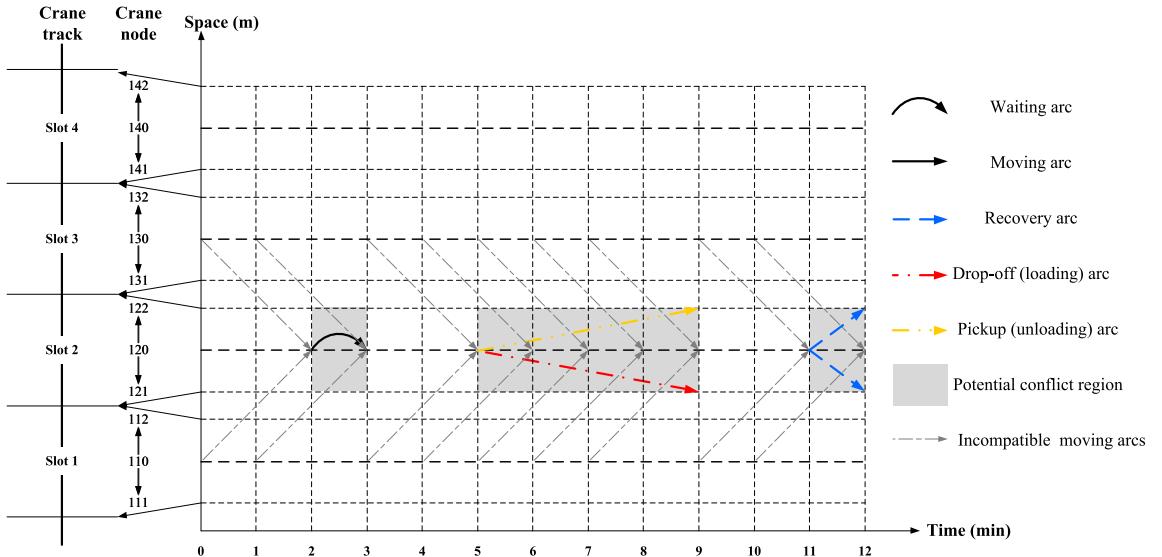
Crane flow balance constraint:

$$\sum_{i,t:(i,j,t,t') \in E_{cr}} y_{i,j,t,t'}^c - \sum_{i,t:(j,i,t,t') \in E_{cr}} y_{j,i,t,t'}^c = \begin{cases} 1 & i = o(c), t = 1 \\ -1 & j = d(c), t = T, \forall c \in C \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

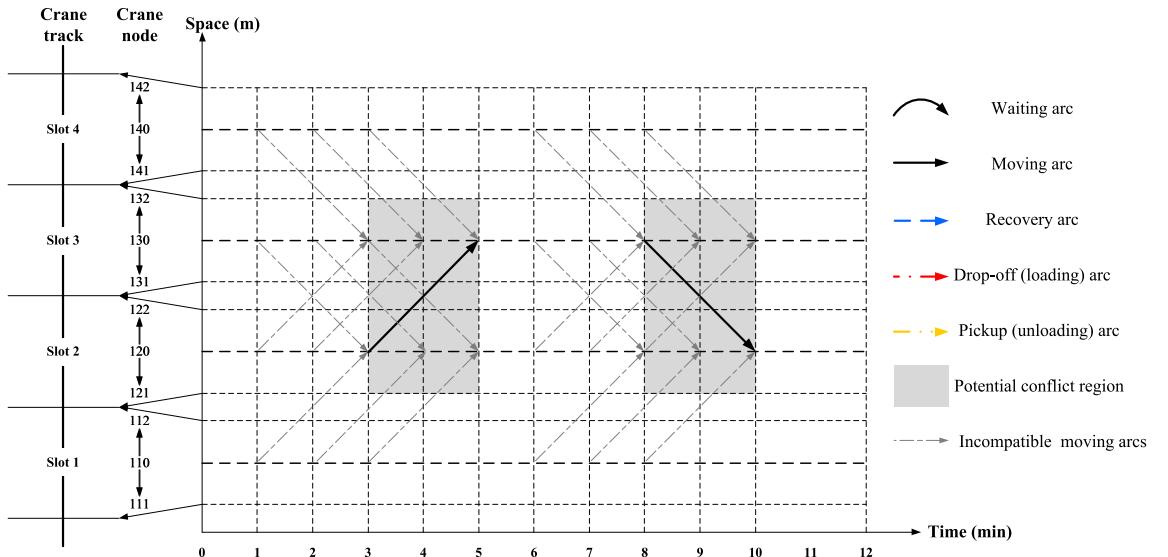
Vehicle flow and crane flow coupling constraint:

$$\sum_{v \in V} x_{i,j,t,t'}^v = \sum_{c \in C} y_{i,j,t,t'}^c, \forall (i, j, t, t') \in E_{tr}^h \quad (4)$$

Non-crossing constraint for crane moving:



(a) Potential conflict regions and incompatible moving arcs of the waiting, drop-off, pickup and recovery arcs



(b) Potential conflict regions and incompatible moving arcs of moving arcs

Fig. 4. Illustration of potential conflict regions and incompatible moving arcs in CSTN.

$$y_{i,j,t,t'}^c + \sum_{(i',j',\tau,\tau') \in \Phi(i,j,t,t')} \sum_{c' \in \varphi(c)} y_{i',j',\tau,\tau'}^{c'} \leq 1, \forall c \in C, (i,j,t,t') \in E_{cr} \quad (5)$$

Gate capacity constraint:

$$\sum_{v \in V} \sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v \leq Lnum(i,j), \forall (i,j,\tau,\tau') \in E_{tr}^g \quad (6)$$

Binary definitional constraints:

$$x_{i,j,t,t'}^v \in \{0, 1\}, \forall v \in V, (i,j,t,t') \in E_{tr} \quad (7)$$

$$y_{i,j,t,t'}^c \in \{0, 1\}, \forall c \in C, (i,j,t,t') \in E_{cr} \quad (8)$$

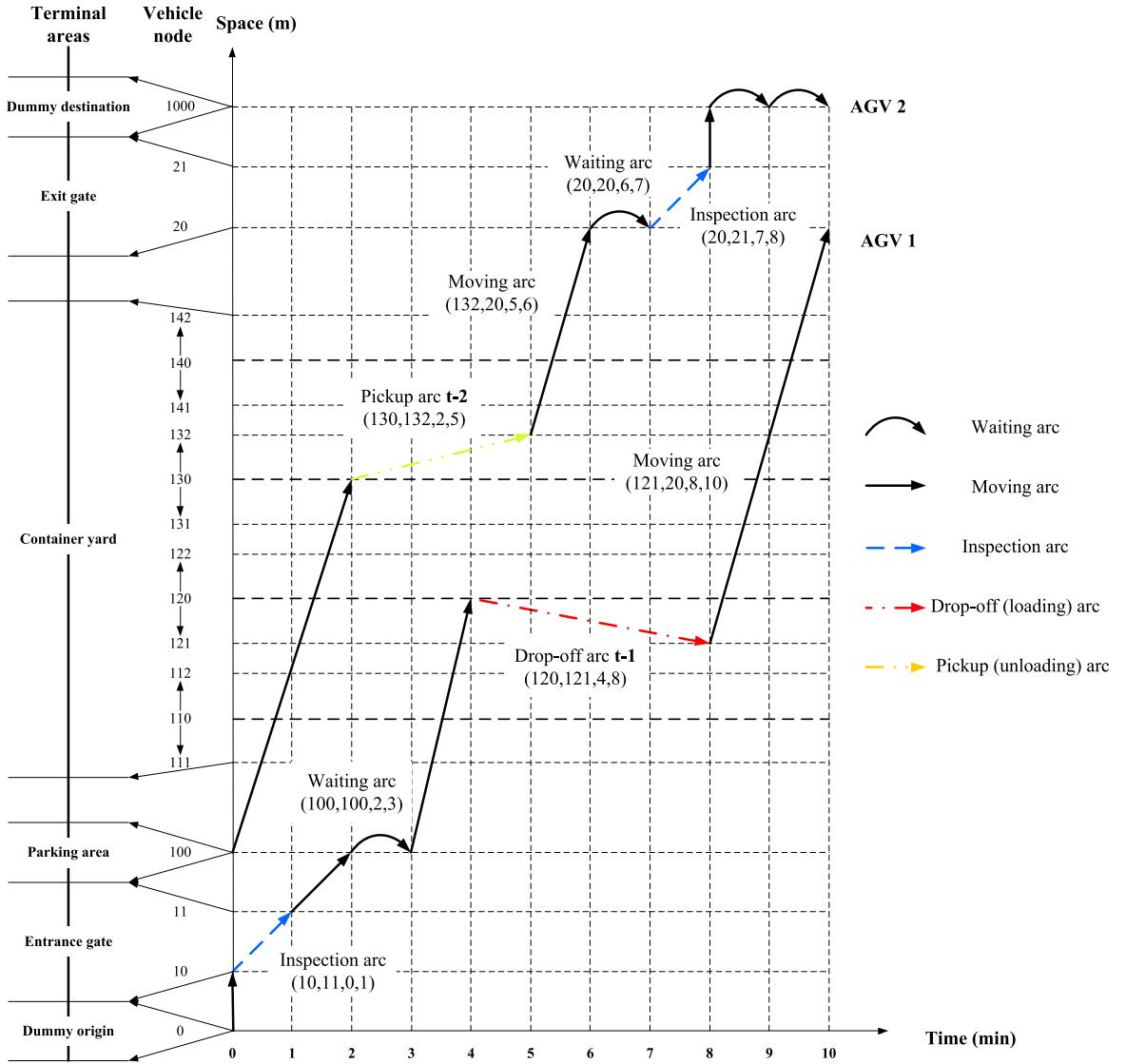


Fig. 5. An illustrative example of vehicle space-time trajectories in VSTN.

The objective function in Eq. (1) aims to minimize the total vehicle turn time in the container terminal, where the parameter $\text{cost}_{i,j,t,t'}^v$ specifically refers to the travel time for AGV v using arc (i, j, t, t') . Eq. (2) is the standard flow balance constraint for the vehicles in VSTN, and a similar flow balance relationship for the cranes in CSTN is specified in Eq. (3). Eq. (4) is the coupling constraint between AGV and crane flows, which ensures that the vehicle flow needs to be assigned on the handling (i.e., pickup/drop-off) arcs selected by the cranes. The non-crossing constraint for cranes is given in Eq. (5). In particular, Eq. (5) enforces that if crane c occupies space-time arc (i, j, t, t') , all the incompatible moving arcs in set $\Phi(i, j, t, t')$ cannot be used by any other cranes $c' \in \varphi(c)$. The illustrative examples in Fig. 4(a) and (b) show how to enumerate the set of incompatible moving arcs of a certain crane arc in CSTN. Eq. (6) defines the gate capacity constraint, which ensures that the number of vehicles undergoing the gate inspection task should not be greater than the total number of truck lanes of terminal gates. In addition, a comparison of model M1 with the classical activity-based RCPSP model in Pritsker et al. (1969) is given in Appendix B, which shows that the proposed STN model is able to handle more complicated requirements for CACSP, particularly the non-crossing constraint and the detailed moving process of cranes.

3.2. Complex task decomposition and alternating direction method of multipliers

In the proposed integer linear programming (ILP) model M1, the vehicle flow and crane flow coupling constraint in Eq. (4) links two sets of variables $X = \{x_{i,j,t,t'}^v\}$ and $Y = \{y_{i,j,t,t'}^c\}$. Therefore, it is important to recognize that model M1 can be viewed as a coupled combination of two sub-problems, where each sub-problem can be more efficiently solved. The first sub-problem is a multi-

commodity flow model for the vehicles with side gate capacity constraint (6), and the second sub-problem is another multi-commodity flow model for the cranes with side non-crossing constraint (5). One important point of this observation is that coupling constraint (4) includes both variables X and Y , which makes the problem particularly difficult to solve, and leads to further computational challenges (Ahuja et al., 1993; Tong et al., 2017).

To reduce complexities in various optimization problems, several useful dual decomposition approaches have been widely used, including the Lagrangian relaxation (Fisher, 1981), augmented Lagrangian relaxation (Fortin and Glowinski, 2000), and ADMM (Gabay and Mercier, 1976; Bertsekas, 1999; Boyd, et al., 2010). The augmented Lagrangian relaxation introduces an extra quadratic penalty term into the objective function. Compared with the augmented Lagrangian relaxation, variables can be updated sequentially in a block-by-block manner in ADMM. As a result, ADMM has the inherent advantages of breaking symmetry and strong convexity while maintaining good problem decomposition structure.

ADMM has been effectively applied to problems of the multi-robot system (MRS) to provide a distributed strategy for multi-robot controlling (Van Parys and Pipeleers, 2016; Abdelkader, 2018; Mota et al., 2013). In particular, several researchers (Bento et al., 2013; Zheng et al., 2017; Park et al., 2018) have adopted ADMM for multi-agent space trajectory planning, where complex tasks are distributed among the agents by the ADMM considering the inter-robot constraints (e.g., collision avoidance). Compared with previous studies, we apply ADMM to decompose the original problem into sub-problems in a space–time network, where the space–time trajectories for cranes and vehicles are optimized. The iteratively updating scheme of ADMM is able to facilitate the exchange of information between robots to complete complex MR tasks.

Specifically, we first dualize the coupling constraint (4) after introducing Lagrangian multipliers $\pi_{i,j,t,t'}$ and ADMM penalty parameter ω . Furthermore, to efficiently solve the multi-commodity flow models for the cranes and AGVs, non-crossing constraint (5) and gate capacity constraint (6) are also dualized into the objective function with Lagrangian multipliers $\mu_{i,j,t,t',c}$ and $\lambda_{i,j,t,t'}$ and ADMM penalty parameter σ and ρ , respectively. In addition, to introduce the quadratic terms into the objective function when using ADMM, inequality constraints (5) and (6) need to be transformed into equality constrains by introducing two continuous and non-negative slack variables $s_{i,j,t,t',c}^y \in [0, 1]$ and $s_{i,j,t,t'}^x \in [0, Lnum(i, j)]$ in Eq. (9). Therefore, the proposed original model M1 is transformed into a new model M2, as follows.

