Scheduling double girder bridge crane with double cycling in rail based transfer automated container terminals

Dandan Wang

School of Computer Science & Engineering,

Southeast University, Nanjing, P.R. China

Anne Goodchild

Department of Civil & Environmental Engineering,

University of Washington, Seattle, U.S.A,

Xiaoping Li

School of Computer Science & Engineering,

Southeast University, Nanjing, P.R. China

Email: dandanw@uw.edu

1 Introduction

In automated container terminals, rail based horizontal transfer systems are newly proposed and regarded to be more suitable to intermodal transportation [1]. However, improvements are required in operations scheduling in rail based transfer automated container terminals (RBT-ACT) to take advantage of the infrastructure improvement [2].

In this paper a double girder bridge crane (DGBC) is introduced, whose benefits can be obtained with modest investments, such as combining the existing twin 40-ft double trolley container cranes with a double girder [3]. Each girder has one independent spreader, and the two spreaders work on containers in adjacent bays simultaneously with no change to the safety distance constraints. As a result, operating costs are reduced, protentical collision of QCs can be avoided and the vessel service time is reduced.

Most research in this area aims to minimizing crane cycles, not processing times [4], however is it processing time that is of ultimate interest [5]. Our objective is to minimize total processing time, and the sequence dependent setup time is considered [6]. It is well established that double cycling can greatly improve quay crane productivity [7], and we consider its performance in the scheduling strategy for DGBC.

2 Problem description

2.1 Problem setting

Ordinarily, one unloading/loading operation is divided into two parts: one is moving and the other is lifting. The former can be executed automatically by the spreader. However, the latter requires the driver manually control the spreader. Thus, spreaders of DGBC work similar to ordinary quay cranes.

In order to raise or lower a container, the spreader first moves to the assigned location. The full movement YV denotes that the spreader carrying the container moves from the shore (Y) to the vessel (V) while VY denotes the reverse. Then the empty movements $\overline{YV}$ and $\overline{VY}$ imply that only the spreader itself moves between the yard and vessel. In addition, the empty movement VV represents within the vessel in the double cycling strategy.

2.2 Notation

B	: container bay, $B=1,2$	s_i	: the start time of the node i
n_B	: number of nodes in the bay B	o_{ij}	: the setup time between the activity i and j
n	: number of total nodes $n=n_1+n_2$	V	: set of nodes, $\lvert V\rvert=n$
Ψ	: set of movements, $\Psi=\{YV,VY,VV,\overline{VY},\overline{YV}\}$	O	: set of setup activities
Φ	: set of lifting activities, $\Phi=\{U,L\}$	E	: set of edges of the network
p_l	: moving time of type $l\in\Psi$	b_i	: blocking of i between moving and lifting
H	: the unary resource driver for the lifting	$C_{\max}^B$	: makespan of the bay B
p_i	: processing time of lifting operation for activity i	r_{ik}	: requirement for resource $k\in R$ by activity / setup i
π	: the permutation of the whole nodes	R	: set of all resources $R=\{Q_1,Q_2,H\}$
π^B	: the permutation of the nodes in bay B	σ	: Large number

Decision variables:

$$X_{ij}^l=\begin{cases}1 & \text{, if setup between activity } i \text{ and } j \text{ is of the movement type } l\in\Psi\\ 0 & \text{, otherwise}\end{cases}$$

$$Y_i^c=\begin{cases}1 & \text{, if the activity } i \text{ is of the lifting type } c\in\Phi\\ 0 & \text{, otherwise}\end{cases}$$

$$Z_{ij}=\begin{cases}1 & \text{, if the activity } i \text{ precedes the activity j}\\ 0 & \text{, otherwise}\end{cases}$$

$$\pi=\begin{cases}(\pi_{[1]}^1,\pi_{[1]}^2,\pi_{[2]}^1,\pi_{[2]}^2,\ldots,\pi_{[n_1]}^1,\pi_{[n_1]}^2,\pi_{[n_1+1]}^2,\ldots,\pi_{[n_2]}^2) & ,\ n_1\le n_2\\ (\pi_{[1]}^1,\pi_{[1]}^2,\pi_{[2]}^1,\pi_{[2]}^2,\ldots,\pi_{[n_2]}^1,\pi_{[n_2]}^2,\pi_{[n_2+1]}^1,\ldots,\pi_{[n_1]}^1) & ,\ n_1>n_2\end{cases}$$

