Multihoist Cyclic Scheduling With Fixed Processing and Transfer Times

Yun Jiang and Jiyin Liu

***Abstract*—In this paper, we study the no-wait multihoist cyclic scheduling problem, in which the processing times in the tanks and the transfer times between tanks are constant parameters, and develop a polynomial optimal solution to minimize the production cycle length. We first analyze the problem with a fixed cycle length and identify a group of hoist assignment constraints based on the positions of and the relationships among the part moves in the cycle. We show that the feasibility of the hoist scheduling problem with fixed cycle length is consistent with the feasibility of this group of constraints which can be solved efficiently. We then identify all of the special values of the cycle length at which the feasibility property of the problem may change. Finally, the whole problem is solved optimally by considering the fixed-cycle-length problems at these special values.**

***Note to Practitioners*—Automated electroplating lines are commonly used in the production of many products, such as printed-circuit boards. The productivity of these systems depends heavily on effective scheduling of the material handling hoists that move the products between the processing tanks. This paper studies the hoist scheduling problem in systems where the processing times in the tanks and the intertank move times are fixed parameters. An efficient optimal algorithm is developed to solve the problem with any number of hoists. The resulting schedule can be programmed as programmable logic controller codes to directly control the hoist operations. The algorithm can also be used to develop heuristic solutions for multihoist scheduling in systems where the processing times may vary in given intervals.**

***Index Terms*—Hoist scheduling, multiple hoists, no-wait, optimization.**

I. Introduction

ELECTROPLATING is a common process in the production of many products, such as printed-circuit boards (PCBs). An electroplating system is an automated production line with a series of tanks that contain the required chemical solutions. Tanks are also called processing stations. Fig. 1 shows a sample electroplating line. The parts to be processed must visit a given sequence of tanks according to the technological requirements. The required processing time in a tank may be a fixed parameter or a flexible parameter within given limits (the limits define a feasible window for the processing time). Hoists mounted on a common track are used to move the parts between the tanks. Effective operations of the hoists are vital to achieve high productivity of the line.

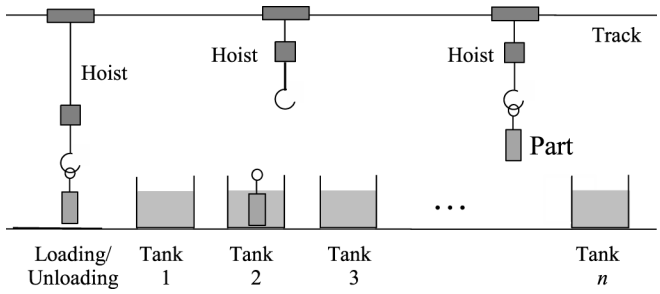

Fig. 1. Sample electroplating line.

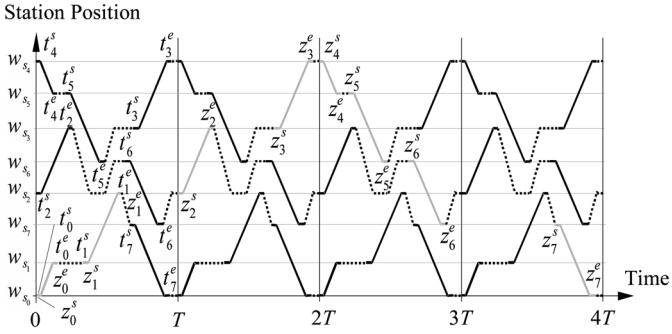

Fig. 2. Sample of a time-way diagram.

An electroplating line normally produces identical parts cyclically. In each cycle, one part enters the line and another leaves the line after completing all of the required processing steps. Without loss of generality, we take the time point at which a part starts moving to the first tank as the starting point of a cycle. The duration of a cycle is called the cycle length. Since there can be a number of parts being processed at the same time in different tanks of the system, each part can stay in the system for several cycles. Each hoist repeats a sequence of part moves in every cycle. The cyclic production can be illustrated by a "time-way diagram" such as the one in Fig. 2. A time-way diagram is a special type of Gantt Chart emphasizing the hoist movements. The horizontal axis represents time and the vertical axis represents positions along the electroplating line. We use horizontal thin gray lines to show the station positions and use solid lines to indicate the part moves between stations, where a horizontal segment indicates either lifting up or dropping off of a part at a station and an inclined segment indicates the travel between two stations. The processing time of a part at a station is the duration between the ending point of the move to this station and the starting point of the move away from the same station. Dotted inclined lines are used to show empty hoist moves while a dotted horizontal line indicates hoist

Manuscript received September 30, 2005; revised January 16, 2006. This paper was recommended for publication by Associate Editor K. Saitou and Editor N. Viswanadham upon evaluation of the reviewers' comments.

Y. Jiang is with Bilkent University, Ankara 06800, Turkey (e-mail: jiangyun@bilkent.edu.tr).

J. Liu is with Loughborough University, Leicestershire LE11 3TU, U.K. (e-mail: J.Y.Liu@lboro.ac.uk).

Digital Object Identifier 10.1109/TASE.2006.884057

waiting at a position. All of the loaded and empty moves of a hoist form the route of that hoist. Since the hoists cannot run across each other on the track, the routes of any two hoists will never have intersection in the time-way diagram. The symbols on the diagram in Fig. 2 will be explained and used later.

The hoist scheduling problem is to allocate the hoists to perform all of the required part moves in a cycle to maximize the production throughput (i.e., minimize the cycle length).