■ Dualized formulation M2:

Objective function:

$$\begin{aligned} \text{Min}L = & \sum_{v \in V} \sum_{(i,j,t,t') \in E_{tr}} \{ \text{cost}_{i,j,t,t'}^v \times x_{i,j,t,t'}^v \} + \\ & \sum_{(i,j,t,t') \in E_{tr}^h} \left\{ \pi_{i,j,t,t'} \times \left[\sum_{v \in V} x_{i,j,t,t'}^v - \sum_{c \in C} y_{i,j,t,t'}^c \right] \right\} + \sum_{(i,j,t,t') \in E_{tr}^h} \left\{ \omega / 2 \times \left[\sum_{v \in V} x_{i,j,t,t'}^v - \sum_{c \in C} y_{i,j,t,t'}^c \right]^2 \right\} + \\ & \sum_{(i,j,\tau,\tau') \in E_{tr}^g} \left\{ \lambda_{i,j,\tau,\tau'} \times \left[\sum_{v \in V} \sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v - Lnum(i, j) \right] \right\} + \\ & \sum_{(i,j,\tau,\tau') \in E_{tr}^g} \left\{ \rho / 2 \times \left[\sum_{v \in V} \sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v + s_{i,j,\tau,\tau'}^x - Lnum(i, j) \right]^2 \right\} + \\ & \sum_{(i,j,t,t') \in E_{cr}} \sum_{c \in C} \mu_{i,j,t,t',c} \left[y_{i,j,t,t'}^c + \sum_{(i,j',\tau,\tau') \in \Phi(i,j,t,t')} \sum_{c' \in \varphi(c)} y_{i,j',\tau,\tau'}^{c'} - 1 \right] + \\ & \sum_{(i,j,t,t') \in E_{cr}} \sum_{c \in C} \sigma / 2 \left[y_{i,j,t,t'}^c + \sum_{(i,j',\tau,\tau') \in \Phi(i,j,t,t')} \sum_{c' \in \varphi(c)} y_{i,j',\tau,\tau'}^{c'} + s_{i,j,t,t',c}^y - 1 \right]^2 \end{aligned} \quad (9)$$

Subject to:

Constraints (2), (3), (7), (8), (10), (11), and (12).

$$\pi_{i,j,t,t'} \in R, \forall (i, j, t, t') \in E_{tr}^h \quad (10)$$

$$\lambda_{i,j,t,t'} \geq 0, \forall (i, j, \tau, \tau') \in E_{tr}^g \quad (11)$$

$$\mu_{i,j,t,t',c} \geq 0, \forall c \in C, (i, j, t, t') \in E_{cr} \quad (12)$$

It can be shown that after dualizing coupling constraint (4), non-crossing constraint (5), and gate capacity constraint (6), model M2 can be decomposed into two sup-problems P_x and P_y corresponding to the vehicle routing sub-problem and the crane scheduling sub-problem, respectively. Note that owing to coupling constraint (4) being an equality constraint, the value of Lagrangian multiplier $\pi_{i,j,t,t'}$ is not constrained, whereas the values of Lagrangian multipliers $\lambda_{i,j,\tau,\tau'}$ and $\mu_{i,j,t,t',c}$ are restricted to be greater than or equal to 0 (namely, constraints (11) and (12)). Note that compared with the Lagrangian relaxation method (see Fisher (1981) and Mahmoudi and Zhou (2016)), the value of Lagrangian multiplier $\pi_{i,j,t,t'}^{k+1}$ at the $(k + 1)^{th}$ iteration is updated in Eq. (13), where the step sizes are replaced by the penalty parameters and k represents the iteration number. The update scheme for multipliers $\lambda_{i,j,\tau,\tau'}$ and $\mu_{i,j,t,t',c}$ is similar to that of $\pi_{i,j,t,t'}$.

$$\pi_{i,j,t,t'}^{k+1} = \pi_{i,j,t,t'}^k + \omega \left[\sum_{v \in V} x_{i,j,t,t'}^v - \sum_{c \in C} y_{i,j,t,t'}^c \right] \quad (13)$$

However, the quadratic penalty term $\left[\sum_{v \in V} x_{i,j,t,t'}^v - \sum_{c \in C} y_{i,j,t,t'}^c \right]^2$ in Eq. (9) still couples variables $x_{i,j,t,t'}^v$ and $y_{i,j,t,t'}^c$ together, and thus it is difficult to solve sub-problems P_x and P_y independently. On the other hand, if each crane and each AGV are treated as individual blocks, the rolling update scheme of ADMM allows us to optimize the variables of a block with the variables in the rest of blocks being fixed (Boyd et al., 2010; Yao et al., 2019; Zhang et al., 2019). In particular, when sub-problem P_x is being optimized, the space-time paths of the cranes are fixed, and thus the term $\sum_{c \in C} y_{i,j,t,t'}^c$ is stable and can be treated as a constant term $sum_y = \sum_{c \in C} y_{i,j,t,t'}^c$. Similarly, when sub-problem P_y is being optimized, term $\sum_{v \in V} x_{i,j,t,t'}^v$ can also be treated as a constant term $sum_x = \sum_{v \in V} x_{i,j,t,t'}^v$. Thus, we can decompose M2 into vehicle-specific and crane-specific sub-problems as shown in models **M3** and **M4**, respectively.

■ Vehicle sub-problem M3: P_x in VSTN

Objective function:

$$\begin{aligned} MinL_X = & \sum_{v \in V} \sum_{(i,j,t,t') \in E_{tr}^v} \{cost_{i,j,t,t'}^v \times x_{i,j,t,t'}^v\} + \sum_{(i,j,t,t') \in E_{tr}^h} \{\pi_{i,j,t,t'} \times \sum_v x_{i,j,t,t'}^v\} + \\ & \sum_{(i,j,t,t') \in E_{tr}^h} \left\{ \omega / 2 \times \left[\sum_{v \in V} x_{i,j,t,t'}^v - sum_y \right]^2 \right\} + \sum_{(i,j,\tau,\tau') \in E_{tr}^g} \left\{ \lambda_{i,j,\tau,\tau'} \times \sum_{(i,j,t,t') \in \mathbb{N}(i,j,\tau,\tau')} \sum_v x_{i,j,t,t'}^v \right\} + \\ & \sum_{(i,j,\tau,\tau') \in E_{tr}^g} \left\{ \rho / 2 \times \left[\sum_{v \in V} \sum_{(i,j,t,t') \in \mathbb{N}(i,j,\tau,\tau')} x_{i,j,t,t'}^v + s_{i,j,\tau,\tau'}^x - Lnum(i,j) \right]^2 \right\} \end{aligned} \quad (14)$$

Subject to:

Constraints (2), (7), (10), and (11).

To apply the block coordinate descent method to solve each vehicle-specific sub-problem, we need to address an important property of ADMM; it is possible to linearize the quadratic penalty term in ADMM when only binary variables are involved in sub-problems with linear and separable structure. Interested readers can refer to Yao et al. (2019) and Zhang et al. (2019) for the handling of binary variables in the ADMM for vehicle routing and train timetabling applications. It should be noted that compared with their studies, where only one type of agent is considered during the path searching process, we handle two types of dynamically moving agents along with the difficult coupling constraints between them.

As mentioned before, owing to the rolling update scheme of ADMM, only variables $x_{i,j,t,t'}^v$ related to one specific vehicle v are optimized each time, with all the variables related to other AGVs and cranes being fixed. If the current vehicle to be optimized is denoted by v , the rest of the vehicles belong to set $V' = V \setminus \{v\}$. We use constant $m_v(i, j, t, t')$ and its short notation m_v to represent the number of vehicles in set V' that use the handling space-time arc $(i, j, t, t') \in E_{tr}^h$, i.e., $m_v = \sum_{v' \in V'} x_{v'}(i, j, t, t')$. Similarly, we use constant n_v to represent the number of vehicles in set V' that occupy the space-time resources in set $\mathbb{N}(i, j, \tau, \tau') \in E_{tr}^g$, i.e., $n_v = \sum_{v' \in V'} \sum_{(i,j,t,t') \in \mathbb{N}(i,j,\tau,\tau')} x_{v'}(i, j, t, t')$. By substituting m_v and n_v into the quadratic terms of Eq. (14), they can be linearized as shown in Eqs. (15) and (16), respectively.

It should be noted that owing to $x_{i,j,t,t'}^v$ being a binary variable, the transformation from the quadratic term $(x_{i,j,t,t'}^v)^2$ into $x_{i,j,t,t'}^v$ in Eq. (15) is quite straightforward. In addition, because vehicle v can use at most one of the space-time arcs (i, j, t, t') in set $\mathbb{N}(i, j, \tau, \tau')$, the quadratic term $\left[\sum_{(i,j,t,t') \in \mathbb{N}(i,j,\tau,\tau')} x_{i,j,t,t'}^v \right]^2$ can also be directly linearized as $\sum_{(i,j,t,t') \in \mathbb{N}(i,j,\tau,\tau')} x_{i,j,t,t'}^v$ in Eq. (16).

$$\begin{aligned} \left[\sum_{v \in V} x_{i,j,t,t'}^v - sum_y \right]^2 &= [x_{i,j,t,t'}^v + m_v - sum_y]^2 = (x_{i,j,t,t'}^v)^2 + 2x_{i,j,t,t'}^v(m_v - sum_y) + (m_v - sum_y)^2 \\ &= x_{i,j,t,t'}^v(2m_v - 2sum_y + 1) + (m_v - sum_y)^2 \end{aligned} \quad (15)$$

$$\begin{aligned} &\left[\sum_{v \in V} \sum_{(i,j,\tau,\tau') \in \mathbb{N}(i,j,\tau,\tau')} x_{i,j,t,t'}^v + s_{i,j,\tau,\tau'}^x - Lnum(i,j) \right]^2 \\ &= \sum_{(i,j,t,t') \in \mathbb{N}(i,j,\tau,\tau')} x_{i,j,t,t'}^v(2n_v - 2Lnum(i,j) + 1) + (n_v + s_{i,j,\tau,\tau'}^x - Lnum(i,j))^2 \end{aligned} \quad (16)$$

Eq. (16) contains slackness variable $s_{i,j,\tau,\tau'}^x$ whose value needs to be determined. We first express Eq. (16) as a function of $s_{i,j,\tau,\tau'}^x$ in Eq. (17). As $L_{s_{i,j,\tau,\tau'}^x}$ is a quadratic function and $s_{i,j,\tau,\tau'}^x$ is a continuous variable, the optimal value of $s_{i,j,\tau,\tau'}^x$ to minimize the value of $L_{s_{i,j,\tau,\tau'}^x}$ can be obtained at $s_{i,j,\tau,\tau'}^{x,*}$ in Eq. (18). Then two cases can be discussed to determine the feasible value of $s_{i,j,\tau,\tau'}^x$ according to the value of n_v .

(1) Case 1: $n_v \geq Lnum(i,j)$

When $n_v \geq Lnum(i,j)$, the theoretical value of $s_{i,j,\tau,\tau'}^{x,*}$ will be less than or equal to 0. Because $s_{i,j,\tau,\tau'}^x \in [0, Lnum(i,j)]$, the optimum value of $s_{i,j,\tau,\tau'}^x$ is equal to 0. Then the quadratic terms in Eq. (16) can be transformed into

$$\sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v (2n_v - 2Lnum(i,j) + 1) + (n_v + s_{i,j,\tau,\tau'}^x - Lnum(i,j))^2.$$

(2) Case 2: $0 \leq n_v \leq Lnum(i,j) - 1$

In this case, the theoretical value of $s_{i,j,\tau,\tau'}^x$ is within the range $[0, Lnum(i,j)]$; therefore, the feasible optimum value $s_{i,j,\tau,\tau'}^x = s_{i,j,\tau,\tau'}^{x*}$. Then the value of the quadratic terms in Eq. (16) is equal to 0.

$$\begin{aligned} L_s^x &= s_{i,j,\tau,\tau'}^x - 2s_{i,j,\tau,\tau'}^x \left(\sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v + n_v - Lnum(i,j) \right) + \\ &\left[\sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v (2n_v - 2Lnum(i,j) + 1) + (n_v - Lnum(i,j))^2 \right] \end{aligned} \quad (17)$$

$$s_{i,j,\tau,\tau'}^{x*} = Lnum(i,j) - \sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v - n_v \quad (18)$$

$$\begin{aligned} &\left[\sum_{v \in V} \sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v + s_{i,j,\tau,\tau'}^x - Lnum(i,j) \right]^2 \\ &= \begin{cases} \sum_{(i,j,t,t') \in N(i,j,\tau,\tau')} x_{i,j,t,t'}^v (2n_v - 2Lnum(i,j) + 1) + (n_v + s_{i,j,\tau,\tau'}^x - Lnum(i,j))^2 & n_v \geq Lnum(i,j) \\ 0 & 0 \leq n_v \leq Lnum(i,j) \end{cases} \end{aligned} \quad (19)$$

Based on the above discussion, the linearized results for the two cases are expressed in Eq. (19). After linearizing the two quadratic terms in Eq. (14), we can generate the linearized space-time arc usage costs for each vehicle v in sub-problem $P_x(v)$, which is denoted as model **M3-L**.

■ **Linearized vehicle sub-problem M3-L:** Sub-problem $P_x(v)$ for each vehicle v

$$\text{Min } L_X^v = \sum_{v \in V} \sum_{(i,j,t,t') \in E_{tr}} \{\hat{c}_{i,j,t,t'}^v \times x_{i,j,t,t'}^v\} + Q_X \quad (20)$$

where $\hat{c}_{i,j,t,t'}^v$ is the usage cost of the space-time arc $(i, j, t, t') \in E_{tr}$ for vehicle v in VSTN, and Q_X denotes the items irrelevant to $x_{i,j,t,t'}^v$. The value of $\hat{c}_{i,j,t,t'}^v$ is given in Eq. (21) according to different types of space-time arcs. In particular, the two cases in Eq. (19) are unified by using term $\max\{0, \frac{\rho}{2}(2n_v - 2Lum(i,j) + 1)\}$.

$$\hat{c}_{i,j,t,t'}^v = \begin{cases} cost_{i,j,t,t'} + \pi_{i,j,t,t'} + \frac{\omega}{2} \times (2m_v - 2sum_y + 1) & \forall (i, j, t, t') \in E_{tr}^h \\ cost_{i,j,t,t'} + \sum_{(i,j,\tau,\tau') \in E_{tr}^g, (i,j,t,t') \in N(i,j,\tau,\tau')} [\lambda_{i,j,\tau,\tau'} + \max\{0, \frac{\rho}{2}(2n_v - 2Lnum(i,j) + 1)\}] & \forall (i, j, t, t') \in E_{tr}^g \\ cost_{i,j,t,t'} \text{ otherwise} & \end{cases} \quad (21)$$

■ **Crane sub-problem M4:** P_y in CSTN

Objective function:

$$\begin{aligned} \text{Min } L_Y &= - \sum_{(i,j,t,t') \in E_{cr}^h} \left\{ \pi_{i,j,t,t'} \times \sum_{c \in C} y_{i,j,t,t'}^c \right\} + \sum_{(i,j,t,t') \in E_{cr}^h} \left[\frac{\omega}{2} \times \left(sum_x - \sum_{c \in C} y_{i,j,t,t'}^c \right)^2 \right] + \\ &\sum_{(i,j,t,t') \in E_{cr}} \sum_{c \in C} \mu_{i,j,t,t',c} \left[y_{i,j,t,t'}^c + \sum_{(i,j,\tau,\tau') \in \Phi(i,j,t,t')} \sum_{c' \in \varphi(c)} y_{i,j,\tau,\tau'}^{c'} - 1 \right] + \\ &\sum_{(i,j,t,t') \in E_{cr}} \sum_{c \in C} \sigma / 2 \left[y_{i,j,t,t'}^c + \sum_{(i,j,\tau,\tau') \in \Phi(i,j,t,t')} \sum_{c' \in \varphi(c)} y_{i,j,\tau,\tau'}^{c'} + s_{i,j,t,t',c}^y - 1 \right]^2 \end{aligned} \quad (22)$$

Subject to:

Constraints (3), (8), (10), and (12).

