Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Rescheduling of interacting machines in automated container terminals

Jianbin Xin * **Rudy R. Negenborn** * **Gabriel Lodewijks** *

* *Section of Transport Engineering and Logistics, Department of Maritime and Transport Technology, Delft University of Technology, Mekelweg 2, 2628CD, Delft, The Netherlands*
e-mail: {j.xin, r.r.negenborn, g.lodewijks}@tudelft.nl.

Abstract: The current scheduling scheme of container terminals is typically determined off-line. This may result in delays of the complete operation, when disturbances (e.g., the operation delays or the breakdown of a machine) appear. This paper provides a method for rescheduling interacting machines in automated container terminals. The rescheduling is carried out using the current state measurements of the machines. These measurements are used to update the processing time of ongoing operations. The effect of rescheduling on both a time-efficient schedule and for an energy-efficient schedule is illustrated in a simulation study. The delay in the container handling system is reduced both for the time-efficient schedule and the energy-efficient schedule. A simulation study illustrates that the energy-efficient schedule is more sensitive to disturbances due to delays of machines than the time-efficient schedule.

Keywords: automated container terminals; rescheduling; energy efficiency.

1. INTRODUCTION

Over the last decades, there has been a significant growth of global freight transport due to the enormous commercial trade. Over 60% of worldwide deep-sea cargo is transported by containers [Stahlbock and Voß, 2007]. The management of freight transport needs to accommodate this increasing demand of containers. Intermodal transport [Crainic and Kim, 2007] is hereby considered frequently nowadays since it provides flexibility and scalability as different transport modalities can cover different areas with respect to transport distance. An intermodal container terminal represents the interface among the modalities of vessel, barge, train and truck. Therefore, container terminals play a crucial role in freight transport. This paper focuses on automated container terminals which aim to achieve a high cost-efficiency of transport process.

The research on the management of container terminals can be generally categorized into two sets of approaches: the analytical approaches and the simulation approaches. The analytical approaches address the mathematical formulation of the management problem and search for the optimal solution for the performance improvements (e.g., makespan [Chen et al., 2013]). The simulation approaches model the dynamical behavior of a container terminal by means of computer programming languages (e.g., object-oriented programming [Duinkerken and Ottjes, 2000, Bielli et al., 2006] and agent-oriented programming [Henesey, 2006]) and then can evaluate different management policies based on the available model. This paper focuses on the former approach.

As the mainstream of the analytical approaches, the scheduling problems of a container terminal have received much attention from a number of researchers due to its high dynamical and complex environment (see [Stahlbock and Voß, 2007] for a comprehensive survey). The scheduling problem is an optimization problem in which a number of jobs are assigned to available resources at particular times. In order to simplify the scheduling problem of a container terminal, several works investigate a particular area, like the quayside [Chen et al., 2012], the transport area of AGVs [Angeloudis and Bell, 2010] and the stacking area [Chen and Langevin, 2011]. However, the transport of a container depends on the interaction of multiple machines from areas all over the container terminal. This motivates the research of integrated management of larger areas [Cao et al., 2010], or even the whole terminal [Chen et al., 2013].

Although the scheduling problem has received much attention in the research of container terminals, little attention has been paid in the scientific literature to the rescheduling of machines. The planned schedule can be influenced by uncertainties in the operation of machines. The uncertainties are due to unexpected events of handling machines, e.g., the delay of handling machines or the breakdown of machines. In such circumstances, a planned schedule is probably not the optimal for the remaining handling. Rescheduling of machines can provide a better performance using updated information.

This paper aims at providing an approach for rescheduling the operations of the interacting machines in automated container terminals. In this paper we consider two types of schedules: the time-efficient schedule and the energy-

⋆ This research is supported by the China Scholarship Council under Grant 2010628012 and the VENI project "Intelligent multi-agent control for flexible coordination of transport hubs" (project 11210) of the Dutch Technology Foundation STW, a subdivision of the Netherlands Organization of Scientific Research (NWO).

efficient schedule. The time-efficient schedule refers to the schedule in which the operation time for each job is minimized, while the energy-efficient schedule indicates that the operation time for each job can be flexible for energy-efficiency purposes. The rescheduling is carried out based on the current state measurements of machines. These measurements are used to update the expected processing time of the ongoing operations. Based on the information update of the processing time, the new schedule is determined by a supervisory controller. The effect of rescheduling on both the time-efficient schedule and the energy-efficient schedule is discussed.

The remainder of this paper is organized as follows. Section 2 describes the mathematical model of the interacting machines of an automated container terminal. Section 3 proposes a rescheduling scheme using the update of the processing times. Section 4 illustrates the proposed rescheduling scheme in a simulation study. Section 5 concludes this paper and provides future research directions.

2. MODEL OF INTERACTING MACHINES

In an automated container terminal, quay cranes (QCs), automated guided vehicles (AGVs) and automated stacking cranes (ASCs) are used to transport containers between the quayside and the stacks. In this paper, one QC, multiple AGVs, and multiple ASCs are considered. The interactions of the machines determine at what time and in which sequence a number of containers are handled.

