

An Efficient Heuristic Algorithm for Overhead Cranes Scheduling Operations in workshop

Peng Ge, Jun Wang, Maozhu Jin*, Peiyu Ren, Huafeng Gao

Business School of Sichuan University, Chengdu, 610064, China

Received: Jul 8, 2011; Revised Oct. 4, 2011; Accepted Oct. 6, 2011

Abstract: This paper discusses the scheduling problem for two factory cranes that move along a single track, which is one of the most important equipment in manufacturing shop. Each crane is assigned a sequence of pickups and deliveries at specified locations and the two cranes must be operated so as to avoid interference with each other. In order to minimize the makespan, we define the bottleneck crane, which takes the maximum time to complete a given set of tasks. And a mixed-integer programming model which considers various constraints related to the operations of cranes is formulated. However, this problem is NP-hard, therefore we propose a heuristic algorithm to solve this problem. Finally, a numerical experiment on a specific manufacturing plant is used to illustrate the efficiency from computational point of view.

Keywords: Overhead crane, Scheduling, Heuristic algorithm.

1. Introduction

In the productive process, it is frequently that production materials should be transported from one operation location to another. Therefore, overhead cranes set up with a hoist traveling along the bridge between parallel runways are designed to meet industrial lifting requirements for the medium or heavy items (production materials or equipments). Actually, overhead cranes play an important role in manufacturing facilities.

In the workshops of Baosteel Metals Company, there was only one crane in the past. However, with the developments of the company and its production process, only one crane can hardly meet requirements of production, consequently, workshop supervisors propose to add one crane to each workshop and make two cranes move along the same horizontal runways. Nevertheless, with the additional crane, the work efficiency has not been improved significantly. In investigations, we find that, in an actual production process of manufacturing workshop, cranes always be asked by several tasks simultaneously, so workshop supervisors must make a crane scheduling to dispatch cranes. After the scheduling for cranes in a workshop is drawn up, cranes should be operated according to the scheduling. Generally, workshop crane schedules are planned by workshop supervisors based on their experi-

ences. However, these schedules always result in weakness that cranes are interfered with each other and make either of them in a state of waiting for a long time. Unquestionably, the carrying capacities of the cranes cannot be taken full advantage of and the working time can hardly be cut down obviously. Thus, we have to systemically analyze the issue and propose an effective and reliable method to make full use of the cranes.

By reviewing the researches relevant to scheduling, it is found that the scholars and researchers have obtained lots of achievement on some aspects of scheduling modeling, scheduling algorithm, etc. For example, in the recent academic papers, solving the problem of on-line scheduling with non-crossing constraints, Zhang et al. [1] provided an optimal competitive ratio heuristics to minimize the latest completion time. Tang et al. [2] considered both machine and crane positions and proposed two-phase algorithm for coping with the problem based on a single crane scheduling problem motivated by batch annealing process in the iron and steel industry. According to two hoists sharing a common track used to move products between the tanks in the production line which was divided into two non-overlapping zones with a hoist assigned to each zone, Zhou and Li [3] developed a mixed integer linear programming model for scheduling hoist moves. Guan et al. [4] developed a heuristic for the problem and perform worst-case

* Corresponding author: e-mail: allen_wangjun@126.com

analysis in solving a scheduling problem in which the processors arranged along a straight line to minimize the total weighted completion time of the jobs, etc. Based on such researches above, several studies have been conducted to improve the efficiency of crane scheduling. Lieberman and Turksen [5, 6] studied the cranes scheduling problems and two-operation cranes scheduling problems for copper smelters and steel mills. Hirofumi et al. [7] proposed an algorithm of a near optimal solution for a cyclic crane scheduling in a computer-integrated manufacturing environment. Hirofumi et al. [8] developed a knowledge-based intelligent crane scheduling for controlling a stacker crane in a computer-integrated manufacturing environment. Aron et al. [9] presented a specialized dynamic programming algorithm to solve that problem, which just need to consider certain types of trajectories. In order to control the state space size, they used an innovative state space representation based on a cartesian product of intervals of states and an array of two-dimensional circular queues. Zheng et al [10] established a model for simulating crane scheduling in workshop based on cellular automata in the paper. Zhou and Li [11] developed a mixed integer linear programming model to find optimal solutions for scheduling the crane, and then the model was extended to problems in the systems with multi-station.