Similar to the linearization process of the quadratic penalty terms in **M3**, we introduce constants $m_c = \sum_{c \in C} y_{i,j,t,t'}^c$ and $n_c = \sum_{(i,j,\tau,\tau') \in \Phi(i,j,t,t')} \sum_{c' \in \varphi(c)} y_{i,j,\tau,\tau'}^{c'}$ into Eq. (22) to linearize **M4**. The linearized sub-problem $P_y(c)$ for each crane $c \in C$ is proposed in **M4-L**.

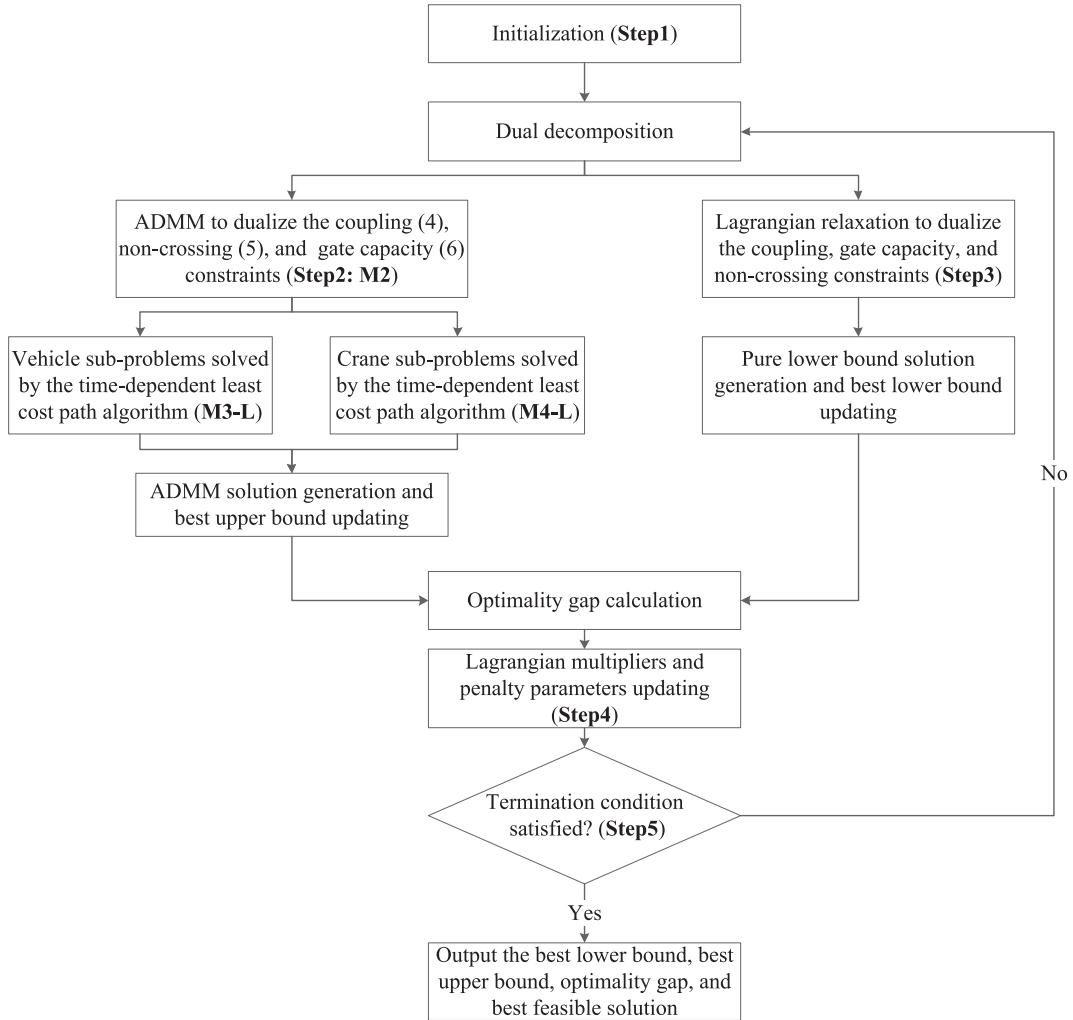


Fig. 6. ADMM-based solution framework for iteratively updating the best lower and upper bounds.

■ *Linearized crane sub-problem M4-L:* Sub-problem $P_y(c)$ for each crane c

$$\text{Min}L_Y^c = \sum_{(i,j,t,t') \in E_{cr}} \sum_{c \in C} \hat{d}_{i,j,t,t'}^c y_{i,j,t,t'}^c + Q_Y \quad (23)$$

where $\hat{d}_{i,j,t,t'}^c$ denotes the usage cost of crane c on space-time arc $(i, j, t, t') \in E_{cr}$, and Q_Y represents the independent items of $y_{i,j,t,t'}^c$.

The original model **M1** is decomposed into a set of vehicle-specific and crane-specific least-cost path searching problems: the vehicle routing sub-problem in model **M3-L** with the space-time arc usage cost $\hat{d}_{i,j,t,t'}^c$, and the crane routing sub-problem in model **M4-L** with the space-time arc usage cost $\hat{d}_{i,j,t,t'}^c$.

4. ADMM- based solution procedure and real-time scheme using the beam-search methodology

4.1. ADMM-based solution procedure

The conceptual illustration of the ADMM-based solution framework is presented in Fig. 6. Based on the above ADMM-based framework, the ADMM-based solution procedure is adopted to solve CACSP. The detailed solution procedure of ADMM is provided in

Algorithm 1.**Algorithm 1: iterative algorithms steps of the ADMM-based solution procedure for two types of robots**

```

// Step 1: Initialization
Set iteration number  $k = 0$ 
Initialize the three Lagrangian multipliers  $\pi_{i,j,t,i}^0 = 0$ ,  $\lambda_{i,j,t,i}^0 = 0$ ,  $\mu_{i,j,t,i,c}^0 = 0$  and the three penalty parameters  $\omega^0 = 10$ ,  $\rho^0 = 2$ ,  $\sigma^0 = 2$ ;
Initialize the best lower bound  $Best\_LB^* = -\infty$  and the best upper bound  $Best\_UB^* = +\infty$ 
Initialize the incremental amount of penalty parameters  $\delta$ .
//Step 2: ADMM solution generation and best upper bound value updating
//Step 2.1: Call the forward dynamic programming algorithm over vehicles
For each vehicle  $v \in V$  do
    Update arc usage costs  $\hat{c}_{i,j,t,i}^v$  by Eq. (21)
    Find the space-time least-cost path for vehicle  $v$  (minimize  $L_X^v$ ) by calling the forward dynamic programming algorithm (Ziliaskopoulos and Mahmassani, 1993; Chabini, 1998);
End for each vehicle  $v \in V$ 
// Step 2.2: Call the forward dynamic programming algorithm over cranes
For each crane  $c \in C$  do
    Update arc usage costs  $\hat{d}_{i,j,t,i}^c$ 
    Find the space-time least-cost path for crane  $c$  (minimize  $L_Y^c$ ) by calling the forward dynamic programming algorithm;
End for each crane  $c \in C$ 
//Step 2.3: Compute  $UB^k$  and update the best upper bound  $Best\_UB^*$ 
If the current ADMM solution is feasible
    Calculate the upper bound  $UB^k$  at the  $k^{th}$  iteration and update the best upper bound  $Best\_UB^* = \min\{Best\_UB^*, UB^k\}$ 
End If
//Step 3: Pure lower bound solution generation and best lower bound updating
//Step 3.1: Generate pure lower bound solution by using Lagrangian relaxation method (Shang et al., 2018)
For each vehicle  $v \in V$  do
    Update arc usage cost  $\bar{c}_{i,j,t,i}^v$  by

$$\bar{c}_{i,j,t,i}^v = \begin{cases} cost_{i,j,t,i} + \pi_{i,j,t,i} & \forall (i, j, t, i) \in E_{tr}^h \\ cost_{i,j,t,i} + \sum_{(i,j,\tau,i') \in E_{tr}^g, (i,j,t,i') \in N(i,j,\tau,i)} \lambda_{i,j,\tau,i'} & \forall (i, j, t, i) \in E_{tr}^g \\ cost_{i,j,t,i} & \text{otherwise} \end{cases}$$

    Find the space-time least-cost path for vehicle  $v$  by calling the forward dynamic programming algorithm;
End for each vehicle  $v \in V$ 
For each crane  $c \in C$  do
    Update arc usage cost  $\bar{d}_{i,j,t,i}^c$  by

$$\bar{d}_{i,j,t,i}^c = \begin{cases} -\pi_{i,j,t,i} + \mu_{i,j,t,i,c} & \forall (i, j, t, i) \in E_{cr}^h \\ \mu_{i,j,t,i,c} & \text{otherwise} \end{cases}$$

    Find the space-time least-cost path for crane  $c$  by calling the forward dynamic programming algorithm;
End for each crane  $c \in C$ 
//Step 3.2: Compute  $LB^k$  and update the best lower bound  $Best\_LB^*$ 
Calculate the upper bound  $LB^k$  at the  $k^{th}$  iteration and update the best lower bound  $Best\_LB^* = \max\{Best\_LB^*, LB^k\}$ 
//Step 4: Lagrangian multipliers and penalty parameter values updating
//Step 4.1: Update the Lagrangian multipliers
Update Lagrangian multipliers  $\pi_{i,j,t,i}^{k+1}$ ,  $\lambda_{i,j,t,i}^{k+1}$ , and  $\mu_{i,j,t,i,c}^{k+1}$ .
//Step 4.2: Update the penalty parameter values
Increase or retain penalty parameter values  $\omega^{k+1}$ ,  $\rho^{k+1}$ , and  $\sigma^{k+1}$  according to the following condition (Yao et al. 2019)

$$violation_\omega = \sum_{(i,j,t,i) \in E_{tr}^h} \left( \sum_v x_{i,j,t,i}^v - \sum_c y_{i,j,t,i}^c \right)$$


$$\omega^{k+1} = \begin{cases} \omega^k + \delta & \|\mathbf{violation}_\omega^{k+1}\|_2^2 \geq \alpha \|\mathbf{violation}_\omega^k\|_2^2 \\ \omega^0 & \|\mathbf{violation}_\omega^{k+1}\|_2^2 = 0 \\ \omega^k & \text{otherwise} \end{cases}$$

where  $violation_\omega$  can be seen as the total violation value of coupling constraints (4). The suitable values of parameters  $\delta$  and  $\alpha$  are in ranges of [2, 10] and [0.1, 0.5] after observing the general performance of the ADMM-based solution method, respectively.
The detailed updating rules of penalty parameters  $\rho^{k+1}$  and  $\sigma^{k+1}$  are similar to that of penalty parameters  $\omega^{k+1}$ .
//Step 5: Evaluate solution quality and termination condition test
Compute the relative gap between  $Best\_LB^*$  and  $Best\_UB^*$  by equation  $\text{Gap} = \frac{Best\_UB^* - Best\_LB^*}{Best\_UB^*} \times 100\%$ ;
If the solution satisfies a convergence condition or  $k$  reaches the maximum iteration number  $k = k^{max}$ , terminate the algorithm and output the  $Best\_LB^*$  and  $Best\_UB^*$ ; otherwise,  $k = k + 1$ , and go back to Step 2.

```

Robots		Iteration k										Iteration $k+1$									
	v_1	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$
	v_2	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$
Vehicles	v_3	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$
\vdots	\vdots	\vdots	\vdots	...	\vdots	\vdots	\vdots	\vdots	\vdots	...	\vdots	\vdots	\vdots	...	\vdots	\vdots	\vdots	\vdots	\vdots	...	\vdots
	$v_{ V }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$
	c_1	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$
	c_2	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$
Cranes	c_3	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$
\vdots	\vdots	\vdots	\vdots	...	\vdots	\vdots	\vdots	\vdots	\vdots	...	\vdots	\vdots	\vdots	...	\vdots	\vdots	\vdots	\vdots	\vdots	...	\vdots
	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$	v_1	v_2	v_3	...	$v_{ V }$	c_1	c_2	c_3	...	$c_{ C }$

○ Vehicle/crane to be optimized at iteration $k+1$ [---] Vehicles/cranes remaining to be optimized at iteration $k+1$
[] Vehicles/cranes temporarily being fixed

Fig. 7. Rolling update scheme of ADMM for vehicles and cranes. O.

Adapted from Yao et al. 2019

The rolling update scheme of ADMM is presented in Fig. 7, where each row represents the least cost path updating of one robot each time. In each row, the robot to be optimized at the $(k+1)^{th}$ iteration is surrounded by the red circle, whereas the vehicles/cranes within the solid rectangle are the agents temporarily being fixed. The vehicles or cranes within the dashed rectangles are the agents remaining to be optimized at the $(k+1)^{th}$ iteration. By dynamically updating the space-time arc usage costs $\hat{d}_{i,j,t,t}^v$ and $\hat{d}_{i,j,t,t}^c$ according to the space-time paths of the fixed robots at each inner iteration, the symmetry between different robots can be broken (Boyd et al., 2010; Zhang et al., 2019).

4.2. Rolling horizon-based real-time synchronization framework

Because a long-time horizon needs to be considered in a 24×7 operational automated container hub, it is difficult to obtain an optimal or even a near-optimal static global solution. Inspired by Meng and Zhou (2011) and Zhan et al. (2016), who divided the whole train rescheduling problem into several optimization stages, with each stage needing to consider only train services in the coming period, a rolling horizon framework is utilized to decompose our CACSP. This approach helps in decreasing the problem complexity and achieves a real-time decision-making framework for the automated hub. The procedure of this framework is shown in

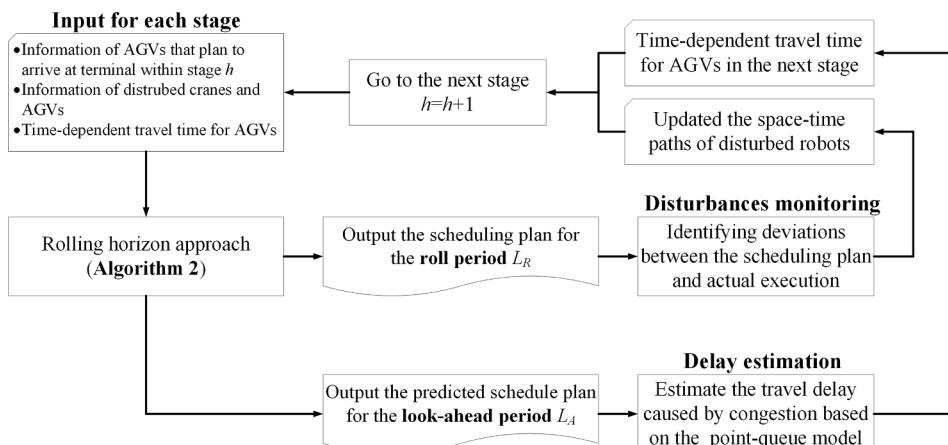


Fig. 8. The procedure of the rolling horizon framework.