The operations of the three types of machines can be considered as a three-stage hybrid flow shop. Here we briefly summarize the model that proposed in Xin et al. [2014]. In a hybrid flow shop, each job has to pass through a number of stages. At each stage a number of identical machines can be operated in parallel to process a part of a job. Each job is processed by the same sequence of machines and each job is being handled for a certain processing time in each stage. In our three-stage flow shop, a job is defined as a complete process of transporting a container from the vessel to its stacking position.

As a three-stage hybrid flow shop, the operations of the three types of machines are described in terms of three stages:

(1) Stage 1: one QC
(2) Stage 2: multiple AGVs
(3) Stage 3: multiple ASCs

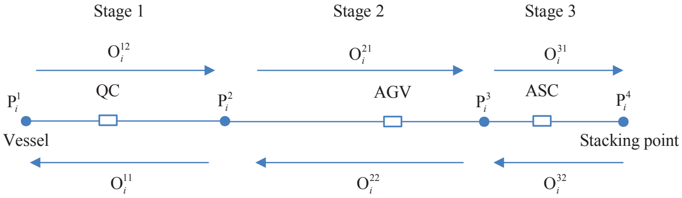

Fig. 1. The sequence of transporting containers using three types of machines.

The operations by the three types of equipment are illustrated in Fig. 1. P_i^1 is defined as the place of container i in the vessel. P_i^2 is the defined as the transfer point at which container i is transferred from a QC to an AGV. P_i^3 is defined as the transfer point at which container i is transferred from an AGV to an ASC. P_i^4 is defined as the storage place of container i in the stack.

In Stage 1, there are two operations O_i^{11} and O_i^{12}. Operation O_i^{11} is defined as the move of the QC from P_i^2 to P_i^1 for container i and operation O_i^{12} is defined as the move of the QC from P_i^1 to P_i^2 with container i. In Stage 2, there are two operations O_i^{21} and O_i^{22} in which an AGV moves from P_i^2 to P_i^3 with container i and the AGV returns from P_i^3 to P_i^2 after unloading container i, respectively. Operations O_i^{31} and O_i^{32} are defined in Stage 3, in which an ASC transports container i from P_i^3 to P_i^4 and the ASC moves from P_i^4 to P_i^3 after unloading container i, respectively.

Let there be n jobs of moving a container from vessel to stack. We define Φ to be the set of jobs (cardinality $|\Phi| = n$). We introduce two dummy jobs 0 and $n+1$ and then define $\Phi_1 = \Phi \cup \{0\}$ and $\Phi_2 = \Phi \cup \{n+1\}$ [Cao et al., 2010]. These two sets are used later in constraints for the first job and the last job. In the hybrid flow shop, the processes of each job by each machine in each stage have time relationships. For a machine to process a job in a certain stage, there is a time constraint with respect to the preceding job and the successive job. For a certain job processed in successive stages, there also exist time constraints to guarantee the sequence of stages. These time constraints can be described as follows:

$$a_i - t_i^{11} - t_i^{12} + M(1-\alpha_{0i}) \geq 0 \qquad \forall i \in \Phi \tag{1}$$

$$a_j - t_j^{11} - t_j^{12} + M(1-\alpha_{ij}) \geq b_i - t_i^{21} \qquad \forall i \in \Phi, \forall j \in \Phi \tag{2}$$

$$a_i + t_i^{21} \leq b_i \qquad \forall i \in \Phi \tag{3}$$

$$b_i + M(1-\beta_{0i}) \geq a_i + t_i^{21} \qquad \forall i \in \Phi \tag{4}$$

$$(b_j - t_j^{21}) + M(1-\beta_{ij}) \geq c_i - t_i^{31} - t_i^{32} + t_i^{22} \qquad \forall i \in \Phi, \forall j \in \Phi \tag{5}$$

$$b_i + t_i^{31} + t_i^{32} \leq c_i \qquad \forall i \in \Phi \tag{6}$$

$$c_j - t_j^{31} - t_j^{32} + M(1-\gamma_{ij}) \geq c_i \qquad \forall i \in \Phi, \forall j \in \Phi, \tag{7}$$

where, for $\forall\ i \in \Phi_1$ and $\forall\ j \in \Phi$ $(i \neq j)$,

$\alpha_{ij} = 1$ means that job j is handled directly after job i in stage 1, otherwise $\alpha_{ij} = 0$;

$\beta_{ij} = 1$ means that job j is handled directly after job i in stage 2, otherwise $\beta_{ij} = 0$;

$\gamma_{ij} = 1$ means that job j is handled directly after job i in stage 3, otherwise $\gamma_{ij} = 0$;

$t_i^{h_1 h_2}$ is the processing time of operation $\mathrm{O}_i^{h_1 h_2}$ with $h_1 \in \{1,2,3\}$, $h_2 \in \{1,2\}$;

a_i is the completion time of job i in the first stage, i.e., the time at which the QC handling job i reaches P_i^2;

b_i is the time at which the AGV handling job i reaches P_i^3;

c_i is the completion time of job i in the third stage, i.e., the time at which the ASC handling job i reaches P_i^3. Hence, $c_i - t_i^{31} - t_i^{32} + t_i^{22}$ is the completion time of job i in the second stage;

M is a large positive number.