Most previous research on hoist scheduling focuses on the problem that considers only one hoist and that requires the processing time in each tank to take a value within a given window. Reference [1] proved that such a single-hoist problem was NP complete. To find optimal cyclic hoist schedules, researchers tried integer programming models (e.g., [2], [3], and [4]) and branch-and-bound algorithms (e.g., [5]–[7] and [8]). For the special case where the hoist route (the sequence of the moves) is given, [9] developed a pseudopolynomial algorithm. Reference [10] presented a polynomial search algorithm for the no-wait single-robot scheduling problem in manufacturing cells, which is equivalent to the no-wait single-hoist scheduling problem. The no-wait condition indicates that the part processing times in the tanks and the transfer times between tanks are all fixed and that, upon completion of the processing in a tank, a part needs to be moved immediately to the next tank.

The multihoist problem is more complicated and has received less research. It involves new decisions on hoist assignment and has to avoid hoist crossings and collisions as more than one hoist runs on the same track. Reference [11] provided a review of research on a broader problem class of cyclically scheduling transporters, including hoists, robots, and other vehicles in flowshops. Reference [12] proposed a mathematical model for general periodical scheduling problems and presented some relevant applications. It can be seen from these references that studies on multitransporter problems are limited and most of them do not consider the common-track constraint. Reference [13] considered two hoists in a special system where parts visit the tanks one by one in one direction from the loading station to the unloading station. They partitioned the line into two nonoverlapping zones and then assigned one hoist to perform the intertank part moves in each zone. The optimal schedule was then determined by solving the two single-hoist problems alternately. Reference [14] studied a multidegree cyclic two-robot scheduling problem in a no-wait flowshop without the common-track constraint. A different problem but related to multihoist cyclic scheduling is to determine the minimum number of hoists for a system. References [15] and [16] proposed heuristics for minimizing the number of hoists in electroplating lines. References [17] and [18] studied the problem of minimizing the number of transporters in other systems without the common-track constraint.

For the no-wait two-hoist cyclic scheduling problem with fixed processing and transfer times, [19] developed a polynomial solution algorithm that allows any part flow pattern. They identified and searched over the threshold values of the cycle length. The optimal cycle length was then obtained by checking the feasibility of each threshold cycle length. An optimal schedule was constructed by considering many different combinations in assigning hoists to a pair of moves. The method is quite efficient for the problem with two hoists. However, such analysis is difficult to be extended to multiple hoists because the hoist-assignment combinations will be too complicated to handle efficiently. To solve the no-wait multihoist problem, new approaches are needed.

In this paper, we study the no-wait multihoist cyclic scheduling problem with any number of hoists and an arbitrary part flow pattern, and develop a polynomial optimal solution algorithm. We first describe this problem formally, analyze the basic relationships between the cycle length and the timing of the moves in the cycle, and present a lowerbound and an upperbound of the cycle length (Section II). For any fixed cycle length, we identify a group of hoist assignment constraints in Section III. In Section IV, we prove that the feasibility of this constraint group is consistent with the feasibility of the corresponding cycle length. Based on this, the hoist scheduling problem for any given cycle length can be solved using an efficient algorithm. In Section V, a set of special cycle lengths, that may cause feasibility change, is identified. Section VI summarizes all of these developments and presents a complete algorithm that searches for the optimal cycle length and constructs an optimal schedule. An example is also given. Section VII concludes the paper.

II. Problem Definition and Analysis

In a similar way as defining the no-wait two-hoist problem in [19], we formally describe the no-wait multihoist scheduling problem as follows. Consider an electroplating system with stations arranged in a line from left to right: the loading station (station 0), n processing stations (stations $1, 2, \ldots, n$), and possibly an unloading station (station $n+1$). The position of station i is w_i, $i = 0, 1, \ldots, n+1$. Some systems do not have a separate unloading station. In these systems, loading and unloading are performed at the same station (station 0) and, therefore, $w_{n+1} \equiv w_0$. Each station can process, at the most, one part at a time. There are m hoists on the same track over the line for moving parts between stations. The hoists cannot travel across each other. We denote the hoists from left to right as $H_1, H_2, \ldots, H_m$. The leftmost position that H_1 can reach is w_l and the rightmost position that H_m can reach is w_r. The range $[w_l, w_r]$ covers the positions of all stations. To avoid collision, any two adjacent hoists must maintain at least a safety distance d between them.

The sequence of stations to be visited by each part, also called part flow pattern, is given as $s = [s_0, s_1, s_2, \ldots, s_n, s_{n+1}]$, where $s_0 = 0$ is the loading station, s_{n+1} is the unloading station, and s_i, $i = 1, 2, \ldots, n$ is the station for the ith processing stage of the part. The required processing time at processing station s_i is a given parameter τ_i, $i = 1, 2, \ldots, n$. After processing at station s_i, the part must be immediately moved to station s_{i+1} by a hoist. We denote this move as m_i. The move consists of lifting up the part from station s_i, carrying the part from station s_i to station s_{i+1}, and dropping off the part into station s_{i+1}. The time needed for the hoist to lift up or drop off a part at a station is μ (the developments of this paper can be easily extended to situations where liftup time and dropoff time are different). The traveling speed of a hoist carrying a part is ν, and the maximum speed of an empty hoist is λ ($\lambda \geq \nu$).