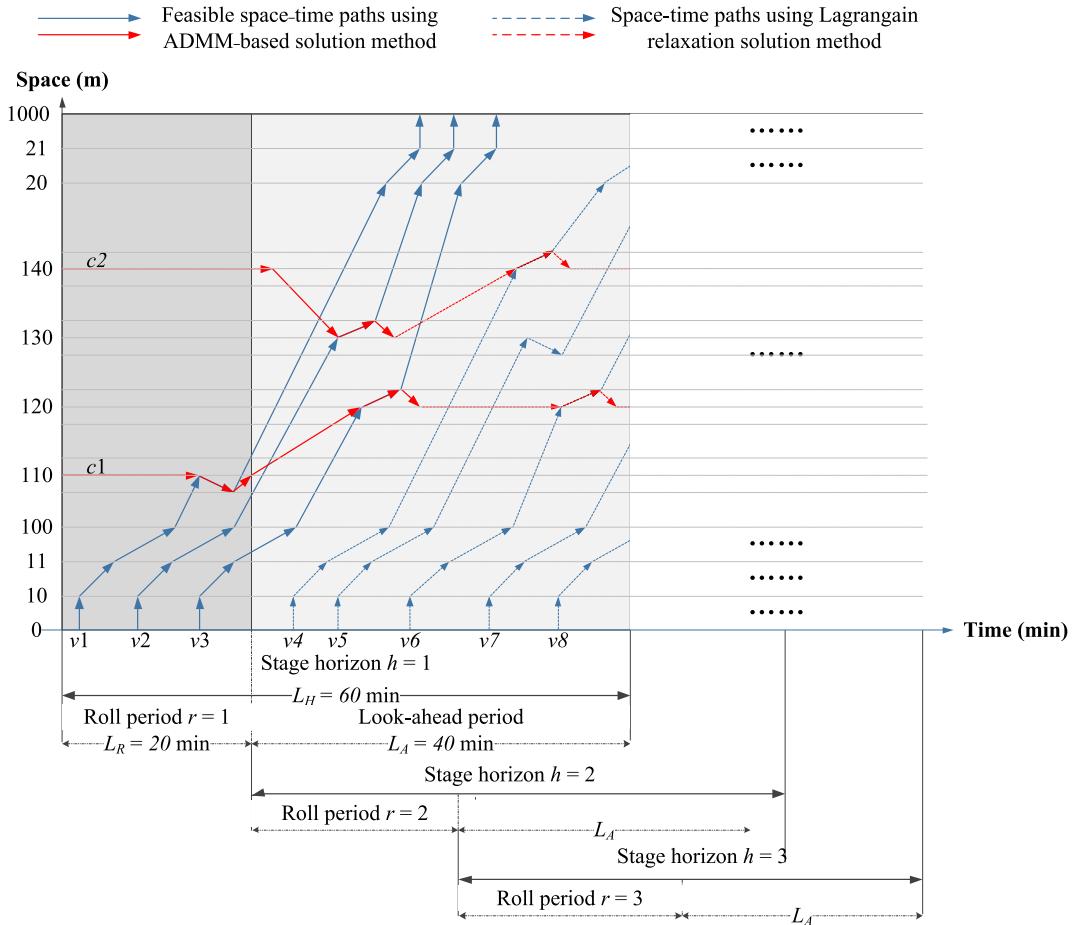


Fig. 9. Illustration of the rolling horizon approach in STN (blue and red lines indicate the space–time paths of AGVs and cranes, respectively). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Symbol definitions for the rolling horizon approach.

Symbol	Definition
h	Index of stage horizons, $h \in H$
r	Index of roll periods, $r \in R$
H	Set of stage horizons
R	Set of roll periods
L_H	Length of stage horizon (e.g., 60 min)
L_R	Length of roll period (e.g., 20 min)
L_A	Length of look-ahead period (e.g., 40 min)
t_h^s	Start time of stage horizon h
t_h^e	End time of stage horizon h
t_r^s	Start time of roll period r
t_r^e	End time of roll period r
m, n	Index of feasible solutions (e.g., space–time paths drawn by the solid line in Fig. 9) by the ADMM-based solution method, $m, n \in S^h$
S^h	Set of feasible solutions of stage horizon h , where $ S^h = \gamma$
$P_cost(m)$	Primal cost of feasible solution m
$D_cost(m)$	Dual cost estimation of feasible solution m (e.g., dual cost of space–time paths drawn by the dashed line in Fig. 9).
$T_cost(m)$	Total cost estimation of feasible solution m
w_p	Weight coefficient of primal cost
w_D	Weight coefficient of dual cost

Fig. 8. And **Fig. 9** illustrates the rolling horizon approach within the STN. The symbol definitions for the rolling horizon approach are listed in **Table 3**.

In this framework, the planning horizon is divided into several stages, each of which includes a roll period L_R and a look-ahead period L_A . In each stage horizon, only the active AGVs arriving at the terminal within the current stage are considered. At the beginning of stage h , the ADMM-based solution method is utilized to generate a feasible schedule plan for the roll period L_R according to the input data obtained at the beginning of stage h . The Lagrangian relaxation method is adopted to generate a predicted schedule for the look-ahead period L_A . At the end of the current roll period, a monitoring program is conducted to give feedback information of the disturbances. When a disturbance occurs, the information (i.e., space-time path) of the affected AGVs and cranes are updated. The updated information is utilized as the input data for the next stage. Moreover, with the aid of a simulation-based point queue model (see Appendix C), the time-dependent travel time for the next stage is estimated according to the predicted schedule plan for the look-ahead period L_A . Finally, the schedule plan for stage $h + 1$ is generated through the proposed algorithms.

To improve the efficiency and effectiveness of the optimization algorithm further, a beam-search method is also embedded into the rolling horizon approach. The beam search strategy is an adaptation of the branch-and-bound method in which only some nodes are evaluated in the search tree. At any level, only some of the promising nodes are kept for further branching and the remaining nodes are pruned permanently (Sabuncuoglu and Bayiz, 1999; Zhou and Zhong, 2007). In this framework, stage horizon h denotes the search stage, where only γ best promising feasible solutions are kept and parameter γ is regarded as the beam width. To find the γ best solutions in each stage, a combined ADMM-based and Lagrangian-based lower bound estimation search procedure is constructed. The solution procedure of the rolling horizon approach is given in **Algorithm 2**.

Algorithm 2: iterative algorithmic steps of rolling horizon approach using beam search method

// Step 1: Initialization

Initialize stage horizon index $h = 1$ and roll period index $r = 1$.

Initialize stage horizon length $L_H = 60$ min, roll period length $L_R = 20$ min, and look-ahead period length $L_A = 40$ min.

Initialize beam width $\gamma = 5$.

//Step 2: Feasible solution generation and cost estimation of stage horizon

For each feasible solution $n \in S^{h-1}$ do

//Step 2.1 Generate feasible solution m using the ADMM-based solution method

For each vehicle (e.g., v_1 , v_2 , and v_3 in **Fig. 9**) arriving at the terminal within roll period r (i.e., $v \in \{v|v \in V, t_r^S \leq AT(v) < t_r^E\}$) and each crane $c \in C$ do

Use **Algorithm 1** to find the feasible space-time least-cost paths for each vehicle and crane (e.g., the space-time paths drawn by solid lines in **Fig. 9**). Note that the existing space-time paths of the feasible solution in the last stage horizon $n \in S^{h-1}$ are considered based on the rolling update scheme of the ADMM method as shown in **Fig. 7**.

End for each vehicle $v \in \{v|v \in V, t_r^S \leq AT(v) < t_r^E\}$ and each crane $c \in C$

Calculate primal cost value $P_cost(m)$.

//Step 2.2 Estimate the dual cost of feasible solution m using the Lagrangian relaxation solution method

For each vehicle (e.g., $v_4 - v_8$ in **Fig. 9**) arriving at the terminal within the look-ahead period of stage horizon h (i.e., $v \in \{v|v \in V, t_r^E \leq AT(v) < t_h^E\}$ and crane $c \in C$ do

Use the Lagrangian relaxation-based solution method to find the space-time least-cost paths for each vehicle and crane (e.g., the space-time paths drawn by the dashed line in **Fig. 9**, and note that the primal feasibility of these paths is not guaranteed).

End for $v \in \{v|v \in V, t_r^E \leq AT(v) < t_h^E\}$ and each crane $c \in C$

Calculate dual cost estimation $D_cost(m)$

//Step 2.3 Calculate total estimation cost $T_cost(m)$ of feasible solution m :

$T_cost(m) = wp * P_cost(m) + wD * D_cost(m)$

where the typical values of wp and wD can be $wp = L_R/L_H$ and $wD = L_A/L_H$, respectively.

End for each feasible solution $n \in S^{h-1}$

//Step 3: Feasible solutions set S^h construction using the beam search strategy

Sort all feasible solutions of stage horizon h in the ascending order of $T_cost(m)$ and select the γ best solutions to construct the feasible solution set S^h of stage horizon h .

//Step 4: Algorithm termination condition

Terminate the algorithm if stage horizon h reaches the maximum stage horizon number $|H|$, otherwise, $h = h + 1$, $r = r + 1$, and go to Step 2.

5. Discussions in the context of MRTA

5.1. Connections and the proposed ADMM-based framework with MRTA

The proposed ADMM-based framework can provide some algorithmic insights for addressing general MRTA problems. The connections between the ADMM-based framework and common MRTA approaches, as well as the potential extensions considering the practical requirements, are summarized in **Fig. 10**.

Market-based optimization algorithms (Dias, 2004; Berhault et al., 2003; Koenig et al., 2007), which are mainly based on the concept of auctions, and optimization-based approaches (Atay and Bayazit, 2006; Jones et al., 2011; Wang et al., 2012) have been commonly applied to solve MRTA problems (Badreldin et al., 2013). In our framework, the ADMM-based decomposition method uses the iterative rolling update scheme to generate feasible upper bound solutions, where the cost of each space-time arc for each robot is updated based on the principles of market economy.

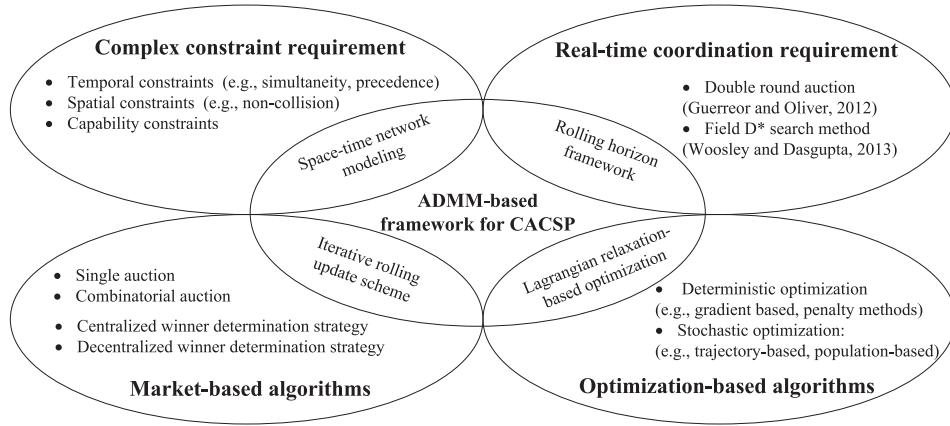


Fig. 10. Connections and potential extensions of proposed ADMM-based framework in the context of major requirements and optimization algorithms for general MRTA problems.

Practical difficulties in MRTA problems involve various types of complex constraints (such as capability, capacity, and ordering) and real-time optimization requirements. We use the space–time network to model the complex coupling constraints and inter-robot constraints (i.e., non-crossing and gate capacity constraints), and a rolling horizon framework is constructed to provide the on-line coordination scheduling solutions.

In summary, the ADMM-based framework has strong connections with the commonly used approaches and provides a reference for solving complex constraints for MRTA problems. Future extensions could be highlighted as follows:

- (1) Some advanced auction strategies (Koenig et al., 2007; Zlot and Stentz, 2006) and optimization-based methods (e.g., cutting plane method (Kallehauge et al., 2001; Kallehauge, 2008) can be integrated into the iterative rolling update scheme to update the Lagrangian multipliers more effectively.
- (2) Additional “state” dimension can be introduced into the CACSP model based on the extended time–space network where more types of complex constraints, such as temporal constraints (Dohn et al., 2009; Rasmussen et al., 2010), spatial constraints (Korsah, 2011), ordering constraints (Lemaire et al., 2004), etc., can also be considered in the multi-robot assignment and scheduling.

5.2. How to incorporate heuristic assignment rules from practical MRTA problems

The proposed CACSP model can also be transformed into an assignment-and-routing mapping model by a variable splitting

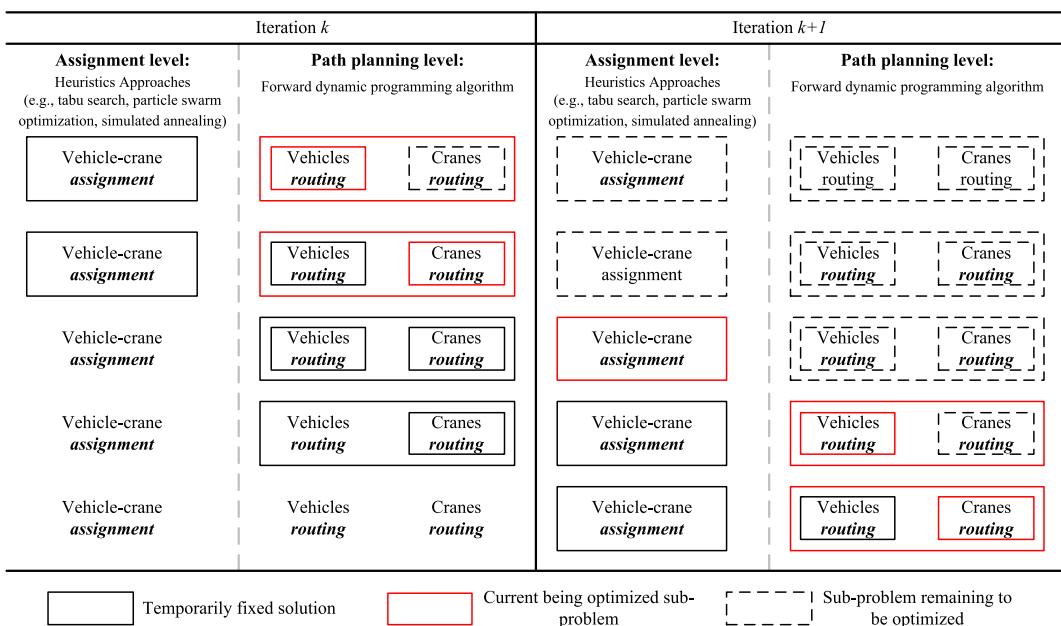


Fig. 11. Rolling update scheme in ADMM-based framework for assignment and routing problem.

technique (Fisher et al., 1997; Niu et al., 2018), where the consistency constraint between the semi-assignment and routing sub-problems can better help us to coordinate the tasks of assigning cranes to multiple vehicles.

These sub-problems can be solved alternately and iteratively in different decision levels as shown in Fig. 11. The results of the vehicle to crane assignment sub-problem can serve as guidance for routing sub-problems. Correspondingly, the routing solutions can also be used as feedback to the assignment layer. This framework across different decision levels also presents additional opportunities for re-interpreting and refining many efficient heuristics from primal and dual perspectives. In particular, some efficient heuristics methods, such as the Tabu search heuristic method (Díaz and Fernández, 2001; Chen et al., 2011), particle swarm optimization method (Tang et al., 2014), and simulated annealing method (Vis and Carlo, 2010), can be integrated to solve the assignment sub-problem.