Inequalities (1) and (4) initialize the first job processed by the QC and an AGV, respectively. Inequality (2) describes the relation among job i and j handled by the particular QC. Inequality (3) guarantees that job i is handled by an AGV after a QC. Inequality (6) guarantees that job i is handled by an ASC after an AGV. Inequalities (5) and (7) represent the relation of job i and j handled by a particular AGV and a particular ASC, respectively.

For a particular machine in each stage, it has to be guaranteed that there is exactly one preceding job and one succeeding job. For this, the discrete decision variables α_{ij}, β_{ij} and γ_{ij} have additional equality constraints. For the first job j ($j \in \Phi$) to be processed, α_{ij}, β_{ij} and γ_{ij} ($i \in \Phi, j \in \Phi, i \neq j$) must be 0, and for the last job i ($i \in \Phi$) to be processed, α_{ij}, β_{ij} and γ_{ij} ($i \in \Phi, j \in \Phi, i \neq j$) must be 0. As defined beforehand, Φ_1 and Φ_2 are used below to satisfy the additional constraints on the first job and the last job. These constraints are formulated as follows:

$$\sum_{j \in \Phi_2} \alpha_{ij} = 1, \qquad \forall i \in \Phi \tag{8}$$

$$\sum_{i \in \Phi_1} \alpha_{ij} = 1, \qquad \forall j \in \Phi \tag{9}$$

$$\sum_{i \in \Phi} \alpha_{0i} = m_1 \tag{10}$$

$$\sum_{i \in \Phi} \alpha_{i(n+1)} = m_1 \tag{11}$$

$$\sum_{j \in \Phi_2} \beta_{ij} = 1, \qquad \forall i \in \Phi \tag{12}$$

$$\sum_{i \in \Phi_1} \beta_{ij} = 1, \qquad \forall j \in \Phi \tag{13}$$

$$\sum_{i \in \Phi} \beta_{0i} = m_2 \tag{14}$$

$$\sum_{i \in \Phi} \beta_{i(n+1)} = m_2 \tag{15}$$

$$\sum_{j \in \Phi_2} \gamma_{ij} = 1, \qquad \forall i \in \Phi \tag{16}$$

$$\sum_{i \in \Phi_1} \gamma_{ij} = 1, \qquad \forall j \in \Phi \tag{17}$$

$$\sum_{j \in \Phi} \gamma_{0j} = m_3 \tag{18}$$

$$\sum_{i \in \Phi} \gamma_{i(n+1)} = m_3. \tag{19}$$

Equalities (8) and (9) represent that for each job $i \in \Phi$, there is exactly one preceding job and one succeeding job assigned to the QC. Equalities (10) and (11) guarantee that each of the m_1 QCs is employed. Equalities (12) and (13) represent that for each job $i \in \Phi$, there is exactly one preceding job and one succeeding job assigned to an AGV. Inequalities (14) and (15) guarantee that each of the m_2 AGVs is used. Equalities (16) and (17) represent that for each job $i \in \Phi$, there is exactly one preceding job and one succeeding job assigned to an ASC. Equalities (18) and (19) guarantee that each of the m_3 ASCs are in use.

Using these inequalities and equalities constraints, the discrete-event dynamics of three types of machines are modeled as a three-stage hybrid flow shop. In this hybrid flow shop, the completion time of job i processed by each stage and the sequence of jobs that are processed by each machine in each stage are decisions variables. These decision variables will be determined by the supervisory controller discussed below.

3. RESCHEDULING SCHEME

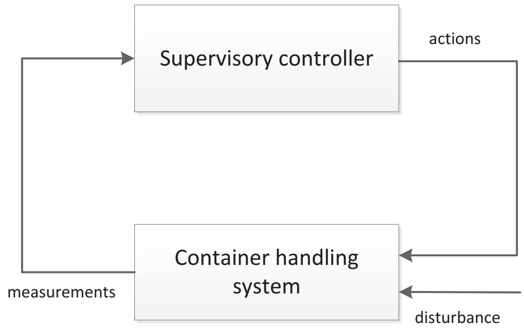

Fig. 2. The structure of the rescheduling scheme.

This section proposes a method for rescheduling interacting machines of automated container terminals as shown in Fig. 2. First, the supervisory controller of the container terminal handling system provides the solution of optimizing the scheduling of all jobs as the initial solution before starting the operation. Then after a time horizon, the supervisory controller measures the state of machines, i.e., the actual position and the actual velocity of available machines. These measurements are then used to update the minimal time needed for completing the ongoing jobs. Then the time of processing a job can be updated for the supervisory controller. The supervisory controller subsequently determines the new schedule for the interacting machines.

3.1 Supervisory controller

The goal of the supervisory controller is to determine the schedule of the interacting machines by solving an optimization problem. In this paper, we consider two types of schedules: the time-efficient schedule and the energy-efficient schedule, presented in Xin et al. [2014].