Quay cranes scheduling problems were also discussed in several papers. However, compared with factory cranes, the scheduling problem differs in various aspects. The quay cranes load (or unload) containers into ships rather than transferring items from one location on the track to another. A given quay crane can reach several ships, or several cranes holding in a single ship, either by rotating its arm or perhaps by moving laterally along the track. Kim and Park [12] proposed a branch and bound (B&B) method to obtain the optimal solution of the quay crane scheduling problem, and they also proposed a heuristic search algorithm, called greedy randomized adaptive search procedure to overcome the computational difficulty of the B&B method. Based on Kim and Parks study, Chung and Choy [13] proposed a modified genetic algorithm to deal with that problem. Tavakkoli-Moghaddam et al. [14] proposed a genetic algorithm to solve the quay crane scheduling and assignment problem, namely QCSAP, in a container port (terminal) of the real-world situations, which was able to solve the QCSAP, especially for large sizes compared against the LINGO software package. Jiang Hang Chen et al. [15] considered the unique features of the quay crane scheduling problem at indented berth and put forward a decomposition heuristic framework developed and enhanced by Tabu search. Han et al. [16] proposed a mixed integer programming model and a simulation based Genetic Algorithm search procedure applied to generate robust berth and QC (allowed to move to other berths before finishing processing on currently assigned vessels, adding more flexibility to the terminal system)schedule proactively, and through computational experiment, the satisfied performance of the developed algorithm under uncertainty was achieved. Christian Bierwirth et al. [17] reviewed the literature re-



Figure 1 A drawing of two overhead cranes in a workshop.

lated to quay crane scheduling and developed quay crane scheduling problems so as to provide a support in modeling problem characteristics and in suggesting applicable algorithms which could be conducive to increasing importance for the terminal management. Lu et al. [18] pointed out the concept of contiguous bay operations and developed a heuristic to generate quay crane schedules. That heuristic was efficient and effective with polynomial computational complexity. He et al. [19] developed a dynamic scheduling model using objective programming for yard cranes based on rolling-horizon approach.

In this paper we analyze the problem of scheduling two factory cranes moving along the same horizontal runways and propose a specific algorithm. The aim of this study is to determine the sequence of pickups and delivers so that the makespan of cranes is minimized. The structure of this paper is organized as follows: in Section 2, we describe this problem briefly and also propose the corresponding mathematical formulation. In Section 3, we introduce a heuristic algorithm, followed by computational experiments in Section 4. Finally, conclusions and future research directions are discussed in Section 5.

2. Problem description and mathematical formulation

2.1. Problem description

The aim of studying the overhead cranes scheduling problem is to determine the sequence of discharging and loading operations that overhead cranes will perform so that the completion time of all the tasks is minimized. Figure.1 provides a drawing of two overhead cranes working in a workshop.

In practice, a crane scheduling problem typically consists of a number of tasks, each of which specifies several jobs to be performed consecutively. For example, a job may require a crane move to the location overhead the equipment, which needs to be conveyed to another location, then the crane makes a vertical movement and pick

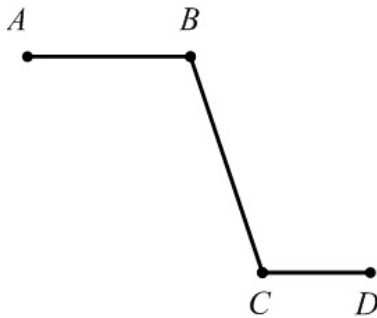


Figure 2 The route of performing a task by an overhead crane.

up the equipment high enough, next, the crane delivers the equipment to a certain location, and then makes the corresponding vertical movement and discharge the equipment. A task can be represented as Figure.2, where Line AB, CD mean the clamping and unloading respectively while, Line BC means the process of transportation.

