Scheduling the operations of a double-load crane in slab yards

Guodong Zhao[a]*, Jiyin Liu[b]*, Yun Dong[a]

a. Institute of Industrial and Systems Engineering, Northeastern University, Shenyang, 110819, China

b. School of Business and Economics, Loughborough University, Leicestershire, LE11 3TU, UK

Abstract

This paper studies a double-load crane scheduling problem (DLCSP) in steel slab yards. A slab yard stores slabs in stacks. To prepare for use in production, some slabs need to be moved from one place to another. These movement tasks are performed by a double-load crane which can hold up to two slabs simultaneously. Given a set of tasks and possibly precedence relationship among them, the scheduling problem is to allocate the tasks to double-load operations and determine the schedule for the crane to perform the tasks so as to minimize the makespan. The problem is first formulated as a mixed integer linear programming (MILP) model with variables representing the order of tasks. Based on properties of the problem, it is then reformulated from a crane operation perspective. Computational experiments are carried out on practical data collected from a steel company. The results show that both models can solve practical sized problems optimally, with the second model being more efficient.

Keywords: Slab yard, Crane scheduling, Double-load crane, MILP model

*Corresponding authors. Email: gascoigne_0509@163.com, j.y.liu@lboro.ac.uk

1. Introduction

The slab yard in a steel production system is a warehouse for storing slabs that come from the continuous casting process and will be used as input of the rolling processes. Operation efficiency of the slab yard has great influence on the production stages before and after it. On one hand, it can ensure smooth output of the slabs from the continuous casting process, reducing the chance of blocking. On the other hand, it can provide raw materials in time and guaranteeing the continuous production of rolling process. The slab yard serves as a very important buffer in the whole steel production system. Bridge cranes are used for handling these slabs in the yard. Similar to those used in most heavy industry, the bridge crane consists of a horizontal beam that runs along an overhead rail track and a hoist that moves along the beam and can lift up and place down slabs. We will refer to the hoist of the crane as the crane for simplicity. The latest crane can hold up to two slabs simultaneously, under certain condition, e.g., the slab underneath must be wider than, or roughly the same as, the above slab. While such new cranes have higher productivity, their operations become more complex and effective scheduling is essential to achieve their potentials. This paper studies the problem of scheduling such new cranes that can hold two slabs. We will call it double-load crane scheduling problem. A crane task is a required move of a slab from one position to another in the yard. Given a set of tasks and possible precedence relationships among them, the scheduling problem for one such crane is to determine the schedule for the crane to perform the tasks so as to minimize the makespan.

Figure 1 shows the typical layout of a slab yard of a hot rolling plant. This example yard has two storage halls and four cranes, Cranes 1 and 2 in Hall 1, Cranes 3 and 4 in Hall 2. The operations in different halls are independent to each other, and their crane scheduling problems can be considered separately. Cranes in the same hall cannot cross each other and a minimum safety distance must be maintained between any two adjacent cranes. In current practice, each crane is allocated to a designated area in the slab yard, and operates above this fixed area. Thus, it is sufficient to solve the problem of one crane in each area. For simplicity, we refer to the storage area being studied as the slab yard, from here on wards. This paper studies this single crane scheduling problem in a steel company.

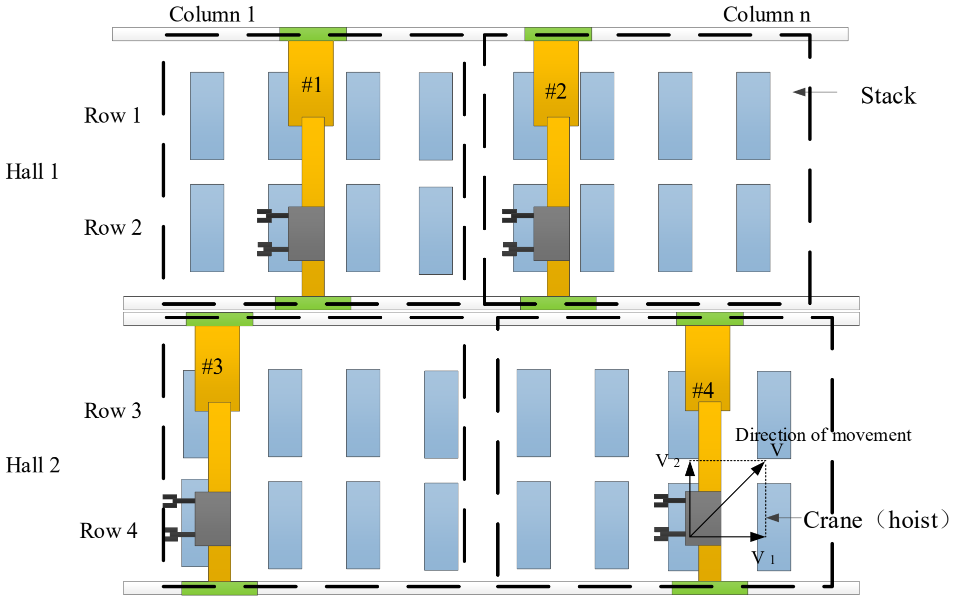

Figure 1 Crane in the slab yard

Crane scheduling problem (CSP) with overlapping working area has received a lot of studies, such as crane cyclic scheduling in automated electroplating lines and quay crane scheduling problem in container yard, and so on. However, crane scheduling in warehouses storing slabs has received little attention.