As described in Section I, the hoist moving in one cycle is exactly the same as those in any other cycle. There is one part entering the system at the beginning of each cycle. For the part entering the system at time 0, the starting time and ending time of move m_i for this part can be calculated from the parameters using the following formulas:

$$z_i^s = \sum_{k=1}^{i} \left(2\mu + \frac{|w_{s_{k-1}} - w_{s_k}|}{\nu} + \tau_k \right)$$
$$i = 1, \ldots, n; \quad z_0^s = 0 \text{ (starting points)} \tag{1}$$
$$z_i^e = z_i^s + 2\mu + \frac{|w_{s_i} - w_{s_{i+1}}|}{\nu}$$
$$i = 0, \ldots, n, \text{(ending points)}. \tag{2}$$

For the example shown in Fig. 2, the moves for the part that enters the system at time 0 are drawn in gray solid lines (lighter than the moves of other parts), and the starting and ending points of these moves are marked. In each production cycle, a move m_i starting from each station s_i, $i = 0, 1, \ldots, n$, is performed exactly once. For a given cycle length T, the starting times of all the required moves in a cycle can be calculated from z_i^s as follows:

$$t_i^s = z_i^s \bmod T, \quad i = 0, \ldots, n. \tag{3}$$

For convenience in later discussions, we define the ending time of moves, with respect to t_i^s, as follows:

$$t_i^e = t_i^s + 2\mu + \frac{|w_{s_i} - w_{s_{i+1}}|}{\nu}, \quad i = 0, \ldots, n. \tag{4}$$

Let $\theta_i = \lfloor z_i^s / T \rfloor$, then

$$z_i^s = \theta_i * T + t_i^s, \quad i = 0, \ldots, n. \tag{5}$$
$$z_i^e = \theta_i * T + t_i^e, \quad i = 0, \ldots, n. \tag{6}$$

The relationships among z_i^s, z_i^e, t_i^s, and t_i^e can be seen in Fig. 2.

Definition 1: A feasible hoist schedule is a cyclic schedule in which each part move is assigned to a hoist; between any two part moves assigned to the same hoist, there is enough time for the empty move of the hoist; and the distance between any two hoists is not shorter than d at any time.

Definition 2: A cycle length T is said to be feasible if a feasible cyclic hoist schedule exists with this cycle length.

The no-wait multihoist cyclic scheduling problem can then be defined as to find the shortest feasible cycle length T^* and a corresponding feasible schedule.

Since each station can process, at most, one part at a time, a feasible cycle length T cannot be shorter than the processing time of any station plus the time for moving the finished part away from this station and moving a new part into it. The two moves may be performed by different hoists but the hoists have to keep the safety distance. This provides a method to determine a lowerbound of the optimal cycle length. On the other hand, if a feasible solution exists for the problem, the total time spent by a part in the system provides an upperbound of the optimal cycle length. The above discussion is valid for systems with any number of hoists and directly leads to the following result.

Proposition 1: If there is a feasible solution for the no-wait multihoist scheduling problem, then $T_0 = \max\{\tau_i | i = 1, 2, \ldots, n\} + 2\mu + d/\nu$ is a lowerbound and $T^0 = z_n^e + |w_{s_{n+1}} - w_0|/\lambda$ is an upperbound for the optimal cycle length.

III. Hoist Assignment Constraints for a Fixed Cycle Length

For any given cycle length T, $T_0 \le T \le T^0$, the starting and ending times of all moves in a cycle are fixed in the time-way diagram and can be calculated from (1)–(4). The problem then becomes verifying the feasibility of the cycle length by searching for a feasible assignment of the moves to the hoists. In this section, we identify necessary conditions that must be satisfied by a feasible assignment compatible with the value T of the cycle length.

We define the following variables for the hoist assignment decisions.

x_i $(1 \le x_i \le m)$: the hoist number assigned to perform m_i, $i = 0, 1, 2, \ldots, n$, in every cycle.

A. Constraints by Positions of Individual Moves

Recall that the hoists run on the same track and are numbered from left to right. If the distance from the position of a tank to the track's leftmost position w_l is less than the safety distance d, then this tank can only be reached by H_1, and any move from or to this tank must be performed by H_1. Similarly, if the distance of a tank to w_l is less than $2d$, then the moves involving this tank can only be performed by H_1 or H_2. Considering the value range of x_i, $1 \le x_i \le m$, such restrictions can be represented by

$$x_i \le \min \left\{ 1 + \left\lfloor \frac{(\min\{w_{s_i}, w_{s_{i+1}}\} - w_l)}{d} \right\rfloor, m \right\}$$
$$i = 0, 1, 2, \ldots, n.$$

Similarly, the track's rightmost position w_r imposes the following constraint:

$$x_i \ge \max \left\{ m - \left\lfloor \frac{(w_r - \max\{w_{s_i}, w_{s_{i+1}}\})}{d} \right\rfloor, 1 \right\}$$
$$i = 0, 1, 2, \ldots, n.$$

These constraints set tighter bounds for x_i. Denoting $l_i = \max\left\{m - \left\lfloor \left(w_r - \max\{w_{s_i}, w_{s_{i+1}}\}\right)/d \right\rfloor, 1\right\}$ and $u_i = \min\left\{1 + \left\lfloor \left(\min\{w_{s_i}, w_{s_{i+1}}\} - w_l\right)/d \right\rfloor, m\right\}$, we can write them in the following form:

$$l_i \le x_i \le u_i \quad i = 0, \ldots, n. \tag{7}$$

Fig. 3 illustrates these constraints for a move. Note that these constraints are independent of T. Therefore, if they are infeasible, then the hoist scheduling problem is infeasible.

B. Constraints From Hoist Assignment of One Move

Since the hoists cannot be very close to each other, the hoist assignments of a pair of moves are also restricted by the rela-

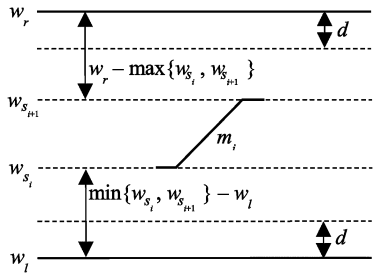

Fig. 3. Illustration of constraints by the position of an individual move.