To simplify the working process of each crane, in this study, we explicitly make the following assumptions:

1. Only two cranes are considered.
2. The two cranes on the corresponding extreme points of the track at the beginning.
3. Cranes cannot cross over each other.
4. Account only for the longitudinal movements of the crane along the track and assume that the crane has time to make the necessary vertical movements as it moves from one task location to another. This result in little loss of generality, because any additionally time necessary for vertical motion can be built into ascent and unload time for the task.
5. The starting location, final location, the duration of ascent and unload of each task are given, and all the tasks are released at the same time.
6. Each task assigned to a given crane must be finished before the next task assigned to that crane begins.
7. For safety, each task must be executed continuously and cannot be disrupted before it is completed. E.g. a loaded crane cannot move in an opposite direction of the destination or beyond the destination, but an unloaded crane can move freely.
8. The travelling velocity of each crane is constant, namely, cranes is either stationary or walk in a constant speed.

2.2. Mathematical formulation

The following notations are used to describe the problem studied throughout the paper:

\emptyset : The empty set.

M : A sufficiently large constant.

I : The set of all tasks.

i, j : Index of task, $i, j \in I$.

I_k : Tasks assigned to crane k .

$|I_k|$: The cardinality of set I_k . Namely, the number of the tasks completed by crane k .

S : The length of the crane track.

$h_k(t)$: The location of crane K at time t . At the beginning, $h_1(0) = 0$ and $h_2(0) = s$, then we call crane 1 is the left crane and crane 2 is the right crane.

D_{I_k} : The delay time of the crane k throughout the process of completing all tasks.

v : The velocity of the cranes, which is a constant.

x_{ij}^k : Equal to 1, if crane k performs task j immediately after performs task i ; equal to 0, otherwise. Specially, x_{0j}^k Equal to 1 means the first task conducted by crane k is i .

$tf(i)$: The actual completion time of task i .

$tl(i)$: The loading time of task i .

$tt(i)$: The transport time from the initial location of task i to the end location.

$td(i)$: The discharge time of task i .

$p(i)$: Processing time required by task i , which includes the loading time, transport time, and discharge time. Namely, $p(i) = tl(i) + tt(i) + td(i)$.

d_{ij} : The distance between terminal location of task i and the starting location of task j . Specially, d_{0j} represents the travel distance from the initial position of crane to the starting location of task j .

Y_k : he completion time of crane k , where $k = 1, 2$.

As previously mentioned, our aim is to determine how to assign tasks to the two cranes so as to minimize the makespan. Therefore, we define the bottleneck crane and put especial emphasis upon the bottleneck crane.

Definition 1. Crane k is a bottleneck crane, if $Y_k = \max_{l=1,2} \{Y_l\}$.

Because all the tasks finished means all cranes finish the tasks assigned to them. Then, the objective function is shown in Eq.(1)

$$\min \max \{Y_k, k = 1, 2\} \tag{1}$$

s.t

$$Y_k = \sum_{i \in I_k} p_i + \frac{1}{v} \left(\sum_{i,j \in I_k} x_{ij}^k d_{ij} + D_{I_k} \right), k = 1, 2, \tag{2}$$

$$\sum_{i,j \in I_1} x_{ij}^1 + x_{0i}^1 = |I_1|, i \in I_1, \tag{3}$$

$$\sum_{i,j \in I_2} x_{ij}^2 + x_{0i}^2 = |I_2|, i \in I_2, \tag{4}$$

$$|I_1| + |I_2| = |I|, \tag{5}$$

$$\sum_{i \in I} x_{ij}^1 + \sum_{i \in I} x_{ij}^2 = 1, \forall j \in I, \tag{6}$$