Aron et al. (2011) proposed a dynamic programming (DP) algorithm to solve a CSP. In this problem, each job has been pre-assigned to a crane. The DP mainly focuses on obtaining the optimal crane trajectories. While in our problem, tasks could be interrupted and merge with another task. The task allocation and combination are decision variables. So, the proposed DP is not applicable to our problem considered here.

Another kind of similar crane scheduling problem is in electroplating line. For example, Zhou et al. (2008) investigated the electroplating line processing identical products cyclically. Each part should visit a given sequence of tanks and the movements are performed by hoists. The objective is to maximize the throughout or equivalently minimize the cycle length. They proposed a heuristic algorithm to solve the problem. The main differences between electroplating scheduling problem and our problem are that our problem is not a cyclic scheduling problem, and the routine of each job (referring to 'part' in electroplating line) is not given in advance.

There are previous studies on a single-crane scheduling problem in steel industry. Xie et al. (2015) studied the problem of scheduling crane operations in a steel coil warehouse to retrieve all required coils to their designated positions in the shortest possible time (makespan). Since the problem is shown to be NP-hard, a genetic algorithm (GA) is proposed. Zapfel et al. (2006) investigated a single crane scheduling problem in a steel coil distribution center that needs to store incoming coils and retrieve coils required by customers. The problem was viewed as a job shop scheduling problem and formulated as a nonlinear integer programming model, which is hard to solve. A local search-based heuristic was proposed and tested through computation. Rei et al. (2008) considered a single crane scheduling problem to store and retrieve steel items with known arrival and retrieval dates to minimize the number of crane movements. The items were stacked one on top of another in a similar way to container stacking. A simulation-based heuristic was proposed to solve the problem. Tang et al. (2009) studied a single-crane scheduling problem in a steel annealing system where the main decisions were to assign furnaces to steel stacks for processing and scheduling the crane to move the furnaces around the plant. There were no shuffling operations involved and the problem was solved using a sequential approach that assigned the furnaces first and then schedule the crane. Dohn and Clausen (2010) studied a slab yard planning problem to decide the slab movements and a crane scheduling problem to carry out the movement tasks. This paper split the problem to two stages which were solved sequentially using greedy heuristic and local search, respectively. Ge and Yih (1995) studied crane scheduling with time windows in circuit board production lines. Fleury et al. (2001) also considered the time window constraints in industrial systems, and they further considered random events in the scheduling problem.

Apart from these, there is relevant research on crane scheduling problems in container yard. Kim et al. (2006) mentioned a branch-and-bound procedure and presented a heuristic for determining the storage positions for shuffled containers during the process of retrieving a given sequence of export containers to minimize the number of shuffles. Wan et al. (2009) developed a linear integer programming model for container retrieval and shuffling problems and also presented heuristics for handling container storage and retrieval in both static and dynamic settings. Lee et al. (2010) presented a three-phase heuristic for retrieving containers in a given sequence to minimize the weighted sum of the number of container movements and the crane's working time. Lee et al. (2009) solved larger instances of the pre-marshalling export problem using neighborhood search heuristics. Li et al (2009) developed an efficient discrete model for yard crane (YC) scheduling by considering realistic operational constraints such as inter-crane interference, fixed YC separation distances and simultaneous container storage/retrievals. Li et al. (2012) developed an efficient continuous time MILP model for YC scheduling, which considers constraints about multiple inter-crane interference, container inbound and outbound, as well as yard spacing and priority constraints. Liu et al. (2006) studied the problem of scheduling quay cranes at container terminals where incoming vessels have different ready times. They decomposed and solved the problem in two levels, scheduling given numbers of cranes for each vessel at the vessel level and allocate cranes at the terminal level. A unidirectional movement formulation for the vessel level problem was proposed. Chen et al. (2014) also considered unidirectional movement in a cluster-based quay crane scheduling problem and developed a compact mathematical formulation considering crane initial positions. Zhang et al. (2015) modelled a problem of scheduling both quay cranes (QC) and yard cranes (YC) at container terminals and solved it using a bi-level genetic algorithm. At the lower level, the stowage plan of the outbound containers and the YC schedule are determined to minimize the operating time of YCs, while at the higher level, the total operating time of both QC and YC is minimized by optimizing the loading and unloading sequence for the hatches and stacks in each ship-bay, considering double cycle operation of the QCs. Zheng et al. (2018) studied a single yard crane scheduling problem with uncertain release times of retrieval tasks in a container yard. They proposed a two-stage stochastic programming model, and developed a heuristic algorithm to solve the problem. The first stage problem was the general single crane scheduling problem for container yard, then in the second stage they considered the retrieval information on the release times. Guo et al. (2018) studied a crane scheduling problem in a railroad container terminal. In this problem each task needs to be performed by a gantry crane at one fixed position, while the hoist loading the containers to truck/railcar or unloading from truck/railcar.. In addition, the gantry crane can move only one container in one task. These are different from the tasks in the problem studied in this paper. For more research on container terminal operations, please see an overview by Stahlbock and Voß (2008).