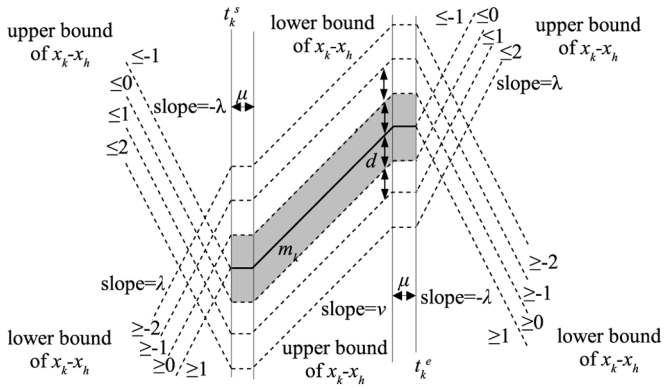

Fig. 4. Relation between a point and a move in a time-way diagram.

tionship of the two moves in time and in position. As preparation for presenting these move–pair relation constraints concisely (in Section III-C), we first discuss here the constraints imposed by the hoist assignment of one move on positions of all hoists. Consider a move m_k and an arbitrary point (t_h, w_h) in the time-way diagram. Suppose m_k is assigned to hoist x_k, then the hoists that can reach the point (t_h, w_h) are restricted by this assignment. We denote a hoist that can reach this point as x_h and identify the constraints that limit the possible values of x_h. The diagram in Fig. 4 illustrates the move m_k and the bounding lines around it. For point (t_h, w_h) being in different areas separated by these lines, the bounds for $x_k - x_h$ are different and marked in the diagram. From the diagram, we can see that the format of the constraints may be different when the point is at different positions with respect to move m_k. In the following, we discuss these different situations.

For the case $t_h \geq t_k^e$, we denote the position of hoist x_h at time t_k^e as p_h. Since m_k is performed by x_k, the position of x_k at time t_k^e will be $p_k = w_{s_{k+1}}$. Due to the safe distance d between two adjacent hoists, we have $w_{s_{k+1}} - p_h \geq (x_k - x_h)d$ for $x_h \leq x_k$ (i.e., for x_h is the same as x_k or is a hoist below x_k on the diagram), and $p_h - w_{s_{k+1}} \geq (x_h - x_k)d$ for $x_h \geq x_k$ (i.e., for x_h is the same as x_k or is a hoist above x_k on the diagram). To reach the point (t_h, w_h) from (t_k^e, p_h), hoist x_h should be able to arrive at position w_h on or before the time t_h in the maximum empty move speed λ. Therefore, we have

$$t_h \geq t_k^e + \frac{\left[w_h - w_{s_{k+1}} + (x_k - x_h)d\right]}{\lambda}, \text{ for } x_h \leq x_k$$
$$t_h \geq t_k^e + \frac{\left[w_{s_{k+1}} - (x_k - x_h)d - w_h\right]}{\lambda}, \text{ for } x_h \geq x_k.$$

Considering that x_h and x_k are integers, the first inequality above (for $x_h \leq x_k$) can be written as

$$x_k - x_h \leq \left\lfloor \frac{(w_{s_{k+1}} - w_h + (t_h - t_k^e)\lambda)}{d} \right\rfloor, \text{ for } x_h \leq x_k.$$

With the condition $x_h \leq x_k$, this constraint can be satisfied only when $\lfloor (w_{s_{k+1}} - w_h + (t_h - t_k^e)\lambda)/d \rfloor \geq 0$. Reversely, when $\lfloor (w_{s_{k+1}} - w_h + (t_h - t_k^e)\lambda)/d \rfloor < 0$ (i.e., the point (t_h, w_h) is above the top upwards-sloped dotted line on the right part of Fig. 4), the condition $x_h \leq x_k$ cannot be satisfied (i.e., it is impossible for hoist x_k or any hoist below it to reach point (t_h, w_h)) and, thus, we must have $x_k < x_h$ or, equivalently, $x_k - x_h \leq -1$ considering again the integer values of x_h and x_k. This analysis leads to the following constraint which is valid both for $x_h \leq x_k$ and for $x_h > x_k$. In the case $x_h \leq x_k$, it is equivalent to the first inequality above

$$x_k - x_h \leq \max\left\{-1, \left\lfloor \frac{(w_{s_{k+1}} - w_h + (t_h - t_k^e)\lambda)}{d} \right\rfloor\right\}.$$

In a similar way, we can also transform the second inequality (for $x_h \geq x_k$) to the following equivalent constraint which is always valid regardless of the relations between x_h and x_k:

$$x_k - x_h \geq \min\left\{1, \left\lceil \frac{(w_{s_{k+1}} - w_h - (t_h - t_k^e)\lambda)}{d} \right\rceil\right\}.$$

Combining the above two constraints, we know that $x_k - x_h$ is bounded by

$$\begin{aligned} &\min\left\{1, \left\lceil \frac{(w_{s_{k+1}} - w_h - (t_h - t_k^e)\lambda)}{d} \right\rceil\right\} \\ &\leq x_k - x_h \\ &\leq \max\left\{-1, \left\lfloor \frac{(w_{s_{k+1}} - w_h + (t_h - t_k^e)\lambda)}{d} \right\rfloor\right\}. \end{aligned} \quad (8)$$

Similarly, $x_k - x_h$ is restricted by the constraint below in the case of $t_h \leq t_k^s$

$$\begin{aligned} &\min\left\{1, \left\lceil \frac{(w_{s_k} - w_h + (t_h - t_k^s)\lambda)}{d} \right\rceil\right\} \\ &\leq x_k - x_h \\ &\leq \max\left\{-1, \left\lfloor \frac{(w_{s_k} - w_h - (t_h - t_k^s)\lambda)}{d} \right\rfloor\right\}. \end{aligned} \quad (9)$$

In the case of $t_k^s < t_h < t_k^e$ (t_h is within the duration of move m_k), the possible value of x_h depends on the vertical distance between the point (t_h, w_h) and the move m_k.