Time-efficient schedule In general, the objective of the scheduling problem is to minimize the makespan, referring to minimization of the completion time of all n jobs. In the scheduling problem of three types of machines, the makespan is defined as the maximal value of the completion time of all jobs in stage 2 by the AGVs and the completion time of all jobs in stage 3 by the ASCs. In other words, it is defined as max $\{c_1, c_1 - t_1^{31} - t_1^{32} + t_1^{22}, ..., c_n, c_n - t_n^{31} - t_n^{32} + t_n^{22}\}$, i.e., $\|\mathbf{d}\|_\infty$ where $\mathbf{d} = [c_1, c_2, \ldots, c_n, c_1 - t_1^{31} - t_1^{32} + t_1^{22}, c_2 - t_2^{31} - t_2^{32} + t_2^{22}, \ldots, c_n - t_n^{31} - t_n^{32} + t_n^{22}]^{\mathrm{T}}$ and $\|\cdot\|_\infty$ denotes the infinity norm.

The goal of the time-efficient scheduling problem we consider is to minimize the makespan subject to the discrete-event dynamics. In such a scheduling problem the time required by a machine to process a job in a particular stage is fixed [Cao et al., 2010]. After defining

$$\mathbf{a} = [a_1, a_2, \cdots, a_n]^{\mathrm{T}}$$

$$\mathbf{b} = [b_1, b_2, \cdots, b_n]^{\mathrm{T}}$$

$$\mathbf{c} = [c_1, c_2, \cdots, c_n]^{\mathrm{T}}$$

$\boldsymbol{\alpha}$: the vector of $\{\alpha_{ij}\}_{i\in\Phi_1, j\in\Phi_2, i\neq j}$

$\boldsymbol{\beta}$: the vector of $\{\beta_{ij}\}_{i\in\Phi_1, j\in\Phi_2, i\neq j}$

$\boldsymbol{\gamma}$: the vector of $\{\gamma_{ij}\}_{i\in\Phi_1, j\in\Phi_2, i\neq j}$

this time-efficient scheduling problem can be written as follows:

$$\min_{\mathbf{a},\mathbf{b},\mathbf{c},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} \|\mathbf{d}\|_\infty \tag{20}$$

subject to (1) - (7) and (8) - (19).

This time-efficient scheduling problem is formulated as a mixed integer linear programming problem. In this paper, the solver CPLEX in the OPTI toolbox [Currie and Wilson, 2012] is used to solve this optimization problem.

Energy-efficient schedule Besides the makespan, the energy consumption minimization can be considered as well as a scheduling criterion [Xin et al., 2014]. Instead of a fixed value in time-efficient schedule, the time processed by each machine can be more flexible. Due to the interaction of different types of machines, one type of machine may need to wait until another type of machine is available. However, for a given travel distance, the energy consumption of a machine can be reduced when the processing time increases so as to reduce the waiting time. Still, considering the conflict between processing time and energy consumption, the processing time of an operation by one machine depends on schedule of all machines. For a fixed travel distance, a slower operation can result in less energy consumption. In this way, the processing time of each operation can be more flexible without loss of the makespan. Therefore, the objective is to maximize the sum of the processing time of all operations subject to the minimal makespan. Here we define

$$\mathbf{t}_{\mathrm{qc}} = \left[t_1^{11}, t_2^{11}, \cdots, t_n^{11}, t_1^{12}, t_2^{12}, \cdots, t_n^{12}\right]^{\mathrm{T}}$$

$$\mathbf{t}_{\mathrm{agv}} = \left[t_1^{21}, t_2^{21}, \cdots, t_n^{21}, t_1^{22}, t_2^{22}, \cdots, t_n^{22}\right]^{\mathrm{T}}$$

$$\mathbf{t}_{\mathrm{asc}} = \left[t_1^{31}, t_2^{31}, \cdots, t_n^{31}, t_1^{32}, t_2^{32}, \cdots, t_n^{32}\right]^{\mathrm{T}}$$

$$\mathbf{t} = \left[{\mathbf{t}_{\mathrm{qc}}}^{\mathrm{T}}, {\mathbf{t}_{\mathrm{agv}}}^{\mathrm{T}}, {\mathbf{t}_{\mathrm{asc}}}^{\mathrm{T}}\right]^{\mathrm{T}}.$$

This optimization problem, which generates the energy-efficient schedule, can be rewritten as follows:

$$\max_{\mathbf{t},\mathbf{a},\mathbf{b},\mathbf{c},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} \|\mathbf{t}\|_1 \tag{21}$$

subject to

$$\min_{\mathbf{t},\mathbf{a},\mathbf{b},\mathbf{c},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} \|\mathbf{d}\|_\infty \tag{22}$$

$$s_i^{11} \leq t_i^{11}, s_i^{12} \leq t_i^{12} \tag{23}$$

$$s_i^{21} \leq t_i^{21}, s_i^{22} \leq t_i^{22} \tag{24}$$

$$s_i^{31} \leq t_i^{31}, s_i^{32} \leq t_i^{32} \tag{25}$$

and subject to (1) - (7) and (8) - (19),

$s_i^{h_1 h_2}$ is the lower bound of $t_i^{h_1 h_2}$ ($h_1 \in \{1,2,3\}$, $h_2 \in \{1,2\}$). The lower bound is obtained by the stage controller. n is the number of jobs to be processed. (21) represents that the processing time of all jobs in each stage should be maximized.

Table 1. The time windows of operations in three stages.