$$h_2(t) - h_1(t) > 0, \forall t \in [0, \max\{Y_1, Y_2\}], \tag{7}$$

$$0 \leq h_k(t) \leq S, k = 1, 2, \tag{8}$$

$$tf(i) + \frac{1}{v}d_{ij} + p(j) - tf(j) \leq M(1 - x_{ij}^k), \quad (9)$$

$$\forall i, j \in I_k, k = 1, 2,$$

$$x_{ij}^k \in \{0, 1\}, i, j \in I_k, K = 1, 2. \quad (10)$$

In the objective Eq.(2), our goal is to minimize the makespan which is depending on the completion time of each crane. Then we want to make the bottleneck crane complete its tasks as earlier as possible. Eq.(2) represents the completion time of crane k , where $k \in \{1, 2\}$, which consists of the processing time of these tasks be assigned to the crane k , the travelling time between the terminal location of the predecessor task and the starting location of the next task, and the delay time of the crane throughout the process of completing all tasks. Eqs.(3-4) represent the number of task assigned to crane 1 and crane 2. Eq.(5) indicates that all the tasks have to be allocated. Eq.(6) ensures that each task will be completed by one crane only. Eq.(7) ensures to avoid the interference between these two cranes. Eq.(8) requires that the cranes stay on the track, Eq.(9) ensures that processing continues for the required amount of time once it starts.

It is obvious that the formulation of Eqs.(1-10) is a mixed-integer linear program. Computational experiments showed that the computational time is excessive for practical use. Then, we will propose a heuristic approach to solve this crane scheduling problem.

3. Crane scheduling procedure

3.1. Supplementary Definitions of Variables

s_{i0} : The initial location of task i .

s_{id} : The end location of task i .

FI : The set of finished tasks.

UI : The set of unfinished tasks.

$Id_k(t)$: Equal to 1, if crane k is idle at time t ; equal to 0, otherwise.

$Cr(t)$: Equal to 1, if the cranes k interfered with each other at time t ; equal to 0, otherwise.

3.2. Crane scheduling procedure

In this section, we will propose a heuristic approach to obtain an optimal allocation plan to determine the sequence of pickups and delivers that the two cranes will perform. The process of obtaining the optimal allocation plan includes two phases: obtaining an initial feasible solution and the solution improvement. In the phase of obtaining an initial feasible solution, all the tasks have to be assigned to each crane available. Consider that the two cranes on the extreme point of the track, then all tasks are assigned to comply with the principle that for each crane, assigning the nearest task to it, meanwhile, we have to ensure that the two cranes cannot be interfered with each other, although one crane may need to yield to another. Therefore,

we should make every effort to assign the task close to the leftmost position to the left crane, and assign the task close to the rightmost position to the right crane.

Phase 1: Obtain an initial feasible solution

Step 1: Set $t = 0$, $I_k = \emptyset$, where $k = 1, 2$ and $Id_1(0) = Id_2(0) = 1$.

Step 2: Distribute first task to each crane.

(1) Check UI , if $UI = \emptyset$, turn to Step 7;

(2) else, compare the tasks which belong to UI .

① if $s_{i0} < s_{j0}$, where $s_{i0} = \min_{i \in UI} \{S - s_{i0}\}$, $s_{j0} =$

$\min_{i \in UI} \{s_{i0}\}$, task i_1 is assigned to crane 2, $I_2 =$

$\{i_1\}$. Simultaneously, delete i_1 from UI , $UI = UI -$

$\{i_1\}$ and set $Id_2 = 0$. Next, select j_1 as the first

task of crane 1, where j_1 satisfies that $Cr(t) =$

0 , $t \in [0, \max\{tf(i_1), tf(j_1)\}]$, then, put task j_1

in set I_1 , $I_1 = \{j_1\}$, and at the same time, delete

j_1 from UI , $UI = UI - \{j_1\}$ and set $Id_1 = 0$.

② Otherwise, vice versa.

Step 3: Without loss of generality, let $I_1 = \{j_1, j_1, \dots, j_n\}$ and $I_2 = \{i_1, i_1, \dots, i_n\}$ at time t .