A move consists of three line segments in the time-way diagram. It has a starting point, an ending point, and two corner points in between (one close to starting point and the other close to the ending point). To simplify later presentations, we define the position of a move m_k (position of x_k as well) as a function of time t, in the range between t_k^s and t_k^e, as shown in the equation at the bottom of the next page.

During the period from t_k^s to t_k^e, x_k is performing m_k and the distance between it and any other hoist cannot be closer than d.

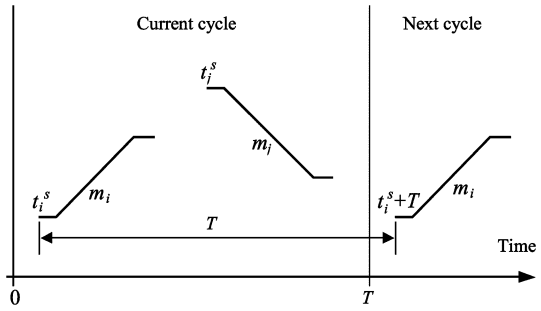

Fig. 5. Relations between two moves.

Therefore, if $|(f_k(t_h) - w_h)| < d$ (the shaded area in Fig. 4), there will be no feasible value for x_h.

If $(f_k(t_h) - w_h) \geq d$ (the point is below m_k), to ensure safe distance between the hoists, we must have $1 \leq x_k - x_h \leq \lfloor (f_k(t_h) - w_h)/d \rfloor$.

Similarly, if $(f_k(t) - w_h) \leq -d$, we must have $\lceil (f_k(t_h) - w_h)/d \rceil \leq x_k - x_h \leq -1$.

The constraints in the above three situations for $t_k^s < t_h < t_k^e$ can be summarized as shown below

$$\left.\begin{array}{ll} 1 \leq x_k - x_h \leq \left\lfloor \frac{(f_k(t_h) - w_h)}{d} \right\rfloor, & \text{if } (f_k(t_h) - w_h) \geq d \\ \left\lceil \frac{(f_k(t_h) - w_h)}{d} \right\rceil \leq x_k - x_h \leq -1, & \text{if } (f_k(t) - w_h) \leq -d \\ \text{infeasible}, & \text{if } |(f_k(t_h) - w_h)| < d \end{array}\right\}. \tag{10}$$

C. Constraints From Move Pairs

We now come back to a pair of moves m_i and m_j in the cycle ($i = 0, 1, \ldots, n$; $j = 0, \ldots, n$; $i \neq j$; $t_i^s \leq t_j^s$). If any one of them is assigned to a hoist, then the hoist assignment of the other is restricted. As shown in Fig. 5, such restrictions may exist between m_i and m_j in the same cycle and between m_j in one cycle and m_i in the next cycle. But they can be handled in the same way.

We consider the two moves in the same cycle first. Based on the discussions in the last subsection, if the hoist assignment constraint between one of the two moves and some point of the other is infeasible, then the cycle length is infeasible. Note that the constraint is in the format of inequality (8), (9), or (10), depending on the relative positions of the move and the point. If the constraints between one move and every point of the other are all feasible, then feasible hoist assignments for this pair of moves exist. In fact, as we will show below, there is no need to consider the constraints between one move and every point of the other.

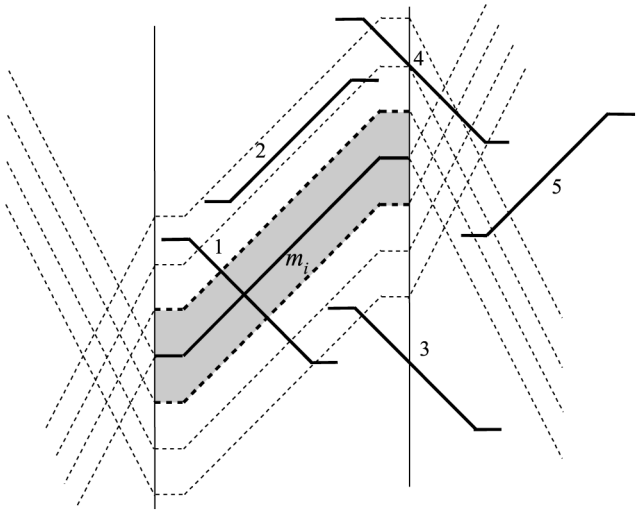

Fig. 6. Examples of relative positions between m_i and m_j.

Proposition 2: For a pair of moves m_i and m_j $\left(t_i^s \leq t_j^s\right)$, the hoist assignment variables x_i and x_j satisfy all of the constraints between one move and every point of the other if and only if they satisfy all of the constraints between one move and the starting, ending, and corner points of the other.

Proof: The condition is clearly necessary because if the constraint between a move and a point of the other is not satisfied, then the two hoists will violate the safety distance requirement at the time of that point.

We now prove that the condition is sufficient by considering different cases for the relative positions of a move pair m_i and m_j (e.g., different positions of m_j relative to m_i in Fig. 6). For simplicity, we will refer to the constraint from the relation between a point of one move and the other move as the constraint for that point. This will not cause confusion as there are only two moves involved.