2.3 Graph definition

The project can be described by an activity-on-node graph *G* (*V*, *E*). The set of nodes $\lvert V\rvert=n$ corresponds to the *n* activities. The nodes can be further divided into two subsets $\lvert V_B\rvert=n_B$, *B=1,2*, in which V_B is the nodes set in each bay *B*, and $V_1\cup V_2=V\wedge V_1\cap V_2=\Phi$. One unloading/loading activity needs exactly one lifting operation. Arc set $E=\{(i,j):i,j\in V;i\rightarrow j\}$ represents the temporal precedence constraints between two activities, i.e. $i\rightarrow j$ if activity *i* must finish before activity *j* can start. Adjacent lifting operations are separated by a series of movements. Spreader movements before/after each lifting operation are defined as the sequence dependent setup o_{ij}, which must be required by the consequently scheduled activities in the same bay. In addition, dummy activities *0* and *n+1* with zero duration are added to make sure only one starting and one finishing node in *G*. Each node is characterized by its processing time, resource requests (Q_1, Q_2, H) and precedence relations with other activities.

There are three resources: one driver H and two spreaders Q_1 & Q_2. Driver is the dedicated resource [8] only required for lifting operation, while spreaders are the allocable resource [9] used in lifting and movement. All the three are unary resource [10,11] with the available amount 1. In detail, Q_1/Q_2 serves bay 1/2, and the capacity of each is 1.

2.4 Setup modes

A series of movements may be executed between the adjacent lifting activities. Unloading (*U*)/ loading (*L*) in the single or double cycling strategy usually require their own combination of movements. Therefore, the setup has four movement modes (o^1~o^4) according to the sequence of the lifting operations [12], as listed in Table 1.

Table 1 Four movement modes

Strategy	lifting	movements	lifting
Single cycling	U	$o^1 = VY + \overline{YV}$	U
	L	$o^2 = \overline{VY} + YV$	L
Double cycling	U	$o^3 = VY + YV$	L
	L	$o^4 = VV$	U

Setups may overlap with different required resources at the same time (Q_1 and Q_2 can be in parallel). For description convenience, setups are treated as the additional activities denoted as *O*. Each setup can be depicted as a virtual node o_{ij} inserted between the original defined activity *i* and *j* in *G*. As a result, setup activity has the duration of setup time o_{ij} and the demand of the allocatable resources (spreaders corresponding to the bay).

2.5 Mathematical model

The problem is defined as resource-constrained project scheduling problem with sequence dependent setup [13], characterized by directed acyclic graph, and formulated into an integer programming model. The formulation is excluded here for brevity.

3 Methodology

In order to minimize the completion time of two bays, a two step heuristic is proposed. Firstly, double cycling is used for each bay since it achieves better crane processing efficiency than single cycling. However, because there is only one driver in charge of all the lifting operations on both two spreaders with DGBC. Two spreaders cannot be treated as two independent cranes. As a result, there exist resource conflicts between two double cycling schedules, in which one spreader cannot perform lifting directly after moving, and has to wait for the driver released from the previous lifting with the other spreader. Therefore, a timetabling heuristic is presented after the double cycling procedure to settle the conflicts. The double cycling procedure can be transferred from the traditional method in [7], then the emphasis will be focused on the timetabling step. Two step heuristic is described as below:

1. Scheduling B_1 and B_2 in double cycling. Obtain π^1 and π^2.
2. Compute the double cycles D and C in π^1 and π^2.
3. Mix π^1 and π^2 basing on FirstComeFirstServe to form the initial timetable π_0.
4. Fix one schedule of the bay B with $D_B \geq D_{B'}$ and $C_B \geq C_{B'}$.
5. While ($D_{B'} \geq 0 \vee C_B \geq C_{B'}$) // $B' \neq B$
    5.1 iteratively local search
        5.1.1 right shift activities to remove conflicts
        5.1.2 make the schedule π_1 more tighter, and update $C_{B'}$.
        5.1.3 If better than π_0, then replace.
    5.2 change π_1 by $D_{B'} - 1$.
6. Alter the permutation and timetable by FirstComeFirstServe to solve the rest conflicts.