(1) Check UI , if $UI \neq \emptyset$, turn to Step 7;

(2) Else, construct the following sets:

$$L(t) = \{i | s_{i0} \leq h_1(t), i \in UI\}$$

$$R(t) = \{i | s_{i0} \geq h_1(t), i \in UI\}$$

$$M(t) = \{i | h_1(t) \leq s_{i0} \leq h_2(t), i \in UI\}$$

Step 4: Check the state of each crane:

(1) if $\sum_{k=1}^K Id_k = 0$, turn to Step 7;

(2) Else, turn to Step 5.

Step 5: (1) If $Id_1(t) = Id_2(t) = 1$

① if $h_2(t) - s_{r0} < s_{w0} - h_1(t)$, where $s_{r0} =$

$\max_{i \in UI} \{s_{i0}\}$, $s_{w0} = \min_{i \in UI} \{s_{i0}\}$, assign task r to I_2 ,

and let $Id_2(t) = 0$, $UI = UI - \{r\}$;

② else, assign task w to I_1 , let $Id_1(t) = 0$ and

$UI = UI - \{w\}$

(2) Else if $Id_1(t) = 1$ and $Id_2(t) = 0$, assign task

w to I_1 , where w satisfies that $\sum_t^{t+p_w} Cr(t) = 0$,

and $s_{w0} = \min_{i \in L(t)} \{s_{i0}\}$, at the same time, let $UI =$

$UI - \{w\}$ and $Id_1 = 0$;

(3) Else, $Id_2(t) = 1$ and $Id_1(t) = 0$, assign task r to

I_2 , where task r satisfies that $\sum_t^{t+p_r} Cr(t) = 0$, and

$s_{r0} = \max_{i \in R(t)} \{s_{i0}\}$, then let $UI = UI - \{i_{m+1}\}$

and $Id_2 = 0$;

Step 6: Let $t = t + 1$;

Step 7: Terminate;

Figure. 3 shows the flow chart of obtaining an initial feasible solution as follows:

Phase 2: Solution improvement

In the solution improvement phase, we define the key task, which is in the bottleneck crane and with the longest

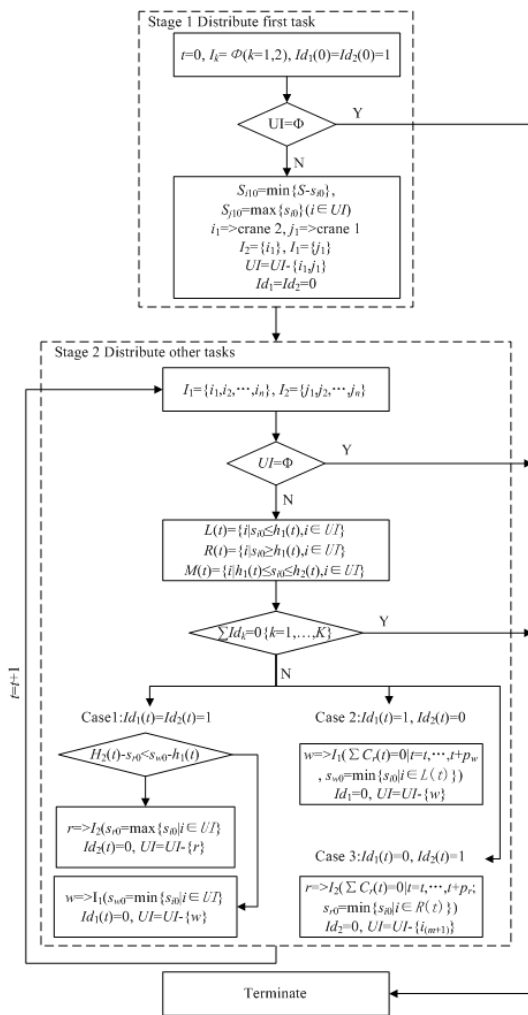


Figure 3 The flow chart of obtaining an initial feasible solution.

idle time. Obviously, the bottleneck crane and key task are changed with the adjustment of task sequence.

Definition 2. Task l is a key task, if $idl_k(l) + x_{ij}^k \frac{d_{ij}}{v} = \max_{i \in I_k} \{idl_k(i) + x_{ij}^k \frac{d_{ij}}{v}\}, \forall j \in I_k$, where crane k is the bottleneck crane.