5.3. How to model the AGV congestion and to estimate travel delay inside the terminal

Based on the proposed space–time model, the capacity of the key bottlenecks (e.g., parking area, truck lanes) can be explicitly modeled to capture the congestion caused by layout issues and limited capacity (Carlo et al., 2014). In particular, a subset of parking links L_{tr}^p and moving links L_{tr}^m can be constructed in the VSTN with given capacity $Cap(i, j)$. Then, a capacity constraint can be formulated as shown below.

5.3.1. Capacity constraint for parking area and truck lanes

$$A(i, j, t) - D(i, j, t) \leq Cap(i, j), \forall (i, j) \in L_{tr}^p \cup L_{tr}^m, \forall (t) \in T \quad (24)$$

$A(i, j, t)$ in Eq. (24) denotes the cumulative arrival count of vehicles on link (i, j) by time t , and $D(i, j, t)$ denotes the cumulative departure count of vehicles on link (i, j) by time t . The values of $A(i, j, t)$ and $D(i, j, t)$ both depend only on the decision variable $x_{i,j,\tau,\tau}^v$, which are formulated below.

$$A(i, j, t) = \sum_{v \in V} \sum_{\tau \leq t} x_{i,j,\tau,\tau}^v, \forall (i, j, \tau, \tau) \in E_{tr}^p \cup E_{tr}^m, \forall (t) \in T \quad (25)$$

$$D(i, j, t) = \sum_{v \in V} \sum_{\tau \leq t} x_{i,j,\tau,\tau}^v, \forall (i, j, \tau, \tau) \in E_{tr}^p \cup E_{tr}^m, \forall (t) \in T \quad (26)$$

Similar to the gate capacity constraint (Eq. (6)), the parking and moving capacity constraint can also be dualized into the objective function with Lagrangian multipliers $\lambda_{i,j,t,i}$ and ADMM penalty parameter ρ . Then the proposed ADMM-based solution method can be applied to solve the problem.

However, introducing numerous multipliers may put pressure on the solving process. As we focus on the scheduling problem considering the impact of congestion, a widely used simulation-based point-queue model (Lawson et al., 1996; Zhou et al., 2015; Bartlett et al., 2014) can be adopted to estimate the time-dependent travel delay for AGVs, which is more applicable and efficient. A brief introduction of the point-queue model is illustrated in Appendix C.

In the point-queue model, the link travel time is assumed to consist of two parts: free-flow travel time FFT and delay t_W (Lu et al., 2016). Delay is a typical measure of the impact of congestion on vehicles' travel time, which is related to the current queue length of link (as shown in Fig. C2).

Based on Tong et al. (2019), the time-dependent queue length $Q(i, j, t)$ can be derived from the grant traffic state variables of $A(i, j, t)$ and $D(i, j, t)$, as shown in Eq. (27). Then, according to the classical speed-density model, the travel speed in congestion can be calculated by Eq. (28), where V_F denotes the free-flow travel speed, K_J denotes the jam density, and K_Q denotes the congested density. The physical length of link $L(i, j)$ is known, thus K_Q and V_Q can be calculated based on $Q(i, j, t)$. As a result, the time-dependent total travel time $TT(i, j, t)$ can be estimated.

$$Q(i, j, t) = A(i, j, t) - D(i, j, t) \quad (27)$$

$$V_Q = V_F \left(1 - \frac{K_Q}{K_J} \right) \quad (28)$$

The point queue model can be well embedded in the proposed rolling horizon framework. Specifically, in stage h , a predicted schedule plan for the look-ahead period can be generated based on Algorithm 2. As shown in Fig. 9, the space–time trajectories of AGVs in the look-ahead period can be used to estimate the time-dependent travel time for AGVs in stage $h + 1$.

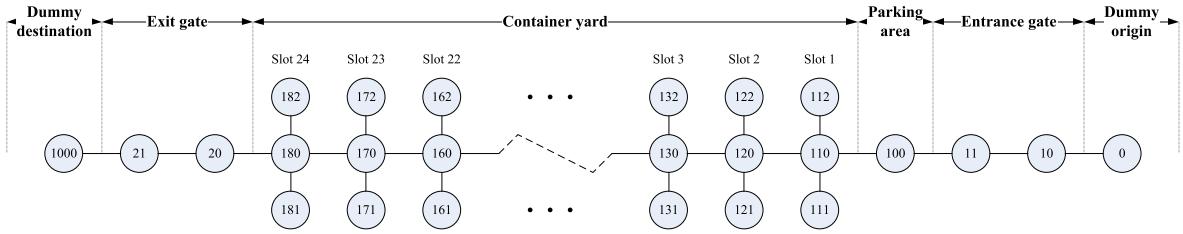
6. Numerical experiments

As discussed in Section 5, the key challenges in applying the multi-robot scheduling method include the following: (1) providing feasible and effective solutions for complex tasks with various complicated constraints (Khamis et al., 2015) and (2) providing an on-line scheduling framework for large-scale automated MRS (Katrakazas et al., 2015). In several recent applications of multi-robot scheduling, for example, Zheng et al. (2017) considered an experiment with 5 waterborne AGVs; a study by Atay and Bayazit (2006) provided a mixed-integer linear programming solution for MRTA problem with 8 robots; and Jones et al. (2011) studied a disaster evacuation problem with 3 fire trucks and 12 bulldozers and 100 tasks as well as the intra-path precedence constraints between

Table 4

Lower and upper bound values (in time unit) of the illustrative examples with different numbers of cranes and vehicles.

# of cranes	# of vehicles	ADMM-STN-C#		RCPSP-CPLEX	
		Upper bound	Lower bound	Upper bound	Lower bound
2	20	808	800	808	808
2	40	1702	1619.287	1683	1681.993
2	60	2496	2415.441	2492	2490
3	20	800	797	800	800
3	40	1647	1616.997	1646	1646
3	60	2469	2310.75	2461	2461
4	20	800	797	800	800
4	40	1658	1615.33	1642	1640.825
4	60	2427	2305.472	2422	2422

**Fig. 12.** Hypothetical container terminal network with 24 slots.

different robots using a genetic algorithm. In comparison, we design 9 illustrative scenarios in Section 6.1 with various numbers of cranes and AGVs as shown in **Table 4**. Scenario 9 contains a total of 4 cranes and 60 AGVs to be scheduled. Furthermore, a large-scale experiment based on a real-world case is implemented in Section 6.2.

Considering the limitation of space, we focus on testing our proposed solution method from the perspectives of optimality and efficiency in this paper. The ADMM-based solution method (i.e., ADMM-C#) is implemented in C# on the Visual Studio 2017 platform. All the models and algorithms are tested on a computer with i7-8750 @ 2.2 GHz CPU and 12 GB RAM. The C# source codes and space-time path visualization tool can be downloaded from the GitHub website: https://github.com/xchChen/CACSP_ADMM. Note that we also compare the ADMM-based approach and the STN reformulation model that involves the standard optimization solver in Appendix A, which provides a valid verification of the necessity of the proposed approach.

6.1. Illustrative experiments based on a hypothetical vehicle and crane network

A hypothetical, 79-node network is shown in **Fig. 12**. Both the entrance and exit gates include two truck lanes. The length of a time interval in the space-time network is 20 s. Based on the practice statistics (Yan et al., 2018), the average time cost for a container handling operation for cranes is set as 6 to 8-time units. The free-flow travel time of AGV within the container yard is set as 360 m/s.

We first perform a series of comparison tests of the ADMM-based solution and the RCPSP-CPLEX solution. Next, we perform a set of comparison tests between the fixed- and flexible-working-zone crane scheduling schemes using the ADMM-based solution method. The solution quality and computation time of each scheme are presented.

(1) Comparison between ADMM-based algorithm and RCPSP-CPLEX method

Table 5

Computation times and optimality gaps in illustrative examples with different numbers of cranes and vehicles.

# of cranes	# of vehicles	ADMM-STN-C#			RCPSP-CPLEX		
		CPU time (s)	Gap	Best Gap	CPU time (s)	Gap	Best Gap
2	20	0.916	1.00%	0.00%	33.479	0.00%	0.00%
2	40	12.337	5.11%	1.18%	263.332	0.06%	0.06%
2	60	39.675	3.34%	0.24%	1441.529	0.08%	0.08%
3	20	0.912	0.38%	0.00%	36.541	0.00%	0.00%
3	40	6.002	1.86%	0.06%	265.419	0.00%	0.00%
3	60	24.392	6.85%	0.32%	2314.108	0.00%	0.00%
4	20	0.895	0.38%	0.00%	34.029	0.00%	0.00%
4	40	4.689	2.64%	1.04%	214.458	0.07%	0.07%
4	60	17.001	5.27%	0.21%	1398.187	0.00%	0.00%

As it is difficult to consider the interferences among cranes by using the RCPSP modeling framework, we assume that each crane has a specific fixed working zone in the computation tests, i.e., non-crossing constraint (5) is not considered. Moreover, RCPSP model assumes that a new activity can be implemented as soon as a resource is available and the activity satisfies the adjacent constraint, as a result, the set-up times of cranes (i.e., the time cost of the horizontal movement of cranes and the movement of the spreaders) are neglected. In order to ensure the fairness of the comparison, the time cost of moving and recovery arcs in CSTN is set to 0 when using the ADMM-based approach.

The computation times and corresponding optimality gaps obtained experimentally with each method are listed in Tables 4 and 5, respectively. In Table 5, “GAP” represents the original optimality gap between upper bounds and lower bounds of the corresponding methods, and “BestGAP” denotes the best optimality gap whose value is calculated by adopting the best lower bound value among all methods.

Based on the computation results in Tables 4 and 5, the following observations can be made:

- (1) CPLEX obtains optimal solutions or near-optimal solutions with limited optimality gaps in most test cases.
- (2) As can be seen from the column “BestGap” of ADMM in Table 5, the upper bound values of ADMM are very close to the optimal values of CPLEX. The largest gap between the upper bound of ADMM and the best lower bound is 1.18%. Moreover, in the test cases with 20 AGVs, ADMM can obtain the optimal solution.
- (3) The computation times of CPLEX increase dramatically when the number of AGVs and cranes increase gradually. By contrast, the computation times of ADMM increase relatively slowly and the advantage is obvious for the scenarios with 60 AGVs. Although the maximum CPU time of CPLEX is less than 1 h, however, it can be inferred from Appendix A that when considering the non-crossing constraint, the computation complexity of CPLEX would be difficult to meet the requirement of real-time scheduling in a practical situation.
- (4) When the number of AGVs is fixed, the computation time in ADMM has a negative correlation with the number of cranes, because each crane has a fixed working range and the size of sub-CSTN of each crane decreases when the quantity of cranes increases, which results in the reduction of the solution space of the space-time least-cost path problem of each crane.
- (5) It is found that the Lagrangian relaxation method is difficult to obtain a good lower bound of the solution when the size of the problem increases, which leads to the relatively large gap of the ADMM-based solution. It can be inferred that, without the symmetry breaking mechanism, the symmetry issues in the AGV-crane synchronization problem are significant when using the pure Lagrangian relaxation method. In fact, owing to the existence of coupling constraints of AGV and crane agents, multiple vehicles are likely to be assigned to the same crane, which leads to the poor performance of the lower bound estimation in the Lagrangian relaxation method.

(2) Comparison between flexible-working-zone and fixed-working-zone crane schemes

As discussed in Section 1, the crane scheduling with the fixed working zone belongs to the *compound task* problem, which is less difficult than the *complex task* problem when the crane-working zone is flexible. However, the workloads of different cranes can be better balanced through cooperation when a flexible-working-zone scheme is adopted (Park et al., 2010; Yan, et al., 2019). To analyze the effectiveness of different working-zone schemes for cranes on the system performance, a set of tests are implemented. It is worth noting that the non-crossing constraint is considered when using the flexible-working-scheme.

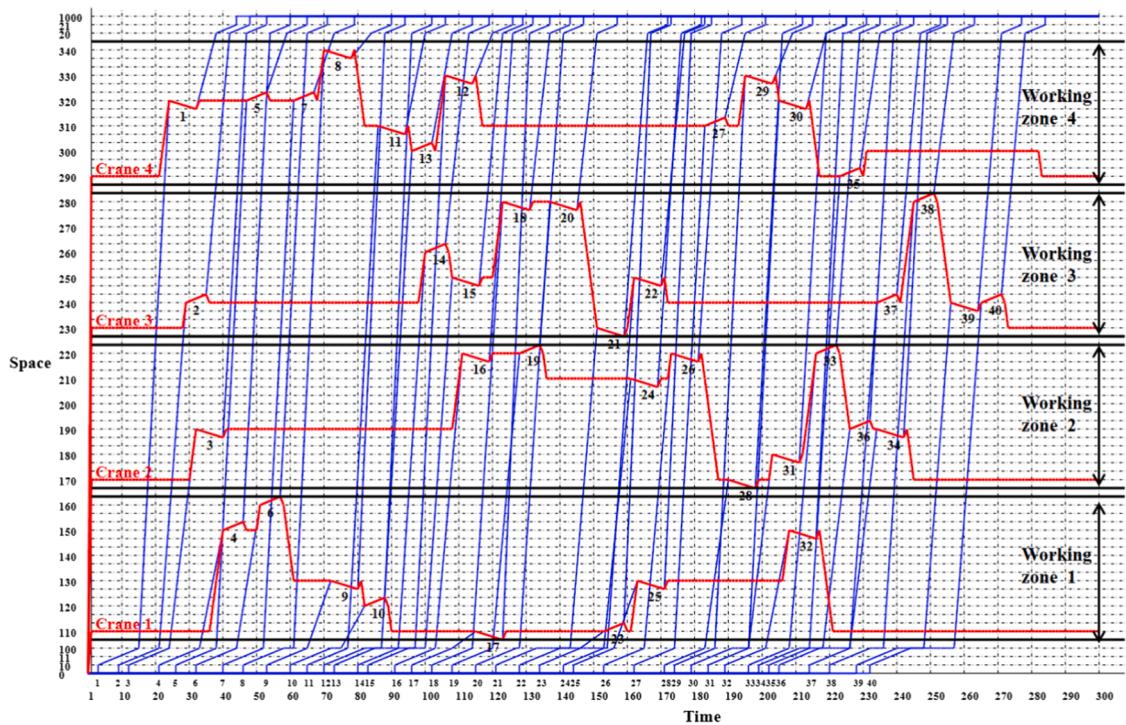
Table 6 shows the experimental results of the tests with different crane-working-zone schemes. Note that the time costs of the horizontal movement of cranes and movement of spreaders are considered in these cases. Therefore, the upper bounds, lower bounds, and computation times obtained in Table 6 are different from those in Table 4.

The scheme with flexible working-zones (FlexWZ) can provide better solutions compared with the fixed working-zones (FixWZ) scheme in all test cases. The space-time paths of cranes and AGVs in the second scenario with 4 cranes and 40 AGVs are presented. As shown in Fig. 13 (a), each crane has a specific working range surrounded by the black solid lines. Due to the limitation of the FixWZ, crane 4 cannot serve AGV 18 and AGV 20, although it is idle during the time period from 120 to 180. In contrast, as shown in Fig. 13 (b), AGV 18 and AGV 20 can be served by crane 4, and crane 3 can handle AGV 21 earlier compared with the case in Fig. 13 (a). The crane resources can be better utilized by applying the FlexWZ scheme, which results in better solutions. In particular, the waiting time of AGVs in scenario 2 can be reduced by 50% when the FixWZ scheme is replaced by the FlexWZ scheme. From the perspective of the solution algorithm, the reason for this phenomenon can be explained as follows. In the ADMM-based approach, the objective of the crane-specific sub-problem is to find the space-time least-cost path for each crane. On the other hand, we could note that the search space for cranes in CSTN is reduced when using the FixWZ scheme.