4 Results and Discussion

We assume two typical quay crane respectively serve the two bays in single cycling. Then the maximum completion time of this traditional transportation is *180(n1+n2)*. By contrast, single cycling is uesd in both spreaders of DGBC, and the driver is scheduled for two bays assuming that the spreader that first arrives at the lifting position will be first served. Therefore, the maximum completion time is $\max\{n_1, n_2\}180 + 60$. Furthermore, we consider a specific case that is DGBC cooperates one spreater (B_1) in single cycling with another (B_2) in double cycling. There are 7 different resource conflicts (one driver) between two bay schedules. Through the experiments, we find that all conflicts in one case will coexist periodically in three styles as (1, 4, 7), (2, 5, 8) or (3, 6, 9). Moreover, the latter two styles can be transformed into (1, 4, 7). Therefore, the maximum completion time of this specific case can be reduced as $\max\{C_{B2} + 50 * NumberOfConflicts, C_{B1} + 60 * NomberOfConflicts\}$.

Because double cycling greatly facilitates one bay operation, DGBC with both two spreaders in double cycling is considered. According to the presented method, the crane utilization rate is 69% higher than typical cranes. Various scenarios with different degrees of double cycling are investigated to explore its impact on crane processing efficiency. The results demonstrate that crane efficiency can be improved up to 32%. However, the best DGBC productivity does not require the highest degree of double cycling in both bays. The most important point is the cooperation between two bays schedules which results in less blocking. As the result, DGBC can significantly improve terminal productivity, and using double cycling can further enhance processing efficiency.

References

[1] Z. Lu, H. L. Loo, P. C. Ek, C. Dao-Fang, and X. Zhi-Xiang, "A Comparative Study on Two Types of Automated Container Terminal Systems", *IEEE Transactions* on *Automation Science and Engineering*, 9, 56-69, 2012.

[2] A.V. Goodchild, J.G. McCall, J. Zumerchik and J. Lanigan, "Reducing Train Turn Times with Double Cycling in New Terminal Designs", *Transportation Research Record*, 2238, 8-14, 2011.

[3] R. Stahlbock, and S. Voss, "Operations research at container terminals: a literature update". *OR Spectrum*, 30(1), 1-52, 2008.

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

[5] F. Meisel and M. Wichmann, "Container sequencing for quay cranes with internal reshuffles". *OR Spectrum*, 32(3), 569-591, 2010.

[6] M. Mika, G. Waligora and J. Weglarz, "Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times," *European Journal of Operational Research*, 187, 1238-1250, 2008.

[7] A.V. Goodchild, C.F. Daganzo, "Double-cycling strategies for container ships and their effect on ship loading and unloading operations", *Transportation Science* 40(4), 473–48, 2006.

[8] L. Bianco, P. Dell'Olmo and M. G. Speranza, "Heuristics for multimode scheduling problems with dedicated resources," *European Journal of Operational Research*, 107, 260-271, 1998.

[9] C. Schwindt and N. Trautmann, "Scheduling the production of rolling ingots: industrial context, model, and solution method," *International Transactions in Operational Research*, 10, 547-563, 2003.

[10] R. Capek, P. Sucha and Z. Hanzalek, "Production scheduling with alternative process plans", *European Journal of Operational Research*, 217, 300-311, 2012.

[11] U. Dorndorf, E. Pesch and T. Phan-Huy, "Constraint propagation techniques for the disjunctive scheduling problem", *Artificial Intelligence*, 122, 189-240, 2000.

[12] M. Mika, G. Walig O Ra and J. Weglarz, "Modelling setup times in project scheduling", *Perspectives in Modern Project Scheduling*, 92, 131-163, 2006.

[13] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem," *European Journal of Operational Research*, 207, 1-14, 2010.