In order to improve the solution, we pay special attention to the bottleneck crane for the completion time of bottleneck crane determines the completion time of all the tasks. Hence, we expect to minimize the time difference between these two cranes, meanwhile, shorten the completion time of bottleneck crane. Therefore, we adjust the task sequence by moving or inserting key task to the non-bottleneck crane, or exchanging it with other tasks. Then, we define insert operator, exchange operator, and translation operator to realize the solution improvement.

Definition 3. Let task i be the key task. For convenience, suppose task i belongs to crane 1. If task i can be moved

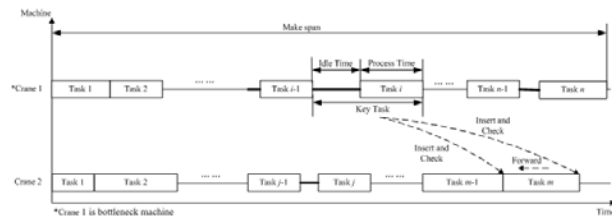


Figure 4 The illustration of insert operator.

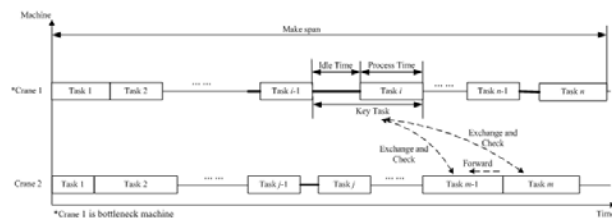


Figure 5 The illustration of exchange operator.

to certain location that behind task j in crane 2, such that task i cannot be interfered with the other tasks, and the completion time of bottleneck crane is shortened, then we let $In_{ij} = 1$, otherwise, $In_{ij} = 0$, and In_{ij} is called insert operator, denoted as:

$$In_{ij} = \begin{cases} 1, & \text{task } i \text{ can be inserted behind task } j, j \in I_2; \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

If other tasks in the task sequence of crane 2 will not be interfered by the key task, it should be inserted in the last position. Otherwise, the key task should be move forward from the last position. Because, compared with other tasks assigned to crane 2, the distance between the key task and crane 2 is farther than others. See Figure.4

Definition 4. Let task i be the key task. For convenience, suppose task i belongs to crane 1. If task i can be exchange with task j in crane 2, such that task i and j cannot be interfered with the other tasks, and the completion time of bottleneck crane is shortened, then we let $Ex_{ij} = 1$, otherwise, $Ex_{ij} = 0$, and Ex_{ij} is called exchange operator, denoted as

$$Ex_{ij} = \begin{cases} 1, & \text{task } i \text{ can be exchanged with task } j, j \in I_2; \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

The following Figure 5 explains the exchange operator detailedly.

Definition 5. Let task i be the key task. For convenience, suppose task i belongs to crane 1. If task i can be exchange with task j in crane 1, such that task i and j cannot be interfered with the other tasks, and the completion time of bottleneck crane is shortened, then we let $Tr_{ij} = 1$,

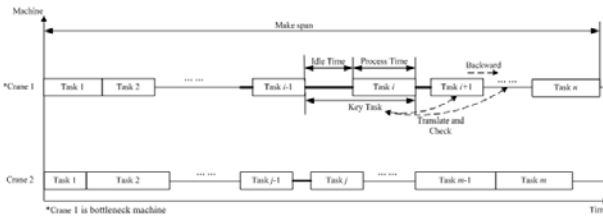


Figure 6 The illustration of translate operator.

otherwise, $Tr_{ij} = 0$, and Tr_{ij} is called translate operator; denoted as

$$Tr_{ij} = \begin{cases} 1, & \text{task } i \text{ can be exchanged with task } j, j \in I_1; \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

The following Figure 6 explains the exchange operator detailedly.