Operation	Machine	Starting time	Ending time
O_i^{11}	QC	$a_i - t_i^{11} - t_i^{12}$	$a_i - t_i^{12}$
O_i^{12}	QC	$a_i - t_i^{12}$	a_i
O_i^{21}	AGV	$b_i - t_i^{21}$	b_i
O_i^{22}	AGV	$c_i - t_i^{31} - t_i^{32}$	$c_i - t_i^{31} - t_i^{32} + t_i^{22}$
O_i^{31}	ASC	$c_i - t_i^{31} - t_i^{32}$	$c_i - t_i^{32}$
O_i^{32}	ASC	$c_i - t_i^{32}$	c_i

Solving the optimization problem (21) involves two steps. More details can be found in Xin et al. [2014]. Obtaining the operation time $t_i^{h_1 h_2}$ accompanied with the completion time a_i, b_i and c_i, the time windows to process job i in three stages is given in Table 1.

3.2 Rescheduling

When the rescheduling takes place, the measurements of machines are used to calculate the minimal time needed for completing the ongoing jobs. The supervisory controller can then determine the new schedule for the interacting machines. Below we discuss how the minimal time needed for completing the ongoing jobs is calculated.

Minimal-time calculation To update the processing time of ongoing jobs, the minimal time to finish the processing of ongoing jobs is required, which depends on the dynamical model and the current states of available machines. Previous approach [Xin et al., 2013] cannot be applied to the minimal-time calculation for ongoing jobs. This paper considers a numerical approach for calculating the minimum time required for ongoing jobs as presented next.

Considering the three types of machines, AGVs are considered to have two-dimensional trajectories while QCs and ASCs have one-dimensional trajectories. The minimal time required by QCs and ASCs is a particular case of AGVs because one-dimensional trajectories is the a particular case of the two-dimensional trajectories. For the sake of simplicity, here we mainly discuss the minimal time calculation of AGVs. The minimal-time calculation of QCs and ASCs can be obtained using the same methodology but in an easier way.

For each AGV, a point-mass model is used to approximate the dynamical behavior of two-dimensional space in terms of a double integrator [Richards and How, 2002]:

$$\begin{bmatrix}\mathbf{r}_p(k+1)\\ \mathbf{v}_p(k+1)\end{bmatrix} = \begin{bmatrix} I_2 & \Delta t I_2\\ 0_2 & I_2\end{bmatrix}\begin{bmatrix}\mathbf{r}_p(k)\\ \mathbf{v}_p(k)\end{bmatrix} + \begin{bmatrix}0.5(\Delta t)^2 I_2\\ \Delta t I_2\end{bmatrix}\mathbf{u}_p(k), \tag{26}$$

where AGV p has a position $\mathbf{r}_p(k) = \left[r_p^{\mathrm{x}}(k)\ r_p^{\mathrm{y}}(k)\right]^{\mathrm{T}}$ and a velocity $\mathbf{v}_p(k) = \left[v_p^{\mathrm{x}}(k)\ v_p^{\mathrm{y}}(k)\right]^{\mathrm{T}}$. Each AGV is assumed to be actuated by control actions $\mathbf{u}_p(k) = \left[u_p^{\mathrm{x}}(k)\ u_p^{\mathrm{y}}(k)\right]^{\mathrm{T}}$. I_2 is a 2×2 identity matrix. Δt is the time step. Velocity and action constraints are given by an approximation to reduce the nonlinearity as follows: $\forall p \in [1, ..., m_2], \forall m \in [1, ..., M]$

$$u_p^{\mathrm{x}}(k)\sin(\frac{2\pi m}{M}) + u_p^{\mathrm{y}}(k)\cos(\frac{2\pi m}{M}) \leq u_{\max} \tag{27}$$

$$v_p^{\mathrm{x}}(k)\sin(\frac{2\pi m}{M}) + v_p^{\mathrm{y}}(k)\cos(\frac{2\pi m}{M}) \leq v_{\max} \tag{28}$$

where $u_{\max}$ and $v_{\max}$ are magnitude limits on acceleration and velocity, and M is the arbitrary number for approximation.

In this minimal time calculation problem, AGV p is required to reach the target $\mathbf{r}_{p,f}$ as fast as possible from the current states $\mathbf{r}_{p,0}$. By introducing the binary variable $b_p(k)$, the minimal time of machine to finish the ongoing job can be obtained as follows: $\forall k \in [1, ..., T_p]$

$$\begin{aligned} r_p^{\mathrm{x}}(k) - r_{p,f}^{\mathrm{x}} &\leq R(1 - b_p(k)) \\ r_p^{\mathrm{x}}(k) - r_{p,f}^{\mathrm{x}} &\geq -R(1 - b_p(k)) \\ r_p^{\mathrm{y}}(k) - r_{p,f}^{\mathrm{y}} &\leq R(1 - b_p(k)) \\ r_p^{\mathrm{y}}(k) - r_{p,f}^{\mathrm{y}} &\geq -R(1 - b_p(k)) \end{aligned} \tag{29}$$

$$\sum_{k=1}^{T_p} b_p(k) = 1, \forall k \in [1, ..., T_p] \tag{30}$$

$$\mathbf{r}_p(0) = \mathbf{r}_{p,0}, \tag{31}$$

where T_p is the initial value as input to calculate the minimal time of machine p, R is the large and positive number to guarantee the constraints in (29) are active only when $b_p(k) = 1$. Equation (29) and (30) force the position $\mathbf{r}_p$ of vehicle from the current state $\mathbf{r}_{p,0}$ to reach the target $\mathbf{r}_{p,f}$ in the condition $b_p(k) = 1$.