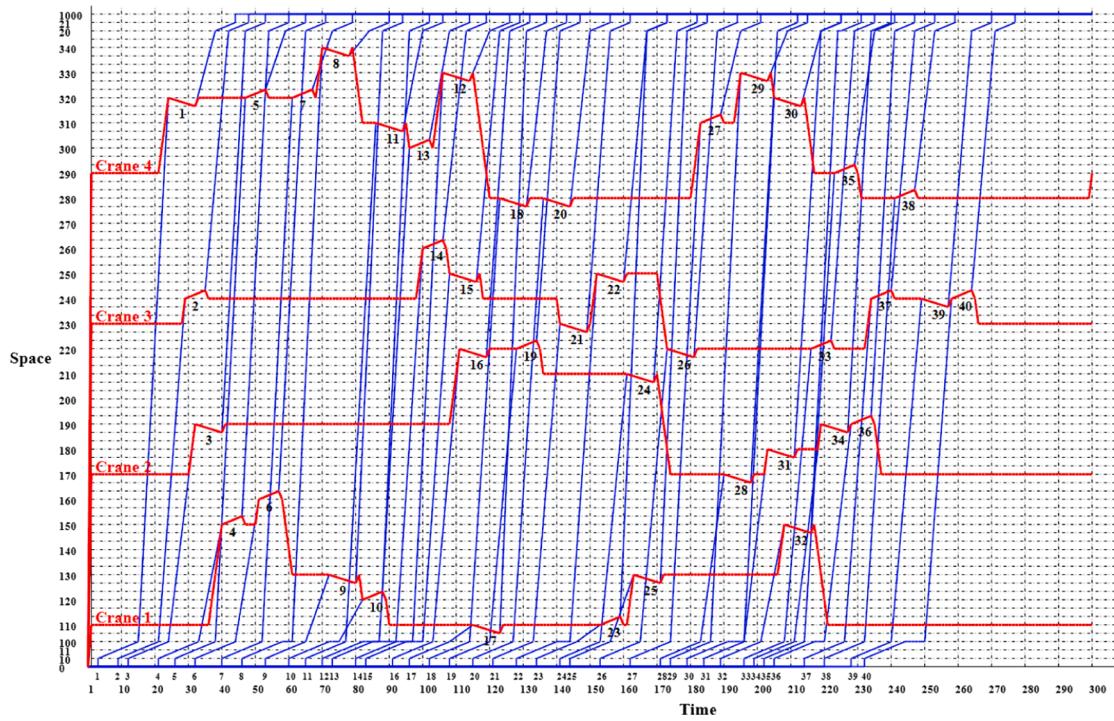
Table 6

Lower and upper bound values of ADMM (in time unit) in illustrative examples with different crane-working-zone schemes.

Scenario	# of cranes	# of vehicles	Flexible working range			Fixed working range		
			Upper bound	Lower bound	Waiting time	Upper bound	Lower bound	Waiting time
1	4	20	800	796	4	806	799	10
2	4	40	1654	1606	48	1711	1618	96
3	4	60	2499	2396	79	2594	2437	174



(a) Fixed crane-working zone (the numbers next to the lines are vehicle numbers)



(b) Flexible crane-working zone (the numbers next to the lines are vehicle numbers)

Fig. 13. Space-time paths obtained by ADMM for scenario 2 (blue and red solid lines indicating paths for vehicles and cranes, respectively). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

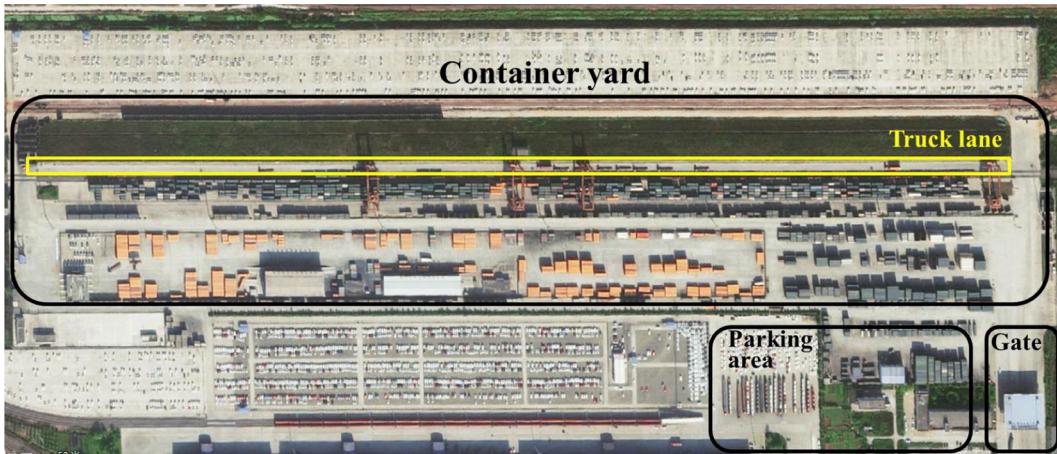


Fig. 14. Aerial view of Putian railway container terminal.

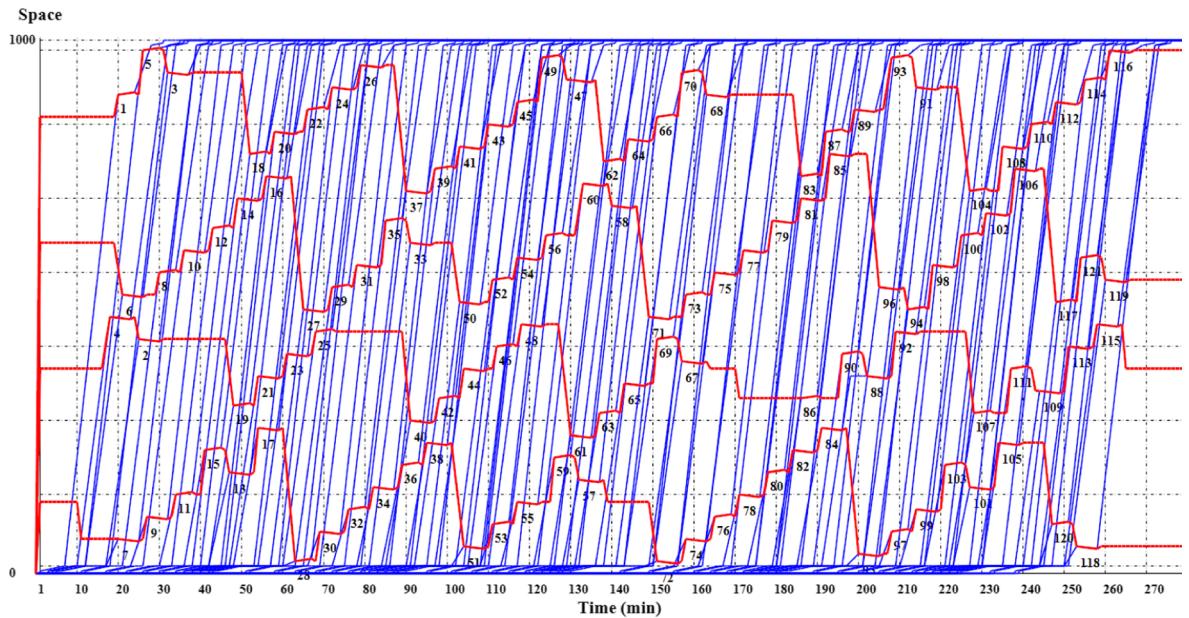


Fig. 15. Space-time paths of cranes and vehicles within a 4-h planning horizon.

6.2. Real-world case study

Putian railway container terminal is located in Zhengzhou city (China). It is one of the 18 core railway container terminals in China. This terminal is equipped with advanced technology including an intelligent gate system, automated container identification system, and a truck appointment system. Currently, Putian terminal has a container yard with 4 cranes, and the annual container volume is 0.7 million TEU. The layout of this terminal is presented in Fig. 14. The total horizontal length of the truck handling lane is 1050 m. Assuming the length of each slot to be 15 m, the handling lanes can be divided into 70 slots. The terminal gate has 6 truck lanes; 3 of which are used for arrival inspection operations and the other 3 are used for departure inspection operations.

This terminal handles more than 100 vehicles and an equal number of container-handling tasks. To operate this terminal efficiently, the proposed ADMM-based rolling horizon method in **Algorithm 2** is used. The total planning horizon length is 4 h, with a roll period of 20 min and a look-ahead period of 40 min. Setting the beam width γ as 5, the proposed ADMM-based algorithm performs 200 iterations at each horizon to solve this case. The computing time for each time horizon is approximately 75 s. The optimized scheduling scheme for the Putian container terminal is shown in Fig. 15, which provides detailed space–time paths for each crane and AGV robot within the planning horizon. The gap between the upper and lower bound solutions is 15.45%, reaching an acceptable level within efficient computation on time considering the size of the problem.

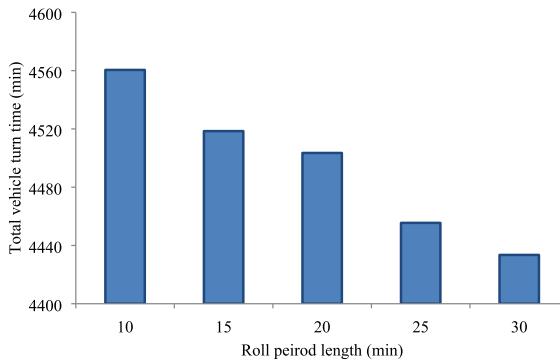


Fig. 16. Objective function values under various roll periods.

Furthermore, we discuss some system performances under various horizon lengths as illustrated in Fig. 16. In CACSP, the roll period r refers to how often the scheduling scheme is updated. From the perspective of responding to uncertainty/emergencies, a smaller value of r tends to provide better solutions (Zhan et al., 2016). However, updating the space-time scheduling scheme too often reduces the global system performance of the scheduling plan and is inconvenient for the terminal management in practical situations. On the other hand, a large roll period length tends to yield better global solutions for the system. As shown in Fig. 16, the total vehicle turn time can be reduced by 127 min by using a roll period of 30 min instead of 10 min. However, a large time horizon needs more computation effort to solve the problem. Thus, the terminal manager has to make a trade-off between the effort to obtain a globally optimal solution versus more adaptive updates.

7. Conclusions and future work

In this study, an AGV-crane synchronization model based on the extended space-time network is constructed for the container hub with automated technology. Compared with the existing crane scheduling studies, this study further integrates the scheduled-based space-time vehicle route planning into the optimization problem to provide the precise space-time paths for AGV and crane robots within the terminal system. To synchronize the vehicle and crane agents in performing complex tasks, the side coupling constraints, terminal gate capacity constraints, and the crane non-crossing constraints are considered through a market-driven ADMM-based dual decomposition approach. We discuss various ways to solve a large-scale crane and AGV scheduling problem in real-time. Moreover, a set of illustrative and real-life experiments are performed to test the efficiency and effectiveness of the proposed model and the solution algorithms. In particular, the proposed model and ADMM-based solution can generate significant time saving compared with the RCPSP model solved by CPLEX solver. Furthermore, the optimality gaps of ADMM for most of the test cases are significantly small.

In our future work, several interesting directions can be studied. First, the synchronization agents have been limited to AGVs and cranes in this study. However, more transshipments and handling agents including cargo trains, vessels, forklifts, and so on can be further synchronized to provide a better system-wide system optimization schedule planning. Second, more practical factors, including different types of layout for container storage areas, container handling time window constraints, and container stacking sequence constraints can be further considered using the space-time network-based formulation. Third, some more effective update mechanisms, such as the cutting plane method (Kallehauge, 2008; Lubin et al., 2013) where the values of Lagrangian multipliers maybe not unique at each iteration can be applied in the future to further improve the convexity of ADMM.

CRediT authorship contribution statement

Xuchao Chen: Conceptualization, Software, Writing - original draft, Visualization. **Shiwei He:** Supervision, Resources. **Yongxiang Zhang:** Writing - original draft, Methodology. **Lu (Carol) Tong:** Supervision, Writing - review & editing. **Pan Shang:** Investigation, Validation, Formal analysis. **Xuesong Zhou:** Conceptualization, Methodology, Writing - review & editing.

Acknowledgments

This study was supported by National Key R&D Program of China (No. 2018YFB1201402), National Natural Science Foundation of China (No. 71801006, No. U1734204), and Research Projects of China Railway Corporation (No. 2016X007-G, No. 2017X004-D, No. P2018X003). The last author was partially funded by National Science Foundation-United States under NSF grant no. CMMI 1663657 “Real-time Management of Large Fleets of Self-Driving Vehicles Using Virtual Cyber Tracks.”

Appendix A . Comparison tests between the ADMM-based approach and the STN model using CPLEX

The experiment is implemented using a hypothetical network that contains 12 slots. The entrance and exit gates both include two

Table A1

Lower and upper bound values (in time unit) of illustrative examples with different numbers of cranes and AGVs.

# of cranes	# of vehicles	ADMM-STN-C#		STN-CPLEX	
		Upper bound	Lower bound	Upper bound	Lower bound
3	5	216	199	216	216
3	10	466	400	463	463
3	15	687	603	672	654
3	20	996	794	987	972.468
3	25	1290	1005	N/A	N/A

Table A2

Computation times and optimality gaps in illustrative examples with different numbers of cranes and AGVs.

# of cranes	# of vehicles	ADMM-STN-C#			STN-CPLEX		
		CPU time (s)	Gap	Best Gap	CPU time (s)	Gap	Best Gap
3	5	4.787	7.87%	0.00%	100.314	0.00%	0.00%
3	10	5.869	14.16%	0.64%	1270.403	0.00%	0.00%
3	15	21.143	12.23%	4.80%	3600	2.68%	2.68%
3	20	27.2	20.28%	2.36%	3600	1.47%	1.47%
3	25	47.487	22.09%	22.09%	N/A	N/A	N/A

truck lanes. A total of 3 cranes are placed in the container yard with a flexible working range and the number of AGVs varies from 5 to 25. The STN model is solved by CPLEX 12.8.0 (i.e., STN-CPLEX). All the models and algorithms are tested on a computer with i7-8750 @ 2.2 GHz CPU and 12 GB RAM. The time limit is set as 3600 s. Computation times and corresponding optimality gaps are listed in [Tables A1 and A2](#). In [Table A2](#), “BestGAP” denotes the best optimality gap whose value is calculated by adopting the best lower bound value among all methods.