Furthermore, solution improvement phases should abide by the following rulers:

Rule 1: Insert operator has precedence to exchange operator; exchange operator has precedence to translate operator;

Rule 2: Insert operator and exchange operator start from the last task of non-bottleneck crane with descending order; translate operator start from next task with ascending order.

Insert operator makes a greater contribution to shorten the completion time of the bottleneck crane. If the key task has been deleted from the bottleneck crane, the working hour will be reduced dramatically, therefore, we give preference to Insert operation. Translation operator deals with the tasks which belong to the same task sequence, but the exchange operation deals with the tasks belong to the bottleneck crane and the non-bottleneck, then, the exchange operation may be superior to translation operator for the task sequence of each crane are optimized by the index of the distance between the task and the corresponding crane.

- Step 1: Determine the bottleneck crane;
- Step 2: Rank the task sequence belong to each crane in chronological order;
- Step 3: Determine the key task;
- Step 4: Optimize the solution by iterating insert operator, exchange operator, and translation operator;
- Step 5: Terminate.

Figure 7 shows the flow chart of solution improvement as above:

4. A numerical example

In order to test the efficiency of the proposed heuristic algorithm, We report computational tests on a representative problem that is based on an actual industry scheduling situation in Baosteel Metals Company. We randomly

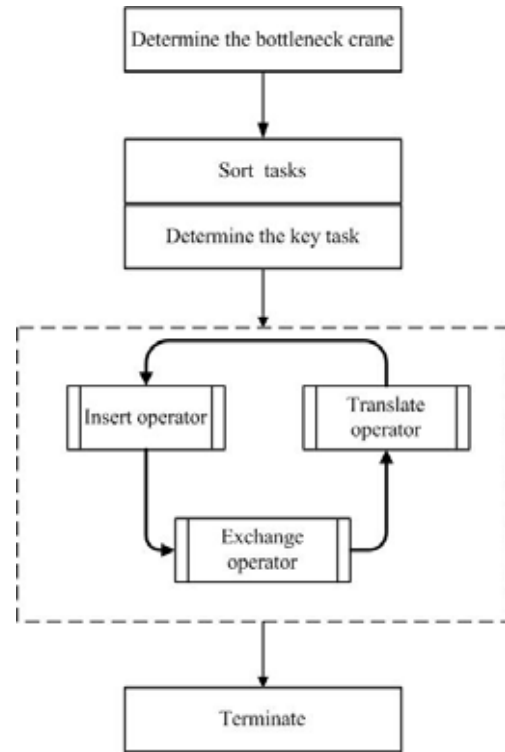


Figure 7 The flow chart of solution improvement.

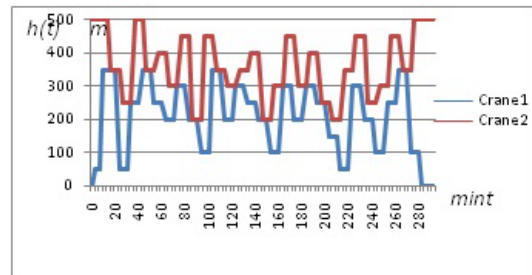


Figure 8 Solution of the 27 tasks

choice a historical work record which included 27 tasks, and consumed 435 minutes. However, based on the proposed heuristic algorithm, we can see that the results for 27 tasks appear in Figure. 8. The vertical axis represents distance along the track with 500 meters while the horizontal axis represents time. Thus the schedule for the 27-tasks problem spans about 280 minutes. The space-time trajectory of the crane 1 appears as a blue line, and as a red line for the crane 2. The crane 1 begins and ends at the leftmost position, and analogously for the crane 2. In the 27-task instance, the completion time of two-lane are almost the same, which may indicate a good allocation of tasks to cranes.

5. Conclusion

This paper described the application of scheduling techniques to an industrial process and aimed to determine the sequence of pickups and delivers that a crane will conduct so as to minimize the completion time of makespan. Therefore, we constructed a mathematical formulation to describe the character of that assignment problem, that mathematical model was a mixed-integer linear program but lack of practical applicability because of excessive computational time. Then we presented a two-step heuristic approach to obtain the optimal solution for two factory cranes, and the corresponding algorithm was developed and described, meanwhile, the concepts of bottleneck crane and key task were proposed. This study also performed a numerical experiment to illustrate the feasibility and effectiveness of the approach.