If we define $t(k)$ as the elapsed time at time k ($t(k) = k$), then $t(k)b_p(k)$ can describe the finishing time if $b_p(k) = 1$. Therefore, the minimal time for transporting container i can be obtained by minimizing the sum of finishing times according to different $\mathbf{r}_f$ as follows:

$$\min_{\mathbf{u},\mathbf{b}} \sum_{k=1}^{T} t(k)b_p(k), \tag{32}$$

subject to (26), (27), (28), (29), (30) and (31),

where $t(k)$ is the elapsed time at time k, $\mathbf{u}$ and $\mathbf{b}$ are continuous and binary control variables of the optimization problem (32), respectively. The value of the objective function in (32) gives the minimal time of transporting container j, i.e., s_i^{21} and s_i^{22}.

Update for optimization Based on the minimal-time calculation, we can update the minimal time required for processing job i by different types of machines while ongoing operations as follows:

$$\begin{aligned} s_i^{11} = \hat{s}_i^{11} + e_i^{11}, s_i^{12} = \hat{s}_i^{12} + e_i^{12} \\ s_i^{21} = \hat{s}_i^{21} + e_i^{21}, s_i^{22} = \hat{s}_i^{22} + e_i^{22} \\ s_i^{31} = \hat{s}_i^{31} + e_i^{31}, s_i^{32} = \hat{s}_i^{32} + e_i^{32} \end{aligned} \tag{33}$$

where $s_i^{h_1 h_2}$ are the update of the minimal time of operation $\mathrm{O}_i^{h_1 h_2}$if $\mathrm{O}_i^{h_1 h_2}$ is ongoing, $\hat{s}_i^{h_1 h_2}$ is the minimal time to finish processing operation $\mathrm{O}_i^{h_1 h_2}$ if operation $\mathrm{O}_i^{h_1 h_2}$ is ongoing, $e_i^{h_1 h_2}$ the elapsed time of operation $\mathrm{O}_i^{h_1 h_2}$if operation $\mathrm{O}_i^{h_1 h_2}$ is ongoing.

In addition to the update of the minimal time to process ongoing jobs, the sequence of jobs to process in each stage α_{ij}, β_{ij} and γ_{ij} can be updated to determine the remaining jobs in three stages by solving the scheduling problem with the updated information.

4. SIMULATION EXPERIMENTS

To illustrate the effect of the proposed approach for rescheduling of interacting machines, we use a benchmark system [Xin et al., 2014] as a case study.

4.1 Benchmark system

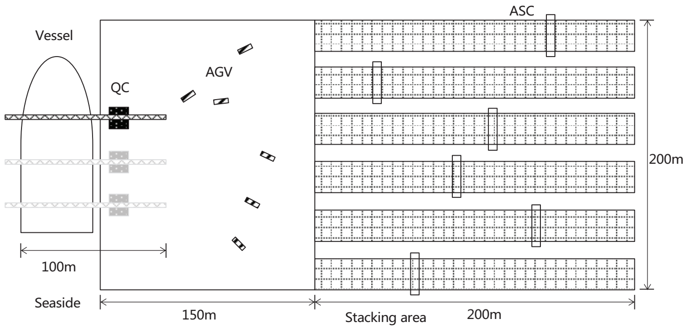

Fig. 3. The layout of a container terminal benchmarking system.

A container vessel, three QCs and six stacks are considered in this benchmark system. One ASC is installed per stack. The features of this benchmark system are given as follows:

- The distance between the furthest container and the interchange point of the QC is 100 meters;
- The quayside transport area is 150 m × 200 m;
- Each stack has a length of 36 TEU, a width of 10 TEU and a height of 6 TEU for capacity;
- The maximum speed of the QC, AGVs and ASCs are assumed to be $v_{\max}^{\mathrm{qc}} = 4$ [m/s], $v_{\max}^{\mathrm{agv}} = 6$ [m/s] and $v_{\max}^{\mathrm{asc}} = 4$ [m/s], respectively;
- The maximum acceleration of the QC, AGVs and ASCs are assumed to be $u_{\max}^{\mathrm{qc}} = 0.4$ [m/s^2], $u_{\max}^{\mathrm{agv}} = 1$ [m/s^2] and $u_{\max}^{\mathrm{asc}} = 0.4$ [m/s^2], respectively;
- Each machine can only transport one container at a time;
- Each AGV is free ranging(i.e., it does not move over predefined tracks).