In all the above studies, a crane can only handle one item, such as a steel coil, a container or a furnace at one time. In this paper, the crane can hold two slabs simultaneously, and so two slab-moving tasks can be combined in one loaded trip. This new feature can potentially increase productivity but also make the problem more complicated.

The following three references consider double operations. Upadhyay et al. (2017) presented a new mathematical model for optimizing the loading of double-stack container trains. The problem is to allocate containers to each carriage where containers can be stacked two-level high. Mantovani et al. (2018) proposed a model that can deal with single- or double-stack railcars as well as arbitrary containers-to-cars matching rules. The objective is to choose the optimal subset of containers and the optimal way of loading them on outbound railcars so as to minimize the resulting loading cost. The decisions of the above two problems are actually on space assignment rather than crane or train scheduling. Lai et al. (2013) studied the problem of transporting goods in containers from a port to importers and from exporters to the port by trucks that can carry one or two containers, without separating the truck and containers during the trip. All the goods in the same container are sent to (or collected from) one customer location. Essentially, each truck has the capacity of one or two containers. The starting point and ending point of each trip are the same terminal, and so the trips are independent from each other. The decision of this problem is to allocate customers to truck trips. Each trip may have up to four nodes and so the routing is very simple once the customers are assigned to truck trips. In our problem, the initial point and target point of each task are different, and so the situation is much more complex. We need not only to decide task combination, but also to decide the overall crane route to perform all the tasks.

We study this double-load crane scheduling problem and optimize the operations. In the following part of the paper, we first describe the problem in more detail in the next section. Then two MILP models are formulated in section 3. Section 4 reports the experiments carried out on data collected from a steel company. Conclusions are drawn in section 5.

2. Problem description

The double-load crane can perform one task at a time (single-load operation) or perform two tasks together (double-load operation). Depending on the positions of the two slabs involved, a double-load operation can be one of the following types.

(1) The target stacks and the initial stacks of the two slabs are all different. The double-load operation process in this case is shown in Figure 2. The crane picks up the first slab from its initial stack, moves to the initial stack of the second slab where it puts the first slab above the second one and then lifts the two slabs together, then moves to the second slab's target stack and puts down the second slab, finally moves to the first slab's target stack and puts down the first slab.

(2) The two slabs have the same initial stack but different target stack. In order to

be moved in a double-load operation, the two slabs must be in adjacent positions in the initial stack. In this case, the crane lifts the two slabs together from the initial stack, then puts the lower slab and then the higher slab at their target stacks respectively.

(3) The two slabs have different initial stacks but the same target stack. In this case, the required positions of the two slabs in the target stack must be adjacent. In the operation, the crane first lifts the slab that is to be stored at a higher position in the target stack and then picks up the other slab, from their initial stacks respectively, and finally puts the two slabs together at their target stack.

(4) The two slabs have the same initial stack and the same target stack. The two tasks can be performed in a double-load operation only if the positions of the two slabs in both stacks are adjacent and their relative positions are the same in both stacks. In this case, the crane can pick up both slabs in the initial stack and put them together at the target stack, similar to a single-load operation.

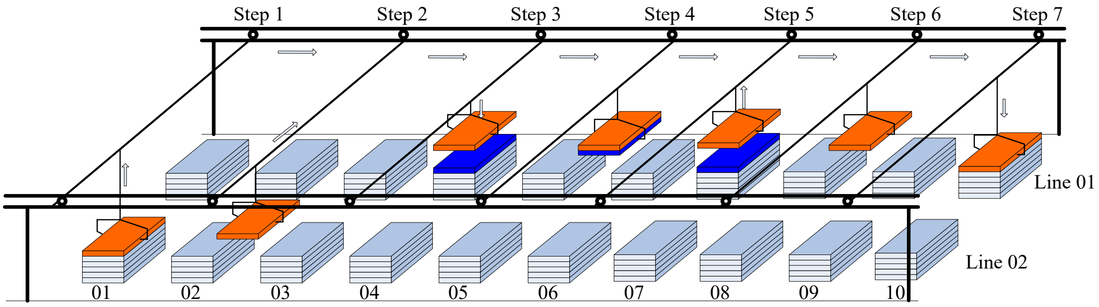

Figure 2 The process of a double-load operation

In the current practice, the tasks are generated from the ERP system according to the production plans in the production stages before and after the slab yard. The system also provides a sequence for performing the tasks considering the priorities and space relations of the slabs. However, the system provided sequence does not take full advantage of the double-load capability and the efficiency of the crane.

(1) Because the system does not consider the actual crane movements when generating the task sequence, the crane may need long empty moves from one

task's target stack to the next task's initial stack, making the total time for the crane to complete all the tasks longer.

(2) The system combines two tasks into a double-load operation only if the two slabs have similar specifications as well as the same initial and target stacks. Potential double-load operations for the tasks with different initial or target stacks are not explored, which leads to low capacity utilization of the crane.

(3) In actual operation, if there are two adjacent tasks which have the same initial (target) stack and close target (initial) stacks and the two slabs have similar specifications, the crane operator will perform these two tasks as a double-load operation. This can improve the plan generated by the system, but the improvement is limited.