Case 1) the two moves do not overlap in time (e.g., m_j is in position 5 for the example in Fig. 6). If x_i and x_j satisfy the constraint for $\left(t_j^s, w_{s_j}\right)$, the starting point of m_j, then they satisfy inequality (8), with m_i as m_k and $\left(t_j^s, w_{s_j}\right)$ as the point (t_h, w_h) in the formula. That is, $x_i - x_j$ is bounded as follows:

$$\begin{aligned} &\min\left\{1, \left\lceil \frac{\left(w_{s_{i+1}} - w_{s_j} - \left(t_j^s - t_i^e\right)\lambda\right)}{d} \right\rceil\right\} \\ &\leq x_i - x_j \\ &\leq \max\left\{-1, \left\lfloor \frac{\left(w_{s_{i+1}} - w_{s_j} + \left(t_j^s - t_i^e\right)\lambda\right)}{d} \right\rfloor\right\}. \end{aligned}$$

$$f_k(t) = \begin{cases} w_{s_k}, & \text{if } t_k^s \leq t < t_k^s + \mu \\ w_{s_k} + \nu\left(t - t_k^s - \mu\right), & \text{if } t_k^s + \mu \leq t \leq t_k^e - \mu \text{ and } w_{s_k} < w_{s_{k+1}} \\ w_{s_k} - \nu\left(t - t_k^s - \mu\right), & \text{if } t_k^s + \mu \leq t \leq t_k^e - \mu \text{ and } w_{s_k} > w_{s_{k+1}} \\ w_{s_{k+1}}, & \text{if } t_k^e - \mu < t \leq t_k^e \end{cases}$$

Since $\lambda \geq \nu > 0$ (i.e., the line segments of m_j are flatter than or with the same slope as the bounding lines on the right of m_i), we have $\left(t_h - t_j^s\right)\lambda \geq |w_h - w_{s_j}|$ for every point (t_h, w_h) on m_j other than the starting point. Then, if we replace w_{s_j} and t_j^s in the above bounds with w_h and t_h, respectively, the lowerbound will be unchanged or become smaller, the upperbound will be unchanged or become larger, and, hence, $x_i - x_j$ will be still within the resulting new bounds. That is, if x_i and x_j satisfy the constraint for the starting point of m_j, they also satisfy the constraints between m_i and all other points on m_j. Therefore, it is enough to consider only the constraint for the start point of m_j. Equivalently, it is enough to consider only the ending point of m_i.

Case 2) the two moves overlap in time. We discuss this case in two more detailed cases.

2.1) the two moves cross each other (e.g., m_j is in position 1 in Fig. 6). This is a definitely infeasible case. Since the two moves cross each other, they overlap in time. Point $\left(t_j^s, w_{s_j}\right)$, the starting point of m_j, is on the left boundary of the overlapping period. On the right boundary is either $\left(t_i^e, w_{s_{i+1}}\right)$ or $\left(t_j^e, w_{s_{j+1}}\right)$, the ending point of m_i or m_j. If the distance from the point on either boundary to the other move is less than d, then the infeasibility is identified by constraint (10) for this point. Otherwise, if $\left(t_j^e, w_{s_{j+1}}\right)$ is on the right boundary, the two points $\left(t_j^s, w_{s_j}\right)$ and $\left(t_j^e, w_{s_{j+1}}\right)$ must be on different sides of m_i due to the crossing. Then, constraint (10) for the point above m_i requires $x_i - x_j \leq -1$ but another constraint (10) for the point below m_i requires $x_i - x_j \geq 1$, which indicates infeasibility. Similarly, if $\left(t_i^e, w_{s_{i+1}}\right)$ is on the right boundary, then one constraint (10) for $\left(t_j^e, w_{s_{j+1}}\right)$ and another constraint (10) for $\left(t_i^e, w_{s_{i+1}}\right)$ will conflict with each other. Therefore, when the two moves cross each other, the infeasibility can always be identified by the constraints between a move and the starting and ending points of the other.

2.2) the two moves do not cross each other (e.g., m_j is in positions 2, 3, or 4 in Fig. 6). If part of m_j is on the right of the vertical line $t = t_i^e$ as in situation 3 or 4 in Fig. 6, the constraint for the point of m_j at the time t_i^e will also be satisfied for all of the points of m_j on its right, as can be proved in the same way as for Case 1) (because $\lambda \geq \nu > 0$). For the same reason, all of the constraints for the points outside the two moves' overlapping period need not be considered. The hoist assignment constraint from the two moves is then defined by the closest vertical distance of the two moves in the overlapping period. Now, we prove that the closest distance between the two moves in the overlapping period is reached at one of the starting, ending, or corner points. Suppose the closest distance occurs between a midpoint on a segment of one move and a midpoint on a segment of the other move, then these two segments of the two moves must be parallel. Then, moving from this point to right (or to the left), the vertical distance between the two parallel line segments is always the closest distance until, and including, the end of one of these two segments. This end of the line segment must be an ending point (or starting point, if moving to the left) or a corner point of a move. Hence, it is sufficient to consider only the constraints for the starting, ending, and corner points.

The above discussions on the sufficiency of the condition have included all possible cases. The proof is complete. ■

From the proof of Proposition 2, we can see that, for m_i and m_j $\left(t_i^s \leq t_j^s\right)$ in the same cycle, it is enough to consider only the constraints between m_i and the start point of m_j if there is no overlapping in time between the two moves $\left(t_j^s > t_i^e\right)$; and if there is overlapping in time between them, it is enough to consider only the constraints between each move and the starting, ending, and corner points of the other in the overlapping time period. Altogether, we have the following constraints for the restrictions between m_i and m_j in the same cycle.