Acknowledgement

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

This work was supported by Major International Joint Research Program of the National Natural Science Foundation of China (71020107027), National High Technology Research and Development Major Program of China (Program 863) (2008AA04A107), Project 985 and 211 of Sichuan University, and Doctoral Fund of Ministry of Education of China (20110181110034).

References

- [1] L.I. Zhang, K. Khammuang, A. Wirth. On-line scheduling with non-crossing constraints, *Operations Research Letters*, 36, 579-583 (2008).
- [2] L.X. Tang, X. Xie, J.Y. Liu. Scheduling of a single crane in batch annealing process, *Computers & Operations Research*, 36, 2853-2865 (2009).
- [3] Z.I Zhou, L. Li. A solution for cyclic scheduling of multi-hoists without overlapping, *Annals of Operations Research*, 168, 5-21 (2009).
- [4] Y.P. Guan, W.Q. Xiao, R. K. Cheung, C.L. Li. A multiprocessor taskscheduling model for berth allocation: heuristic and worst-case analysis, *Operations Research Letters* 30, 343-350 (2002).
- [5] R. W. Lieberman, I. B. Turksen. Crane scheduling problems. *AIIE Transactions*, 133, 04-11(1981).
- [6] R. W. Lieberman, I. B. Turksen. Two-operation crane scheduling problems. *AIIE Transactions*, 14, 147-55 (1982).
- [7] M. Hirofumi, S. S. Jen, S. S. A. Robert. Crane scheduling problem in a computer integrated manufacturing environment. *Management Science*, 37, 587-606 (1991).
- [8] M. Hirofumi, S. S. Jen, S. S. A. Knowledge-based system for stacker crane control in a manufacturing environment. *IEEE Transactions on Systems, Man, and Cybernetics*, 199, 32-45 (1991).
- [9] A. Ionut, G. K. Latife, Harjunkoski, Iiro, Hoda, Samid, Hooker, N. John. Optimal Movement of Factory Cranes. *Tepper School of Business*.143(2008).
- [10] Z. Zheng, L. Xu, X. Q. GAO. Simulation model of crane scheduling in workshop based on cellular automata. *Systems Engineering-Theory & Practice*, 2, 137-142 (2008).
- [11] Z. Zhou, L. Li. Optimal cyclic single crane scheduling for two parallel train oilcan repairing lines. *Computers and Operations Research*, doi:10.1016/j.cor.2011.06.002 (2011).
- [12] K. H. Kim, Y. M. Park. A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156, 752-768 (2004).
- [13] S. H. Chung, K. L. Choy. A modified genetic algorithm for quay crane scheduling operations. *Expert Systems with Applications*, 39, 4213- 4221 (2012).
- [14] R. Tavakkoli-Moghaddama, A. Makui, S. Salahi, M. Bazzazi, F. Taheri. An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering*, 56, 241-248 (2009).
- [15] J. H. Chen, Der-Horng Lee, J. X. Cao. Heuristics for quay crane scheduling at indented berth. *Transportation Research Part E*, 47, 1005-1020 (2011).
- [16] X. I. Han, Z. Q. Lu, L.F. Xi. Heuristics for quay crane scheduling at indented berth. *European Journal of Operational Research*, 207, 1327-1340 (2010).
- [17] C. Bierwirth, F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202, 615-627 (2010).
- [18] Z. Q. Lu, X. L. Han, L. F. Xi, A. L. Erera. A Heuristic for the Quay Crane Scheduling Problem Based on Contiguous Bay Crane Operations. *Computers and Operations Research*, doi:10.1016/j.cor (2012).
- [19] J. L. He, D. F. Chang, W. J. Mi, W. Yan. Transportation A hybrid parallel genetic algorithm for yard crane scheduling. *Research Part E*, 46, 136-155 (2010).