Computational results show that the CPLEX solver obtains optimal solutions for the first two test cases and closes to the optimal solution for the third and fourth cases within the time limit. In the last cases with 25 vehicles, the CPLEX is unable to obtain a feasible solution due to the memory limitation of the computer. However, the ADMM-based approach can obtain optimal or near-optimal solutions for most of the test cases. The best gaps for the first four test cases are below 5%. Besides, the computation times of CPLEX grow sharply when the number of vehicles increases. By contrast, computation times of ADMM-based approach increase slowly due to the efficient decomposition mechanism. Compared with the comparative experiments between ADMM-STN and RCPSP-CPLEX in Section 5, the complexities of the experiments in Appendix A increase due to the non-crossing constraint. The gaps of ADMM-STN-C# show that when dealing with complicated side constraints, the pure Lagrangian relaxation method faces the challenge to obtain satisfactory lower bound solutions. This further indicates the necessity of the block-by-block sequential updating scheme in handling the complex inter-robot constraint.

Appendix B. . Comparison of STN model with the modified activity-based RCPSP model in [Pritsker et al. \(1969\)](#)

To model the CACSP using a standard RCPSP formulation, we assume that each crane has a fixed working zone with no interferences. Besides, the real-time location of the crane is not considered, thus the time costs of the crane horizontal movement and the spreader movement are ignored. If the real-time location of the crane is considered, the problem can be transformed into a more complicated dynamic RCPSP ([Artigues et al., 2003](#)). Seven types of activities are constructed to describe the operations for each vehicle. Furthermore, two dummy activities are added to represent the start and termination activities. Note that the duration of pre-

Table B1

Descriptions of activities in RCPSP.

#	Description	Resource type	Availability of resource	Successor
1	Pre-arrival	None	Unlimited	2
2	Arrival inspection	Entrance gate	number of truck lanes	3
3	Moving from entrance gate to parking area	Truck lanes	Unlimited	4
4	Moving from parking area to yard	Truck lanes	Unlimited	5
5	Container handling	Crane	disjunctive resource	6
6	Moving from yard to exit gate	Truck lanes	Unlimited	7
7	Departure inspection	Exit gate	number of truck lanes	n
0	Start activity	None	Unlimited	1
n	Termination activity	None	Unlimited	/

Table B2
Decision variable and subscripts for model formulation.

Notations	Definitions
i, j	Index of activities, $i, j \in A$, where A is the set of activities
t	Index of time intervals, $t, t' = 1, 2, \dots, T$, where T is the set of time intervals
v	Index of vehicles, $v \in V$, where V is the set of vehicles
r	Index of resources, $r \in R$, where R is the set of resources
$s(v)$	Index of the pre-arrival activity of vehicle v
$e(v)$	Index of the departure inspection activity of vehicle v
p_i	Processing duration time of activity i
C_r	Availability of resource r
E	Set of pairs of activities indicating precedence constraints
x_{it}	0–1 variable, = 1 if activity i starts at time t ; = 0 otherwise.

Table B3

Comparison of STN model with the modified activity-based RCPSP model.

Requirement	Modified RCPSP model	STN model
Precedence limitation	Constraint (B2)	Flow-balanced constraints (2) and (3)
Resource limitation	Constraint (B3)	Coupling constraint (4) and gate capacity constraint (6)
Non-crossing constraint	Not considered by assuming that each crane has a fixed working zone	Constraint (5)
Horizontal movement of cranes	Not considered	By using moving arc(i, j, t, t')
Movement of spreaders	Not considered	By using recovery arc(i, j, t, t')

arrival activity denotes the arrival time of vehicle v . The detailed instructions for each type of activity are presented in [Table B1](#).

The RCPSP model can be constructed using the basic discrete-time formulation given by [Pritsker et al. \(1969\)](#). The decision variable and subscripts for the model formulation are listed in [Table B2](#).

• M B1

Objective function

$$\text{Min} \left(\sum_{t \in H} \sum_{v \in V} x_{e(v)t} + p_{e(v)} - \sum_{t \in H} \sum_{v \in V} x_{s(v)t} \right) \quad (\text{B1})$$

Subject to

$$\sum_{t \in H} t x_{jt} \geq \sum_{t \in H} t x_{it} + p_i, \forall (i, j) \in E \quad (\text{B2})$$

$$\sum_{i \in A} b_{ik} \sum_{\tau=t-p_i+1}^t x_{i\tau} \leq C_r, \forall t \in T, \forall r \in R \quad (\text{B3})$$

$$\sum_{t \in T} x_{it} = 1, \forall i \in A \quad (\text{B4})$$

$$x_{it} \in \{0, 1\}, \forall i \in A, \forall t \in T \quad (\text{B5})$$

The objective function in Eq. (B1) is to minimize the total vehicle turn time. Constraint (B2) restricts the precedence limits of activities, and the resource limit is enforced by constraint (B3). The comparison of the STN model with the modified RCPSP model is summarized in [Table B3](#). It can be concluded that the STN model can handle all constraints in the modified RCPSP model. In addition, the non-crossing constraint and movement process of cranes and spreaders, which are difficult to be described using the RCPSP formulation, can be handled easily in the STN model.

Appendix C . Simulation-based point queue model

In the point-queue model, the link travel time is assumed to consist of two parts: free-flow travel time FFT and delay t_w ([Lu et al., 2016](#)). The illustration of the vehicle trajectory on a link is shown in [Fig. C1](#), where the solid line depicts the actual trajectory of vehicle v in congested conditions, while the dotted line represents the desired trajectory of the vehicle. The queue length is denoted by $Q(i, j)$. Vehicle v enters the link (i, j) at time t , desired to leave the link at time t^d , and eventually leaves the link at time t' .

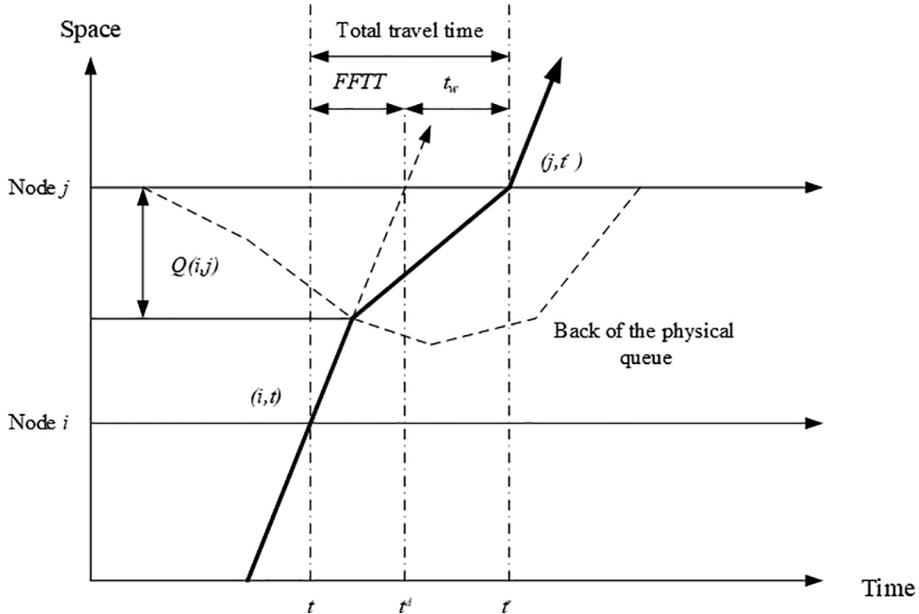


Fig. C1. Vehicle trajectory on a link in congested conditions (adopted from Tong et al., 2019).

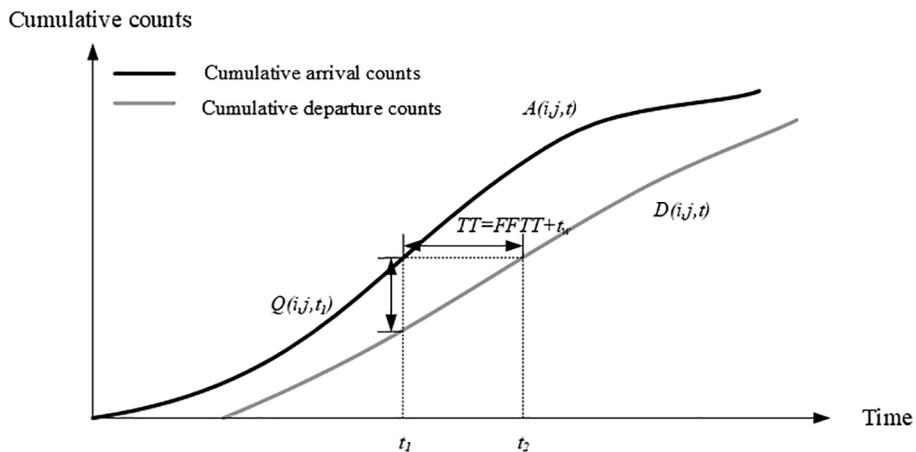


Fig. C2. Queue length and travel time in the simulation process (adopted from Tong et al., 2019).

Therefore, the free-flow time can be denoted as $FFTT = t^d - t$, the delay $t_w = t' - t^d$. The total travel time TT of vehicle v on link (i, j) can be described as follows:

$$TT = FFTT + t_w \quad (C1)$$

Under congested conditions, the total travel time and queue length are time-varying and can be denoted as $TT(i, j, t)$ and $Q(i, j, t)$, respectively. Based on Tong et al. (2019), the waiting time t_w and queue length $Q(i, j, t)$ can be derived from the system-level traffic state variables of $A(i, j, t)$ and $D(i, j, t)$ through simulation. The relationship between the t_w , $Q(i, j, t)$ and cumulative count variables are illustrated in Fig. C2.

The relationship between the queue length and the travel speed can be determined using the classical Greenshields' or other types of speed-density model. With the given length of link $L(i, j)$, the K_Q can be calculated based on the queue length $Q(i, j, t)$. Then, the time-dependent total travel time $TT(i, j, t)$ can be estimated.

References

- Abdelkader, M., 2018. Towards Real-Time Distributed Planning in Multi-Robot Systems. King Abdullah University of Science and Technology.
Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall.

- Angeloudis, P., Bell, M.G., 2010. An uncertainty-aware AGV assignment algorithm for automated container terminals. *Transportation Research Part E: Logistics and Transportation Review* 46 (3), 354–366.
- Artigues, C., Michelon, P., Reusser, S., 2003. Insertion techniques for static and dynamic resource-constrained project scheduling. *Eur. J. Oper. Res.* 149 (2), 249–267.
- Atay, N., Bayazit, B., 2006. Mixed-Integer Linear Programming Solution to Multi-Robot Task Allocation Problem. Washington Univ., St. Louis, Tech. Rep. WUCSE 54(314).
- Badreldin, M., Hussein, A., Khamis, A., 2013. A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation. *Advances in Artificial Intelligence* 2013, 1–11.
- Bartlett, K., Lee, J., Ahmed, S., Nemhauser, G., Sokol, J., Na, B., 2014. Congestion-aware dynamic routing in automated material handling systems. *Comput. Ind. Eng.* 70, 176–182.
- Bertsekas, D.P., 1999. Nonlinear Programming. Athena Scientific, Belmont.
- Bento, J., Derbinsky, N., Alonso-Mora, J., Yedidia, J., 2013. A message-passing algorithm for multi-agent trajectory planning. *Advances in Neural Information Processing Systems* 1–9.
- Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghriby, W., Griffin, P., Kleywegt, A., 2003. Robot Exploration with Combinatorial Auctions. *IEEE International Conference on Intelligent Robots and Systems* 2, 1957–1962.
- Bierwirth, C., Meisel, F., 2009. A fast heuristic for quay crane scheduling with interference constraints. *J. Sched.* 12 (4), 345–360.
- Bierwirth, C., Meisel, F., 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* 202 (3), 675–689.
- Bierwirth, C., Meisel, F., 2015. A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *Eur. J. Oper. Res.* 244 (3), 675–689.
- Bish, E.K., Leong, T.Y., Li, C.L., Ng, J.W., Simchi-Levi, D., 2001. Analysis of a new vehicle scheduling and location problem. *Naval Research Logistics (NRL)* 48 (5), 363–385.
- Bish, E.K., 2003. A multiple-crane-constrained scheduling problem in a container terminal. *Eur. J. Oper. Res.* 144 (1), 83–107.
- BMBF-Internetredaktion, 2016. Zukunftsprojekt Industrie 4.0 - BMBF. Bmbf.de. Retrieved 30 November 2016. <<https://www.bmbf.de/de/zukunftsprojekt-industrie-4.0-848.html>> .
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2010. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3 (1), 1–122.
- Boysen, N., Briskorn, D., Meisel, F., 2017. A generalized classification scheme for crane scheduling with interference. *Eur. J. Oper. Res.* 258 (1), 343–357.
- Boysen, N., Fliedner, M., 2010. Determining crane areas in intermodal transhipment yards: the yard partition problem. *Eur. J. Oper. Res.* 204 (2), 336–342.
- Boysen, N., Fliedner Malte, M., Kellner Michael, M., 2010. Determining fixed crane areas in rail-rail transhipment yards. *Transportation Research Part E: Logistics and Transportation Review* 46 (6), 1005–1016.
- Boysen, N., Stephan, K., 2016. A survey on single crane scheduling in automated storage/retrieval systems. *Eur. J. Oper. Res.* 254 (3), 691–704.
- Bragin, M.A., Luh, P.B., Yan, B., Sun, X., 2018. A Scalable Solution Methodology for Mixed-Integer Linear Programming Problems Arising in Automation. *IEEE Trans. Autom. Sci. Eng.* 16 (2), 531–541.
- Cao, Z., Lee, D.H., Meng, Q., 2008. Deployment strategies of double-rail-mounted gantry crane systems for loading outbound containers in container terminals. *Int. J. Prod. Econ.* 115 (1), 221–228.
- Cao, J.X., Lee, D.H., Chen, J.H., Shi, Q., 2010. The integrated yard truck and yard crane scheduling problem: Benders' decomposition-based methods. *Transportation Research Part E: Logistics and Transportation Review* 46 (3), 344–353.
- Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. *Oper. Res.* 50 (5) 851–861 + 916.
- Carlo, H.J., Martínez-Acevedo, F.L., 2015. Priority rules for twin automated stacking cranes that collaborate. *Comput. Ind. Eng.* 89, 23–33.
- Carlo, H.J., Vis, I.F., Roodbergen, K.J., 2014. Storage yard operations in container terminals: Literature overview, trends, and research directions. *Eur. J. Oper. Res.* 235, 412–430.
- Carlo, H.J., Vis, I.F., Roodbergen, K.J., 2015. Seaside operations in container terminals: literature overview, trends, and research directions. *Flexible Services and Manufacturing Journal* 27 (2–3), 224–262.
- Chabini, I., 1998. Discrete dynamic shortest path problems in transportation applications: complexity and algorithms with optimal run time. *Transp. Res. Rec.* 1645, 170–175.
- Chang, D., Jiang, Z., Yan, W., He, J., 2011. Developing a dynamic rolling-horizon decision strategy for yard crane scheduling. *Adv. Eng. Inf.* 25 (3), 485–494.
- Chen, J.H., Lee, D.H., Cao, J.X., 2011. Heuristics for quay crane scheduling at indented berth. *Transportation Research Part E: Logistics and Transportation Review* 47 (6), 1005–1020.
- Chen, J.H., Lee, D.H., Goh, M., 2014. An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem. *Eur. J. Oper. Res.* 232 (1), 198–208.
- Chen, L., Bostel, N., Dejax, P., Cai, J., Xi, L., 2007. A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *Eur. J. Oper. Res.* 181 (1), 40–58.
- Chen, L., Langevin, A., Lu, Z., 2013. Integrated scheduling of crane handling and truck transportation in a maritime container terminal. *Eur. J. Oper. Res.* 225 (1), 142–152.
- Daganzo, C.F., 1989. THE CRANE PROBLEM. *Transportation Research Part B: Methodological* 23 (3), 159–175.
- Dias, M.B., 2004. TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments.
- Díaz, J.A., Fernández, E., 2001. A Tabu search heuristic for the generalized assignment problem \$50: Juan A. Díaz, and Elena Fernández, *European Journal of Operational Research* 132 (1), 22–38.
- Dohn, A., Rasmussen, M.S., Larsen, J., 2009. The vehicle routing problem with time windows and temporal dependencies. *Networks* 58 (4), 273–289.
- Fisher, M.L., 1981. The Lagrangian relaxation method for solving integer programming problems. *Manage. Sci.* 27 (1), 1–18.
- Fisher, M.L., Jörnsten, K.O., Madsen, O.B.G., 1997. Vehicle routing with time windows: Two optimization algorithms. *Oper. Res.* 45 (3), 488–492.
- Fortin, M., Glowinski, R., 2000. Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems Vol. 15 Elsevier.
- Fu, Y.M., Diabat, A., Tsai, I.T., 2014. A multi-vessel quay crane assignment and scheduling problem: Formulation and heuristic solution approach. *Expert Syst. Appl.* 41 (15), 6959–6965.
- Gabay, D., Mercier, B., 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation 2 (1), 17–40.
- Gerkey, B. P., Mataric, M. J., 2004. Multi-robot task allocation: analyzing the complexity and optimality of key architectures, (Icra), 3862–3868.
- Gharehgolzi, A.H., Laporte, G., Yu, Y., de Koster, R., 2014. Scheduling Twin Yard Cranes in a Container Block. *Transportation Science* 49 (3), 686–705.
- Gharehgolzi, A.H., Vernooy, F.G., Zaerpour, N., 2017. A simulation study of the performance of twin automated stacking cranes at a seaport container terminal. *Eur. J. Oper. Res.* 261 (1), 108–128.
- 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.
- He, J., Chang, D., Mi, W., Yan, W., 2010. A hybrid parallel genetic algorithm for yard crane scheduling. *Transp. Res. Part E* 46 (1), 136–155.
- He, J., Zhang, W., Huang, Y., Yan, W., 2013. A simulation optimization method for internal trucks sharing assignment among multiple container terminals. *Adv. Eng. Inf.* 27 (4), 598–614.
- He, J., Huang, Y., Yan, W., Wang, S., 2015. Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption. *Expert Syst. Appl.* 42 (5), 2464–2487.
- Jones, E.G., Dias, M.B., Stentz, A., 2011. Time-extended multi-robot coordination for domains with intra-path constraints. *Robotics: Science and Systems* 5, 273–280.
- Jung, S.H., Kim, K.H., 2006. Load scheduling for multiple quay cranes in port container terminals. *J. Intell. Manuf.* 17 (4), 479–492.
- Kallehauge, B., 2008. Formulations and exact algorithms for the vehicle routing problem with time windows. *Comput. Oper. Res.* 35 (7), 2307–2330.
- Kallehauge, B., Larsen, J., Madsen, O. B., 2001. Lagrangean duality applied on vehicle routing with time windows-experimental results.
- Katrakazas, C., Quddus, M., Chen, W.H., Deka, L., 2015. Real-time motion planning methods for autonomous on-road driving: state-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies* 60, 416–442.
- Kaveshgar, N., Huynh, N., Rahimian, S.K., 2012. An efficient genetic algorithm for solving the quay crane scheduling problem. *Expert Syst. Appl.* 39 (18), 13108–13117.
- Kellner, M., Boysen, N., 2015. RMG vs. DRMG: an evaluation of different crane configurations in intermodal transhipment yards. *EURO Journal on Transportation and*