We study the double-load cane scheduling problem for a given set of tasks to improve productivity. The constraints that need to take into account are:

(1) Task precedence constraints

There are several types of crane tasks including receiving and storage, rearrangement, retrieval and shuffling. In any case, a task is to move a slab from one position to another. There may be precedency requirements among the tasks according to their priorities in time and positions in space. Tasks have different emergency levels. Priority has to be given to more urgent tasks. For two tasks of moving two slabs away from the same stack, the task of moving the slab in the higher position must be performed first. Similarly, for two tasks of moving two slabs into the same stack, the task of move the slab to the lower position must be performed first.

(2) Crane transport capacity constraints

The crane can hold two slabs at most at the same time. Furthermore, a double-load operation may involve at most two tasks, as any one of the four types described at the start of this section. That is to say, after picking up two slabs from one or two stacks, the crane has to put down both slabs at their target stacks before picking up any other slab.

(3) The task integrity constraints

Each task must and can be executed for only one time by the crane.

The time a crane spends for moving from one stack to another depends on the larger value between the time for moving in row direction and that in column direction. If the x axis is the row direction of the slab yard and the y axis is the column direction, stack A (x_1, y_1) and stack B (x_2, y_2) are the coordinates of the two stacks in the slab yard, the movement speeds of the crane in row and column directions are v_r and v_c, respectively, then the duration for the crane moving from stack A to stack B will be $\max\{|x_1\text{-}x_2|/v_r, |y_1\text{-}y_2|/v_c\}$. The move time between two stacks can be calculated in advance and used as problem parameters.

Given the set of tasks to be performed, whether two tasks may be combined in a double-load operation can be determined according to the conditions described earlier. The precedence relationships between tasks can also be worked out. These can all be expressed as known problem parameters as shown in the next section.

3 Problem Formulation

3.1 Problem formulation using crane routing variables

3.1.1 Parameters

We are given a set of n tasks, $O = \{1, \ldots, N\}$, which include N slabs.

O^+ the set of tasks' original stacks, o_i^+ denote the original stack of task i.

O^- the set of tasks' target stacks, o_i^- denote the target stack of task i.

p_{ij}: a precedence parameter, equal to 1 if i must precede j, otherwise equal to 0.

q_{ij} : a parameter indicating if tasks i and j meet the double-load conditions, equal to 1 if yes, otherwise equal to 0.

$\tau_{\alpha\beta}$: the time needed for the crane holding one slab to move from stack α to stack β.

$\tau^{+}_{\alpha\beta}$: the time needed for the crane holding two slabs to move from stack α to stack β.

$\tau^{-}_{\alpha\beta}$: the time needed for the empty crane to move from stack α to stack β.

3.1.2 Decision variable

$$x_{ij}=\begin{cases}1, & \text{if task } j \text{ starts immediately following task } i,\\ 0, & \text{otherwise}\end{cases}$$

$$x_{0j}=\begin{cases}1, & \text{if task } j \text{ is the first task in the sequence,}\\ 0, & \text{otherwise}\end{cases}$$

$$y_{ij}=\begin{cases}1, & \text{if tasks } i \text{ and } j \text{ are performed together in a double-load operation,}\\ 0, & \text{otherwise}\end{cases}$$

S_i: a variable reflecting the relative order of task i according to the start time. It is used to avoid sub-tours in the crane movement.

R_{ij}: the revision or savings of travel time due to the double load combining tasks i and j.

3.1.3 Mathematical model F1

According to the target and constraints of the actual crane scheduling problem in steel enterprises, the mixed integer programming model is established:

$$\min \ \sum_{i\in O}\tau_{o_i^+ o_i^-}+\sum_{j\in O}\sum_{\substack{i\in O\\ i\neq j}}\tau^{-}_{o_i^- o_j^+}x_{ij}+\sum_{j\in O}\sum_{\substack{i\in O\\ i\neq j}}R_{ij}$$

s.t.

$$\sum_{\substack{i=0\\i\neq j}}^{N} x_{ij} = 1, \qquad j = 0,1,...,N. \tag{1}$$

$$\sum_{\substack{j=0\\i\neq j}}^{N} x_{ij} = 1, \qquad i = 0,1,...,N. \tag{2}$$

$$S_j - S_i \geq Nx_{ij} - N + 1, \qquad i,j = 1,...,N,\ i \neq j. \tag{3}$$

$$S_j - S_i \geq 1, \qquad p_{ij} = 1,\ i,j = 1,...,N,\ i \neq j. \tag{4}$$

$$\sum_{\substack{i=1\\i\neq j}}^{N} y_{ij} + \sum_{\substack{k=1\\k\neq j}}^{N} y_{jk} \leq 1, \qquad j = 1,...,N. \tag{5}$$

$$y_{ij} \leq x_{ij}, \qquad i,j = 1,...,N,\ i \neq j. \tag{6}$$

$$y_{ij} \leq q_{ij}, \qquad i,j = 1,...,N,\ i \neq j. \tag{7}$$

$$R_{ij} \geq \tau_{o_i^+ o_j^+} + \tau^{+}_{o_j^+ o_j^-} + \tau_{o_j^- o_i^-} + \sum_{\substack{k=0\\k\neq j}}^{N} \tau^{-}_{o_i^- o_k^+} x_{jk} - \tau_{o_i^+ o_i^-} - \tau^{-}_{o_i^- o_j^+} - \tau_{o_j^+ o_j^-} - \sum_{\substack{k=0\\k\neq j}}^{N} \tau^{-}_{o_j^- o_k^+} x_{jk} + M(y_{ij} - 1),$$