4.2 Scenario

We choose 1 QC, 2 AGVs and 3 ASCs for transporting 8 inbound containers as a scenario to illustrate the effect of rescheduling. 8 containers are considered to be a horizon for handing containers. Several assumptions are made in this scenario:

- The initial position of the QC is set to its unloading position. The initial position of the AGVs is set to its loading position. The initial position of the ASCs is set to its loading position;
- 8 containers are considered arriving ar the same time;
- The processing time of one container by the QC depends on the specific position away from the quayside;
- Each container considered has the same vertical position in the vessel;
- The storage location of each container to be transported is generated randomly;
- The containers are stored in different storage places of each stack;

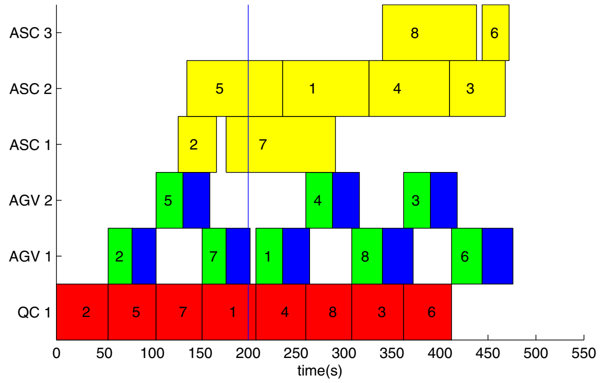

Fig. 4. The original time-efficient schedule (vertical line indicates rescheduling).

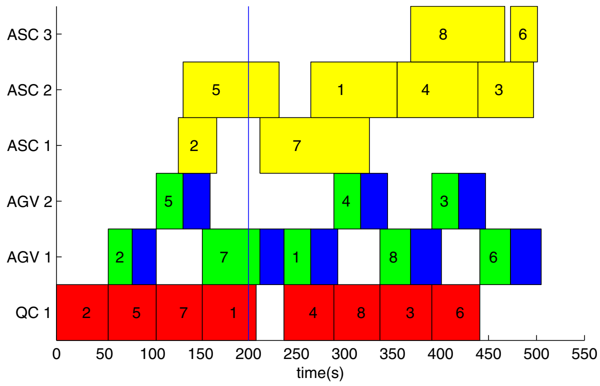

Fig. 5. The result of the time-efficient schedule without reschedule (vertical line indicates rescheduling).

- For the sake of simplicity, the service time of the QC, the AGV and the ASC are ignored;
- There is a 35-second delay from AGV 1 when it was transporting container 7 from the transport area to the stacking area.
- The horizon of scheduling is 200s.

In this scenario, when the rescheduling happens each machine is handling a container or ready to handle a container; meanwhile there are several containers that are left in the vessel to be transported. Here we discuss two types of schedule: the time-efficient schedule and the energy-efficient schedule as described in Section 3.3.

4.3 Result

Fig. 4 gives the original plan of the time-efficient schedule. Fig. 5 and Fig. 6 show the scheduling result of the considered scenario when it comes to the time-efficient schedule. It can be seen from the comparison of Fig. 5 and Fig. 6 that the reschedule could eliminate the delay of interacting machines. The delay elimination results from the waiting time of machines before the next transfer of a container using two different machines. Fig. 6 indicates the reschedule can change the schedule adaptively by means of minimizing the makespan, when there is a delay resulting from AGV1.

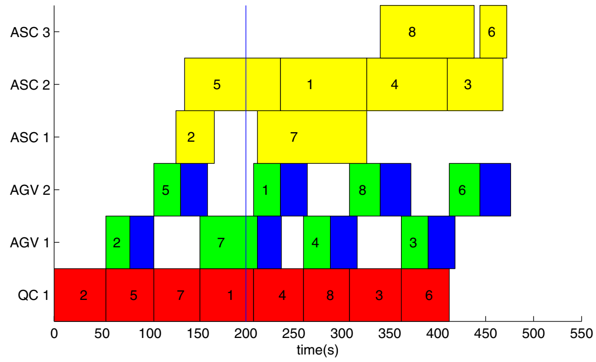

Fig. 6. The result of the time-efficient schedule with reschedule (vertical line indicates rescheduling).

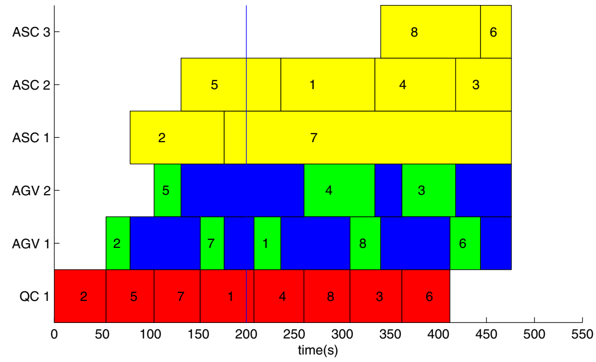

Fig. 7. The original energy-efficient schedule (vertical line indicates rescheduling).

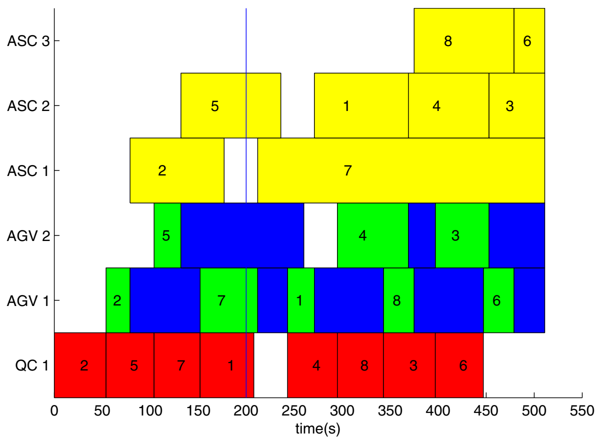

Fig. 8. The result of the energy-efficient schedule without reschedule (vertical line indicates rescheduling).