- Logistics 4 (3), 355–377.
- Khamis, A., Hussein, A., Elmogy, A., 2015. Multi-robot Task Allocation: A Review of the State-of-the-Art. *Cooperative Robots and Sensor Networks* 2014 (554), 31–52.
- Kim, K.H., Park, Y.M., 2004. A crane scheduling method for port container terminals. *Eur. J. Oper. Res.* 156 (3), 752–768.
- Kizilay, D., Eliyi, D.T., Van Hentenryck, P., 2018. Constraint and mathematical programming models for integrated port container terminal operations. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10848 LNCS, 344–360.
- Klaus Lysdal, IContainers, 2018. The future of automation at terminals and ports. < <https://www.iconainers.com/us/2018/10/09/the-future-of-automation-at-terminals-and-ports/> > .
- Koenig, S., Tovey, C., Zheng, X., Sungur, I., 2007. Sequential bundle-bid single-sale auction algorithms for decentralized control. *IJCAI International Joint Conference on Artificial Intelligence* 1359–1365.
- Korsah, G.A., 2011. Exploring Bounded Optimal Coordination for Heterogeneous Teams with Cross-Schedule Dependencies. *ProQuest Dissertations and Theses*.
- Lawson, T.W., Lovell, D.J., Daganzo, C.F., 1996. Using Input-Output Diagram to Determine Spatial and Temporal Extents of a Queue Upstream of a Bottleneck. *Transportation Research Record*, No. 1572, 140–147.
- Lee, D.H., Cao, Z., Meng, Q., 2007. Scheduling of two-transstainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. *Int. J. Prod. Econ.* 107 (1), 115–124.
- Lee, D.H., Wang, H.Q., Miao, L., 2008. Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E: Logistics and Transportation Review* 44 (1), 124–135.
- Lee, D.H., Cao, J.X., Shi, Q., Chen, J.H., 2009. A heuristic algorithm for yard truck scheduling and storage allocation problems. *Transportation Research Part E: Logistics and Transportation Review* 45 (5), 810–820.
- Legato, P., Trunfio, R., Meisel, F., 2012. Modeling and solving rich quay crane scheduling problems. *Comput. Oper. Res.* 39 (9), 2063–2078.
- Lemaire, T., Alami, R., Lacroix, S., 2004, April. A distributed tasks allocation scheme in multi-UAV context. In: *IEEE International Conference on Robotics and Automation*. 2004. Proceedings. ICRA'04. 2004 (Vol. 4, pp. 3622–3627). IEEE.
- Li, C.L., Vairaktarakis, G.L., 2004. Loading and unloading operations in container terminals. *IIE Trans.* 36 (4), 287–297.
- Li, W., Wu, Y., Petering, M.E.H., Goh, M., de Souza, R., 2009. Discrete time model and algorithms for container yard crane scheduling. *Eur. J. Oper. Res.* 198 (1), 165–172.
- Lu, C.C., Liu, J., Qu, Y., Peeta, S., Roushail, N.M., Zhou, X., 2016. Eco-system optimal time-dependent flow assignment in a congested network. *Transportation Research Part B: Methodological* 94, 217–239.
- Lubin, M., Martin, K., Petra, C.G., Sandikçi, B., 2013. On parallelizing dual decomposition in stochastic integer programming. *Operations Research Letters* 41 (3), 252–258.
- Luo, J., Wu, Y., Halldorsson, A., Song, X., 2011. Storage and stacking logistics problems in container terminals. *OR Insight* 24 (4), 256–275.
- Mahmoudi, M., Zhou, X., 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: a dynamic programming approach based on state-space-time network representations. *Transportation Research Part B: Methodological* 89, 19–42.
- McKinsey & Company, 2018. The future of automated ports. < <https://www.mckinsey.com/industries/travel-transport-and-logistics/our-insights/the-future-of-automated-ports> > .
- Meisel, F., Bierwirth, C., 2013. A Framework for Integrated Berth Allocation and Crane Operations Planning in Seaport Container Terminals. *Transportation Science* 47 (2), 131–147.
- Meng, L., Zhou, X., 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach. *Transportation Research Part B: Methodological* 45 (7), 1080–1102.
- Mota, J.F.C., Xavier, J.M.F., Aguiar, P.M.Q., Püschel, M., 2013. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Trans. Signal Process.* 61 (10), 2718–2723.
- Ng, W.C., 2005. Crane scheduling in container yards with inter-crane interference. *Eur. J. Oper. Res.* 164 (1), 64–78.
- Ng, W.C., Mak, K.L., 2005. Yard crane scheduling in port container terminals. *Appl. Math. Model.* 29 (3), 263–276.
- Nguyen, V.D., Kim, K.H., 2009. A dispatching method for automated lifting vehicles in automated port container terminals. *Comput. Ind. Eng.* 56 (3), 1002–1020.
- Niu, H., Zhou, X., Tian, X., 2018. Coordinating assignment and routing decisions in transit vehicle schedules: a variable-splitting Lagrangian decomposition approach for solution symmetry breaking. *Transportation Research Part B: Methodological* 107, 70–101.
- Park, S. S., Ha, J. S., Cho, D. H., Choi, H. L., 2018. A distributed ADMM approach to informative trajectory planning for multi-target tracking. arXiv preprint arXiv:1807.11068.
- Park, T., Choe, R., Ok, S.M., Ryu, K.R., 2010. Real-time scheduling for twin RMGs in an automated container yard. *OR Spectrum* 32 (3), 593–615.
- Peterofsky, R.J., Daganzo, C.F., 1990. A branch and bound solution method for the crane scheduling problem. *Transp. Res. Part B* 24 (3), 159–172.
- Pritsker, A., Watters, L., Wolfe, P., 1969. Multi-project scheduling with limited resources: a zero-one programming approach. *Manage. Sci.* 16, 93–108.
- Rasmussen, M.S., Justesen, T., Dohn, A., Larsen, J., 2010. The Home Care Crew Scheduling Problem: Preference-based visit clustering and temporal dependencies. *Eur. J. Oper. Res.* 219 (3), 598–610.
- Sabuncuoglu, I., Bayiz, M., 1999. Job shop scheduling with beam search 118, 390–412.
- Sammarra, M., Cordeau, J.F., Laporte, G., Monaco, M.F., 2007. A tabu search heuristic for the quay crane scheduling problem. *J. Sched.* 10 (4–5), 327–336.
- Shang, P., Li, R., Guo, J., Xian, K., Zhou, X., 2019. Integrating Lagrangian and Eulerian observations for passenger flow state estimation in an urban rail transit network: A space-time-state hyper network-based assignment approach. *Transportation Research Part B: Methodological* 121, 135–167.
- Shang, P., Li, R., Liu, Z., Yang, L., Wang, Y., 2018. Equity-oriented skip-stopping schedule optimization in an oversaturated urban rail transit network. *Transportation Research Part C: Emerging Technologies* 89 (June 2017), 321–343.
- Tang, F., Parker, L. E., 2007. A complete methodology for generating multi-robot task solutions using ASyMTRe-D and market-based task allocation. In: *Proceedings - IEEE International Conference on Robotics and Automation*, (May), 3351–3358.
- Tang, L., Zhao, J., Liu, J., 2014. Modeling and solution of the joint quay crane and truck scheduling problem. *Eur. J. Oper. Res.* 236 (3), 978–990.
- Tong, L. (Carol), Zhou, L., Liu, J., Zhou, X., 2017. Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. *Transportation Research Part C: Emerging Technologies*, 85(October 2016), 451–475.
- Tong, L., Pan, Y., Shang, P., Guo, J., Xian, K., Zhou, X., 2019. Open-Source Public Transportation Mobility Simulation Engine DTALite-S: A Discretized Space-Time Network-Based Modeling Framework for Bridging Multi-agent Simulation and Optimization. *Urban Rail Transit* 5 (1), 1–16.
- Unsal, O., Oguz, C., 2013. Constraint programming approach to quay crane scheduling problem. *Transportation Research Part E: Logistics and Transportation Review* 59, 108–122.
- Van Parys, R., Pipeleers, G., 2016. Online distributed motion planning for multi-vehicle systems. *2016 European Control Conference, ECC 2016*, 32, 1580–1585.
- Vis, I.F.A., Carlo, H.J., 2010. Sequencing Two Cooperating Automated Stacking Cranes in a Container Terminal. *Transportation Science* 44 (2), 169–182.
- Wang, J.P., Gu, Y., Li, X.M., 2012. Multi-robot task allocation based on ant colony algorithm. *Journal of Computers* 7 (9), 2160–2167.
- Wu, Y., Li, W., Petering, M.E.H., Goh, M., de Souza, R., 2015. Scheduling Multiple Yard Cranes with Crane Interference and Safety Distance Requirement. *Transportation Science* 49 (4), 990–1005.
- Yan, B., Zhu, X., Wang, L., Chang, Y., 2018. Integrated Scheduling of Rail-Mounted Gantry Cranes, Internal Trucks and Reach Stackers in Railway Operation Area of Container Terminal. *Transp. Res. Rec.* 2672 (9), 47–58.
- Yan, W., Huang, Y., Chang, D., He, J., 2011. An investigation into knowledge-based yard crane scheduling for container terminals. *Adv. Eng. Inf.* 25 (3), 462–471.
- Yan, S., Lu, C.C., Hsieh, J.H., Lin, H.C., 2019. A Dynamic and Flexible Berth Allocation Model with Stochastic Vessel Arrival Times. *Networks and Spatial Economics* 1–25.
- Yao, Y., Zhu, X., Dong, H., Wu, S., Wu, H., Tong, L.C., Zhou, X., 2019. ADMM-based problem decomposition scheme for vehicle routing problem with time windows. *Transportation Research Part B: Methodological* 129, 156–174.
- Zeng, Q., Yang, Z., 2009. Integrating simulation and optimization to schedule loading operations in container terminals. *Comput. Oper. Res.* 36 (6), 1935–1944.
- Zhan, S., Kroon, L.G., Zhao, J., Peng, Q., 2016. A rolling horizon approach to the high speed train rescheduling problem in case of a partial segment blockage. *Transportation Research Part E: Logistics and Transportation Review* 95, 32–61.
- Zhang, Y., Peng, Q., Yao, Y., Zhang, X., Zhou, X., 2019. Solving cyclic train timetabling problem through model reformulation: Extended time-space network construct and Alternating Direction Method of Multipliers methods. *Transportation Research Part B: Methodological* 128, 344–379.

- Zheng, H., Negenborn, R.R., Lodewijks, G., 2017. Fast ADMM for Distributed Model Predictive Control of Cooperative Waterborne AGVs. *IEEE Trans. Control Syst. Technol.* 25 (4), 1406–1413.
- Zhou, X., Zhong, M., 2007. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B: Methodological* 41 (3), 320–341.
- Zhou, X., Tanvir, S., Lei, H., Taylor, J., Liu, B., Rouphail, N.M., Frey, H.C., 2015. Integrating a simplified emission estimation model and mesoscopic dynamic traffic simulator to efficiently evaluate emission impacts of traffic management strategies. *Transp. Res. Part D* 37, 123–136.
- Ziliaskopoulos, A.K., Mahmassani, H.S., 1993. A Time-Dependent Shortest Path Algorithm for Real-Time Intelligent Vehicle / Highway System. *Transp. Res. Rec.* 1408, 94–100.
- Zlot, R.M., Stentz, A., 2006. An auction-based approach to complex task allocation for multirobot teams (Doctoral dissertation. Carnegie Mellon University, The Robotics Institute).