$$i,j = 1,...,N,\ i \neq j. \tag{8}$$

$$x_{ij}, y_{ij} \in \{0,1\}, \qquad i,j = 1,...,N,\ i \neq j. \tag{9}$$

$$S_i \geq 0, \qquad i = 1,...,N. \tag{10}$$

The objective function consists of three parts: The first part is the sum of the execution time for single-operation task. The second part is the sum of the travel time from one task to another. The third part is the sum of revisions/savings in travel time due to double-operation tasks. Constraints (1) and (2) ensure that the crane begins from a dummy task 0, performs all the tasks in a sequence according to the start time and returns to the dummy task. Constraints (3) avoid sub-tours in the sequence. Constraints (4) ensure that the required precedence relationships are satisfied. Constraints (5) mean that any task may be carried out on double-load with at most one other task. Constraints (6) indicate that if two tasks are performed by a double-load operation, these two tasks must perform successively and have fixed sequence. Constraints (7) indicate that only the tasks satisfying the conditions can be performed in a double-load operation.

Constraints (8) calculate the travel time revision/savings due to double load operations. Constraints (9) and (10) define the range of variable values.

In order to verify the validity of the mathematical model, the mathematical model was solved using standard software CPLEX. Nine sets of data were derived from the actual production records, each set consists of 10 different problem instances of the same size. The nine sets can be classified into three groups, small, medium and large. Each problem instance was solved with the maximum runtime set to 3600 seconds. In the cases where the optimal solution is not found within the time limit, the best feasible solution is recorded. Table 1 shows the statistics of results for each set of data. The third column indicates the number of instances in the set for which optimal solution was found within the time limit, the forth column gives the numbers of instances for which a feasible solution was obtained but optimality was not verified within the time limit, and fifth column shows the number of instances for which no feasible solution was found within the time limit.

Table 1. Experimental results for F1 using CPLEX

Classification	Number of tasks	Optimal solution obtained	Feasible solution obtained	Feasible solution not obtained	Average solution time (s)	The longest solution time (s)
Small scale	6	10	0	0	0.38352	0.449
	8	10	0	0	0.74732	0.897
	10	10	0	0	3.6271	3.917
Medium scale	20	10	0	0	10.6568	10.870
	30	10	0	0	59.4248	68.339
	45	10	0	0	212.709	229.726
Large scale	60	10	0	0	1009.9224	1211.907
	80	6	4	0	2633.042	3600
	100	1	9	0	3521.608	3600

As can be seen from table 1, CPLEX can find the optimal solution of all the instances of small scale and medium scale, but it can only obtain the optimal solution for part of the large-scale instances. In addition, when the number of tasks increase, the run time increases rapidly. When the number of problem size reaches 80, the program runs until the time limit 3600 seconds for some instances without obtaining an optimal solution. When the problem size is 100, only one instance is solved optimally, though feasible solutions are found for all other instances within the time limit.

3.2 Problem formulation based on crane operation steps

Because the tasks are usually given in short notice, the crane scheduling problem must be solved quickly. Our research is not only to provide effective scheduling scheme for the slab yard, but also to improve the efficiency of decision making as much as possible. Now we formulate another model to solve the problem efficiently from the equipment operations perspective. Clearly, each task involves two slab-handling steps: picking up the slab and dropping off the slab. Hence, there are totally 2*N* steps for the *N* tasks. We redefine variables to represent the sequence of these 2*N* steps and formulate constraints to ensure feasibility of the sequence. Whether two tasks are combined as a double-load operation can also be expressed using the sequence. For normal single-load operations, the lifting-up and dropping-off steps appear alternately in the sequence. A double-load operation will appear in the sequence as a series steps of lifting up two slabs followed immediately by dropping off them in reverse order.

We use $Q = \{1, \ldots, 2N\}$ to denote the set of slab handling steps. Other parameters used are the same as those for the last model. The variables for the new model are defined below.

3.2.1 Decision variables

$$z_{ik}^{+} = \begin{cases} 1, & \text{if slab } i \text{ is picked up at the crane's } k^{\text{th}} \text{ step,} \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ik}^{-} = \begin{cases} 1, & \text{if slab } i \text{ is dropped off at the crane's } k^{\text{th}} \text{ step,} \\ 0, & \text{otherwise} \end{cases}$$

$t_{k,k+1}$: the crane operation time between step *k* and step *k*+1.