Case 1) Between m_i and the starting point of m_j, if $t_i^e < t_j^s$

$$\min\left\{1, \left\lceil \frac{\left(w_{s_{i+1}} - w_{s_j} - \left(t_j^s - t_i^e\right)\lambda\right)}{d} \right\rceil\right\} \leq x_i - x_j \leq \max\left\{-1, \left\lfloor \frac{\left(w_{s_{i+1}} + \left(t_j^s - t_i^e\right)\lambda - w_{s_j}\right)}{d} \right\rfloor\right\}.$$

Case 2) Between m_i and the starting point of m_j, if $t_j^s \leq t_i^e$, as shown

$$\begin{cases} \left\lceil \frac{\left(f_i\left(t_j^s\right) - w_{s_j}\right)}{d} \right\rceil \leq x_i - x_j \leq -1, & \text{if } f_i\left(t_j^s\right) - w_{s_j} \leq -d \\ 1 \leq x_i - x_j \leq \left\lfloor \frac{\left(f_i\left(t_j^s\right) - w_{s_j}\right)}{d} \right\rfloor, & \text{if } f_i\left(t_j^s\right) - w_{s_j} \geq d \\ \text{infeasible}, & \text{if } \left|\left(f_i\left(t_j^s\right) - w_{s_j}\right)\right| < d. \end{cases}$$

$$\begin{cases} \left\lceil \frac{\left(f_i\left(t_j^s + \mu\right) - w_{s_j}\right)}{d} \right\rceil \leq x_i - x_j \leq -1, & \text{if } f_i\left(t_j^s + \mu\right) - w_{s_j} \leq -d \\ 1 \leq x_i - x_j \leq \left\lfloor \frac{\left(f_i\left(t_j^s + \mu\right) - w_{s_j}\right)}{d} \right\rfloor, & \text{if } f_i\left(t_j^s + \mu\right) - w_{s_j} \geq d \\ \text{infeasible}, & \text{if } \left|f_i\left(t_j^s + \mu\right) - w_{s_j}\right| < d \end{cases}$$

Case 3) Between m_i and the corner point $\left(t_j^s+\mu, w_{s_j}\right)$ of m_j, if $t_j^s+\mu \le t_i^e$, as shown in the first equation at the bottom of the previous page.

Case 4) Between m_i and the corner point $\left(t_j^e-\mu, w_{s_{j+1}}\right)$ of m_j, if $t_j^e-\mu < t_i^e$, as shown in the first equation at the bottom of the page.

Case 5) Between m_i and the ending point of m_j, if $t_j^e \le t_i^e$, as shown in the second equation at the bottom of the page.

Case 6) Between m_j and the corner point $(t_i^s+\mu, w_{s_i})$ of m_i, if $t_j^s \le t_i^s+\mu \le t_j^e$, as shown in the third equation at the bottom of the page.

Case 7) Between m_j and the corner point $\left(t_i^e-\mu, w_{s_{i+1}}\right)$ of m_i, if $t_j^s \le t_i^e-\mu \le t_j^e$, as shown in the fourth equation at the bottom of the page.

Case 8) Between m_j and the ending point of m_i, if $t_j^s \le t_i^e \le t_j^e$, as shown in the fifth equation at the bottom of the page.

$$\begin{cases} \left\lceil \frac{\left(f_i\left(t_j^e-\mu\right)-w_{s_{j+1}}\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } f_i\left(t_j^e-\mu\right)-w_{s_{j+1}} \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(f_i\left(t_j^e-\mu\right)-w_{s_{j+1}}\right)}{d} \right\rfloor, & \text{if } f_i\left(t_j^e-\mu\right)-w_{s_{j+1}} \ge d \\ \text{infeasible}, & \text{if } \left|f_i\left(t_j^e-\mu\right)-w_{s_{j+1}}\right| < d \end{cases}$$

$$\begin{cases} \left\lceil \frac{\left(f_i\left(t_j^e\right)-w_{s_{j+1}}\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } f_i\left(t_j^e\right)-w_{s_{j+1}} \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(f_i\left(t_j^e\right)-w_{s_{j+1}}\right)}{d} \right\rfloor, & \text{if } f_i\left(t_j^e\right)-w_{s_{j+1}} \ge d \\ \text{infeasible}, & \text{if } \left|f_i\left(t_j^e\right)-w_{s_{j+1}}\right| < d \end{cases}$$

$$\begin{cases} \left\lceil \frac{\left(w_{s_i}-f_j\left(t_i^s+\mu\right)\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } w_{s_i}-f_j\left(t_i^s+\mu\right) \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(w_{s_i}-f_j\left(t_i^s+\mu\right)\right)}{d} \right\rfloor, & \text{if } w_{s_i}-f_j\left(t_i^s+\mu\right) \ge d \\ \text{infeasible}, & \text{if } \left|w_{s_i}-f_j\left(t_i^s+\mu\right)\right| < d \end{cases}$$

$$\begin{cases} \left\lceil \frac{\left(w_{s_{i+1}}-f_j\left(t_i^e-\mu\right)\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } w_{s_{i+1}}-f_j\left(t_i^e-\mu\right) \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(w_{s_{i+1}}-f_j\left(t_i^e-\mu\right)\right)}{d} \right\rfloor, & \text{if } w_{s_{i+1}}-f_j\left(t_i^e-\mu\right) \ge d \\ \text{infeasible}, & \text{if } \left|w_{s_{i+1}}-f_j\left(t_i^e-\mu\right)\right| < d \end{cases}$$

$$\begin{cases} \left\lceil \frac{\left(w_{s_{i+1}}-f_j\left(t_i^e\right)\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } w_{s_{i+1}}-f_j\left(t_i^e\right) \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(w_{s_{i+1}}-f_j\left(t_i^e\right)\right)}{d} \right\rfloor, & \text{if } w_{s_{i+1}}-f_j\left(t_i^e\right) \ge d \\ \text{infeasible}, & \text{if } \left|w_{s_{i+1}}-f_j\left(t_i^e\right)\right| < d \end{cases}$$

Case 1) corresponds to the situation where there is no overlapping in time between the two moves $\left(t_j^s > t_i^e\right)$. Cases 2)–8) are for the situations where there is overlapping in time between the two moves.