Fig. 7 presents the original plan of the energy-efficient schedule. Fig. 8 and Fig. 9 illustrate the scheduling result of the considered scenario with regards to the energy-efficient schedule. Comparing Fig. 8 and Fig. 9, the rescheduling can reduce the delay of interacting machines for the energy-efficient schedule. When there was a delay result from AGV1 from the transport area to the stacking area, AGV2 adapted to finish its ongoing operation as soon as possible for the compensation of makespan.

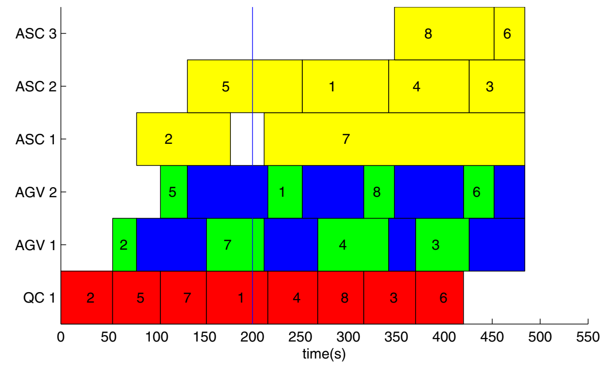

Fig. 9. The result of the energy-efficient schedule with reschedule (vertical line indicates rescheduling).

Table 2. The makespan of all simulation experiments.

options	time-efficiency	energy-efficiency
original	476s	476s
without rescheduling	505s	515s
with rescheduling	476s	484s

Moreover, the rescheduling can still generate the energy-efficient schedule when the new schedule is made.

Table 2 compares the differences of the makespan between the result without rescheduling and with rescheduling for the time-efficient schedule and the energy-efficient schedule. In general, the rescheduling of the time-efficient schedule can provide a better reduction of delays than the rescheduling of the energy-efficient schedule because the latter has tight interaction between two different machines.

5. CONCLUSIONS AND FUTURE RESEARCH

This paper proposes an approach for rescheduling the interacting machines of automated container terminals. The rescheduling is carried out based on the current state measurements of machines. These measurements are used to update the processing time of the ongoing operations. We tested the effect of rescheduling both for a time-efficient schedule and an energy-efficient schedule with rescheduling. The delay of the container handling system was reduced both for the time-efficient schedule and the energy-efficient schedule. The simulation indicates that the energy-efficient schedule is more sensitive to the delay of machines.

Future research will consider the rescheduling of a larger scale system in which more QCs, AGVs and ASCs are involved. Also, a simulation tool integrating the simulation and optimization in terms of a rolling horizon control will be developed.

REFERENCES

P. Angeloudis and M. G. H. Bell. An uncertainty-aware AGV assignment algorithm for automated container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):354–356, 2010.

M. Bielli, A. Boulmakoul, and M. Rida. Object oriented model for container terminal distributed simulation. *European Journal of Operational Research*, 175(3):1731–1751, 2006.

J. X. Cao, D. Lee, J. H. Chen, and Q. Shi. 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, 2010.

J. Chen, D. Lee, and J. Cao. A combinatorial benders' cuts algorithm for the quayside operation problem at container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):266–275, 2012.

L. Chen and A. Langevin. Multiple yard cranes scheduling for loading operations in a container terminal. *Engineering Optimization*, 43(11):1205–1221, 2011.

L. Chen, A. Langevin, and Z. Lu. Integrated scheduling of crane handling and truck transportation in a maritime container terminal. *European Journal of Operational Research*, 225(1):142–152, 2013.

T. G. Crainic and K. H. Kim. Intermodal transportation. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science*, volume 14 of *Transporation*, pages 467–536. Elsevier, Amsterdam, The Netherlands, 2007.

J. Currie and D. I. Wilson. OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User. In N. Sahinidis and J. Pinto, editors, *Foundations of Computer-Aided Process Operations*, Savannah, Georgia, 2012.

M. B. Duinkerken and J. A. Ottjes. A simulation model for automated container terminals. In *Proceedings of the Business and Industry Simulation Symposium*, volume 10, pages 134–139, Washington D.C., April 2000.

L. E. Henesey. *Multi-Agent Systems for Container Terminal Management*. PhD thesis, Blekinge Institute of Technology, Sweden, December 2006.

R. Richards and J. P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the 2002 American Control Conference*, volume 3, pages 1936–1941, Anchorage, Alaska, 2002.

R. Stahlbock and S. Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52, 2007.

J. Xin, R. R. Negenborn, and G. Lodewijks. Hierarchical control of equipment in automated container terminals. In *Proceedings of the 4th International Conference on Computational Logistics*, pages 1–17, Copenhagen, Denmark, September 2013.

J. Xin, R. R. Negenborn, and G. Lodewijks. Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control. *Transportation Research Part C: Emerging Technologies*, 2014.