3.2.2 Mathematical model F2

According to the target and constraint of the actual crane scheduling problem in steel enterprises, the mixed integer programming model is established:

$$\min \sum_{k=1}^{2N-1} t_{k,k+1}$$

s.t.

$$\sum_{i=1}^{N} z_{ik}^{+} + \sum_{i=1}^{N} z_{ik}^{-} = 1, \qquad k = 1,...,2N. \tag{11}$$

$$\sum_{k=1}^{2N} z_{ik}^{+} = 1, \qquad i = 1,...,N. \tag{12}$$

$$\sum_{k=1}^{2N} z_{ik}^{-} = 1, \qquad i = 1,...,N. \tag{13}$$

$$1 + \sum_{k=1}^{2N} k z_{ik}^{+} \le \sum_{k=1}^{2N} k z_{ik}^{-}, \qquad i = 1,...,N. \tag{14}$$

$$\sum_{i=1}^{N} \sum_{k'=k}^{k+2} z_{ik'}^{+} \le 2, \qquad k = 1,...,2N-2. \tag{15}$$

$$z_{ik}^{+} = z_{i,k+1}^{-} + z_{i,k+3}^{-}, \qquad i = 1,...,N,\ k = 1,...,2N-3. \tag{16}$$

$$\sum_{k=1}^{2N} k z_{ik}^{-} \le \sum_{k=1}^{2N} k z_{jk}^{+}, \qquad p_{ij} = 1, i, j = 1,...,N. \tag{17}$$

$$z_{ik}^{+} + z_{j,k+1}^{+} \le q_{ij} + 1, \qquad i, j = 1,...,N,\ k = 1,...,2N-1. \tag{18}$$

$$z_{jk}^{-} + z_{i,k+1}^{-} \le q_{ij} + 1, \qquad i, j = 1,...,N,\ k = 1,...,2N-1. \tag{19}$$

$$t_{k,k+1} \ge \tau_{o_i^{+},o_i^{-}}^{+} \left(\sum_{j=1}^{N} z_{j,k-1}^{+} + z_{ik}^{+} + z_{i,k+1}^{-} - 2 \right), \qquad i, j = 1,...,N,\ k = 1,...,2N-1. \tag{20}$$

$$t_{k,k+1} \ge \tau_{o_i^{+},o_i^{-}} \left(\sum_{j=1}^{N} z_{j,k-1}^{-} + z_{ik}^{+} + z_{i,k+1}^{-} - 2 \right), \qquad i, j = 1,...,N,\ k = 1,...,2N-1. \tag{21}$$

$$t_{k,k+1} \ge \tau_{o_i^{+},o_j^{+}} \left(z_{ik}^{+} + z_{j,k+1}^{+} - 1 \right), \qquad i, j = 1,...,N,\ k = 1,...,2N-1. \tag{22}$$

$$t_{k,k+1} \geq \tau_{o_i^-,o_j^-} \left(z_{ik}^- + z_{j,k+1}^- - 1 \right), \qquad i,j = 1,...,N,\ k = 1,...,2N-1. \qquad (23)$$

$$t_{k,k+1} \geq \tau^-_{o_i^-,o_j^+} \left(z_{ik}^- + z_{j,k+1}^+ - 1 \right), \qquad i,j = 1,...,N,\ k = 1,...,2N-1. \qquad (24)$$

$$z_{ik}^+, z_{ik}^- \in \{0,1\}, \qquad i = 1,...,N,\ k = 1,...,2N. \qquad (25)$$

$$t_{k,k+1} \geq 0, \qquad k = 1,...,2N-1. \qquad (26)$$

The objective of this model is to minimize the total time to complete all the steps. Constraints (11) indicate that for each step *k*, the crane can perform only one handling operation, either picking-up or dropping off a slab. Constraints (12) and (13) require that each slab *i* must be picked up in exactly one step and dropped off in exactly one step, respectively. Constraints (14) ensure that the step of picking up a slab must be before the step of dropping off it. Constraints (15) are crane capacity constraints requiring that there cannot be more than two picking-up steps consecutively. Constraints (16) indicate that when a slab is picked up in step *k*, the crane must drop off it in step *k*+1 or step *k*+3. If it is the first slab in a double-load operation, then it will be dropped off in step *k*+3, otherwise, it will be dropped off in step *k*+1. Constraints (17) guarantee that if the slab *i* and the slab *j* have precedence relations, then the step of picking up slab *j* must be later than the step of dropping off slab *i*. Constraints (18) and (19) require that if two tasks are performed by a double-load operation, these two tasks must perform successively with a fixed order of picking-up and dropping-off steps. Constraints (20) to (24) ensure that there is sufficient time for the crane operations from each step to the next. Constraints (25) and Constraints (26) define the range of the decision variables.

4. Computational Experiments

Computational experiments are carried out to test the effectiveness and efficiency of the two models. 20 real problem instances of different sizes are collected from the slab yard of a steel company. The actual production schedules for these instances are also obtained for comparison. Standard solver CPLEX12.51 is used to solve the models on a PC with Pentium IV 3.0GHz CPU.

The objective values of the model results are compared with those of the practical production schedules in Table 2. Column Size shows the number of tasks in instance. Column Improve is the relative improvement in objective value by the optimal solution of the models as compared to the actual production schedule.

Table 2 Comparison of objective values of actual production and optimal solution

Instance index	Size	Actual production(min)	Optimal Solution(min)	Improve
1	6	50	39	21.31%
2	6	63	47	25.66%
3	10	71	58	18.83%
4	10	84	60	28.29%
5	15	88	73	16.67%
6	15	106	78	26.34%
7	20	114	82	28.05%
8	20	119	90	24.19%
9	30	127	92	27.65%
10	30	131	92	29.72%
11	30	121	93	23.01%
12	45	132	95	28.03%
13	45	130	101	22.27%
14	45	163	108	33.66%
15	45	152	108	29.07%
16	60	188	135	28.33%
17	60	598	432	27.81%
18	60	672	460	31.57%
19	80	628	476	24.19%
20	80	691	497	28.05%
AVG				26.14%

Both formulations give optimal solutions for all the instances. As can be seen from Table 2, the optimal solution significantly improves the actual schedule used in practice for all instances, with an average relative improvement of 26.14%.

To compare the efficiency of the two formulations, we recorded their running time (seconds) for solving each instance. The results are shown in Table 3. Column F1-Time (sec) and F2-Time (sec) are the running times (seconds) for solving F1 and F2, respectively. Saving is the relative saving in solution time by F2 as compared to F1.

Table 3 Comparison of the running times of F1 and F2

Instance index	Size	F1-Time(sec)	F2-Time(sec)	Saving
1	6	1.168	0.655	43.90%
2	6	0.409	0.280	31.58%
3	10	1.491	1.217	18.38%
4	10	4.256	3.728	12.42%
5	15	5.748	2.902	49.51%
6	15	6.359	1.217	80.86%
7	20	11.293	4.883	56.76%
8	20	11.865	9.309	21.54%
9	30	63.391	12.712	79.95%
10	30	67.203	37.159	44.71%
11	30	66.014	23.317	64.68%
12	45	286.153	142.503	50.20%
13	45	304.597	241.068	20.86%
14	45	325.093	132.799	59.15%
15	45	273.233	150.774	44.82%
16	60	999.964	560.693	43.93%
17	60	1113.162	851.455	23.51%
18	60	1617.612	445.581	72.45%
19	80	1919.892	1189.031	38.07%
20	80	2664.714	1444.917	45.78%
AVG				45.15%

We can observe from the results in Table 3 that F2 constantly takes shorter time to solve the problem than F1. The average saving of running time by F2 is around 45%, showing that F2 is much efficient than F1. Although the saving varies for different instances, it does not show a general trend as the problem size increases, indicating that the relative efficiency is not affected by problem size. This can also be seen clearly from Figure 3 where the average running times of F1 and F2 and the saving for each problem size are shown.

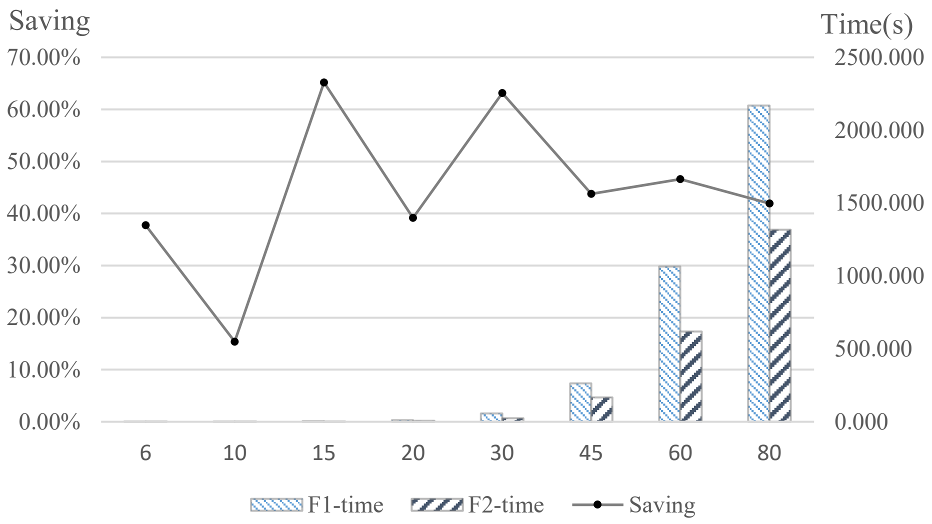

Figure 3 The saving and the average running times of each size

Because the decision variables *z* of the second model consider the crane lifting-up and dropping-off operations, according to the integrity of the task and the requirement for the crane double-load operation, constraints (14) to (19) limit values of the decision variables, which greatly reduce the search space. This may be the reason why the computational efficiency of the second model is much higher than the first one.

5. Conclusions

In this paper, we have studied a double-load crane scheduling problem in a steel company. The double-load crane can handle two slabs at the same time and so can be more productive. However, the problem of scheduling its operations becomes much more complex and is quite different from previous studies in the literature. We first formulated the problem as a mixed integer linear programming model considering the route of the crane movements. Then another formulation was proposed to determine the slab lifting-up and dropping-off operations in each crane operation step. The computational experiments were conducted to verify the efficiency and effectiveness of the models. The results show that the model solutions greatly improve the schedules used in practice with an average improvement about 26% on the test instances. The experiments also show that the models proposed in this paper can solve instances of practical size, and that the second model is more efficient than the first one, with an average savings of around 45% in computation time.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2016YFB0901900), the Fund for Innovative Research Groups of the National Natural Science Foundation of China (71621061), the Major International Joint Research Project of the National Natural Science Foundation of China (71520107004), General fund of the National Natural Science Foundation of China (71472081), and the 111 Project (B16009).

References

Aron, I.D., L. Genç-Kaya, I. Harjunkoski, S. Hoda, and J. N. Hooker. 2011. "Factory crane scheduling by dynamic programming." *Proceedings of 12th INFORMS Computing Society Conference (ICS 2011)*, 93-107.

Chen, J. H., D. H. Lee, and M. Goh. 2014. "An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem." *European Journal of Operational Research*, 232: 198-208.

Dohn, A., and J. Clausen. 2009. "Optimising the Slab Yard Planning and Crane Scheduling Problem using a two-stage heuristic." *International Journal of Production Research*, 48(15): 4585-4608.

Fleury, G., M, Gourgand, and P. Lacomme. 2001. “Metaheuristics for the Stochastic Hoist Scheduling Problem (SHSP).” *International Journal of Production Research*, 39(15): 3419-3457.

Ge, Y., and Y. Yih. 1995. “Crane scheduling with time windows in circuit board production lines.” *International Journal of Production Research*, 33(5): 1187-1199.

Guo, P., W. M. Cheng, Y. Wang, and N. Boysen. 2018. “Gantry crane scheduling in intermodal rail-road container terminals.” *International Journal of Production Research*, 56 (16): 5419-5436.

Kim, K. H., and G. P. Hong. 2006. “A heuristic rule for relocating blocks.” *Computers & Operations Research*, 33: 940-954.

Kuo, W. C., S. H. Kung, and C. K. Huang. 2011. “Order picking algorithms for a two-carousel-single-crane automated storage and retrieval system.” *Journal of Information & Optimization Sciences*, 32(3): 763–776.

Lai M., T. G. Crainic, Ma. D. Francesco, and P. Zuddas. 2013. “An heuristic search for the routing of heterogeneous trucks with single and double container loads.” *Transportation Research Part E*, 56: 108-118.

Lee, Y., and S. Chao. 2009. “A neighborhood search heuristic for pre-marshalling export containers.” *European Journal of Operational Research*, 196: 468-475.

Lee, Y., and Y. Lee. 2010. “A heuristic for retrieving containers from a yard.” *Computers & Operations Research*, 37: 1139-1147.

Li, W. K., Y. Wu, M. E. H. Petering, M. Goh, and R. D. Souza. 2009. “Discrete time model and algorithms for container yard crane scheduling.” *European Journal of Operational Research*, 198: 165-172.

Li, W. K., M. Goh, Y. Wu, M. E. H. Petering, R. D. Souza, and Y. C. Wu. 2012. “A continuous time model for multiple yard crane scheduling with last minute job arrivals.” *International Journal of Production Economics*, 136: 332-343.

Liu, J. Y., Y. W. Wan, and L. Wang. 2006. “Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures.” *Naval Research*

Logistics, 53(1): 60–74.

Mantovani, S., G. Morganti, N. Umang, T. G. Crainic, E. Frejinge, and E. Larsen. 2018. "The load planning problem for double-stack intermodal trains." *European Journal of Operational Research*, 267: 107-119.

Rei, R. J., M. Kubo, and J. P. Pedroso. 2008. "Simulation-based optimization for steel stacking, in Modelling." In An, L.T.H., Bouvry, P. and Tao, P.D. (eds), *Computation and Optimization in Information Systems and Management Sciences*, Springer, Berlin, Germany, pp.254-263.

Stahlbock, S., and S. Voß. 2008 "Operations research at container terminals: a literature update." *OR Spectrum*, 30(1): pp.1-52.

Tang, L. X., X. Xie, and J. Y. Liu. 2009. "Scheduling of a single crane in batch annealing process." *Computers & Operations Research*, 36: 2853-2865.

Upadhyay, A., W. H. Gu, and N. Bolia. 2017. "Optimal loading of double-stack container trains." *Transportation Research Part E*, 107: 1-22.

Wan, Y. W., J. Y. Liu, and P. C. Tsai. 2009. "The assignment of storage locations to containers for a container stack." *Naval Research Logistics*, 56:699-713.

Xie, X., Y. Y. Zheng, and Y. P. Li. 2015. "Genetic Algorithm and Its Performance Analysis for Scheduling a Single Crane." *Discrete Dynamics in Nature and Society*, Vol. 2015, Article ID 618436, 12 pages.

Zapfel, G., and M. Wasner. 2006. "Warehouse sequencing in the steel supply chain as a generalized job shop model." *International Journal of Production Economics*, 104: 482-501.

Zhang, R., Z. H. Jin, Y. Ma, and W. X. Luan. 2015. "Optimization for two-stage double-cycle operations in container terminals." *Computers & Industrial Engineering*, 83: 316-326.

Zheng, F. F., X. Y. Man, F. Chu, M. Liu, and C. B. Chu. 2018. "A two-stage stochastic programming for single yard crane scheduling with uncertain release times of retrieval tasks." *International Journal of Production Research*, published online:

https://doi.org/10.1080/00207543.2018.1516903.

Zhou, Z. L., and J. Y. Liu. 2008. “A heuristic algorithm for the two-hoist cyclic scheduling problem with overlapping hoist coverage ranges.” *IIE Transactions*, 40(8): 782-794.