For the restrictions between m_j in one cycle and m_i in the next cycle $\left(t_j^s \le t_i^s + T\right)$, we can get the constraints in the same way. For simplicity, we use m_i' to mean "m_i in the next cycle." The case where they do not have overlapping $\left(t_j^e \le T + t_i^s\right)$ is similar to case 1) from before, and the constraints can be easily written

$$\begin{aligned}&\min\left\{1, \left\lceil \frac{\left(w_{s_{j+1}} - w_{s_i} - \left(t_i^s + T - t_j^e\right)\lambda\right)}{d} \right\rceil\right\}\\ &\quad \le x_j - x_i\\ &\quad \le \max\left\{-1, \left\lfloor \frac{\left(w_{s_{j+1}} + \left(t_i^s + T - t_j^e\right)\lambda - w_{s_i}\right)}{d} \right\rfloor\right\}.\end{aligned}$$

We change the sign of this inequality to make it as a constraint on $x_i - x_j$

$$\begin{aligned}&\min\left\{1, \left\lceil \frac{\left(w_{s_i} - w_{s_{j+1}} - \left(t_i^s + T - t_j^e\right)\lambda\right)}{d} \right\rceil\right\}\\ &\quad \le x_i - x_j\\ &\quad \le \max\left\{-1, \left\lfloor \frac{\left(w_{s_i} + \left(t_i^s + T - t_j^e\right)\lambda - w_{s_{j+1}}\right)}{d} \right\rfloor\right\}.\end{aligned}$$

Similarly, we can obtain constraints for the restrictions between m_j and m_i' in all cases. These are listed in Appendix A as Cases a)–h). Case a) corresponds to the situation where there is no overlapping in time between m_j and m_i' $\left(T + t_i^s > t_j^e\right)$. Cases b)–h) are for the situations where there is overlapping in time between the two moves.

When the cycle length is fixed, for any pair of moves m_i and m_j ($i = 0, 1, \ldots, n$; $j = 0, \ldots, n$; $i \ne j$; $t_i^s \le t_j^s$), a set of hoist assignment constraints based on the relationship between the two moves can be obtained as discussed before. Note that some of the cases may not be relevant for a particular move pair. According to the relative positions of the two moves, we can list all of the relevant constraints by checking the Cases 1)–h). If any of the constraints is infeasible, then the cycle length is infeasible. Otherwise, it is easy to see that all of the constraints are in the format $L_{ij}^k \le x_i - x_j \le U_{ij}^k$. Calculating $L_{ij} = \max_k\left\{L_{ij}^k\right\}$ and $U_{ij} = \min_k\left\{U_{ij}^k\right\}$, we can combine these constraints into one $L_{ij} \le x_i - x_j \le U_{ij}$.

Considering all move pairs, we have the following hoist assignment constraints. Each constraint is a "difference constraint" that sets lower and upperbounds for the difference of two variables

$$\begin{aligned}&L_{ij} \le x_i - x_j \le U_{ij}, \quad i = 0, 1, \ldots, n\\ &\quad j = 0, \ldots, n; \quad i \ne j; \ t_i^s \le t_j^s.\end{aligned} \tag{11}$$

IV. Hoist Scheduling With a Given Cycle Length

A. Problem Transformation

When the cycle length is given, our problem becomes to check whether this cycle length is feasible. To solve this problem, we first transform it to a problem of solving a group of difference constraints.

Proposition 3: A cycle length T is feasible if and only if a feasible integer solution exists for the group of constraints (7) and (11) corresponding to T.

Proof: From the process of deriving the constraints in the last section, it is obvious that all of the constraints are necessary conditions for a cycle length being feasible. We only need to show that this group of constraints is also sufficient for a cycle length being feasible.

When the constraint group for a cycle length has a feasible integer solution, to prove the cycle length is feasible, we need to show: 1) that every move is assigned to a hoist and 2) that there is a feasible route for each hoist in a complete cycle (i.e., the moves assigned to the same hoist can be feasibly linked in the time-way diagram to form a route for the hoist and the routes of any two adjacent hoists keep at least a safety distance from each other all the time).

1) From an integer feasible solution of constraints (7) and (11), we can get an integer value for each x_i, $i = 0, 1, \ldots, n$. Constraint (7) ensures that the x_i values all satisfy $1 \le x_i \le m$. Therefore, every move is assigned to one of the m hoists for a feasible integer solution. Note that different x variables may take the same value (i.e., a hoist may be assigned to perform several moves). But the hoist assignment satisfies constraint (11) between any pair of moves (i.e., after performing one of the moves, there is enough time for the hoist to travel to the next move and perform it).
2) Now we show the existence of a feasible schedule by constructing a route for each hoist in the time-way diagram and proving the feasibility of these routes. To construct the routes, we order all moves in cycle $[0, T]$ in ascending order of their starting times. The corresponding ordered moves are denoted by $m_{(0)}, m_{(1)}, \ldots, m_{(n)}$. Without loss of generality, we define the moves performed by hoist k $(k = 1, 2, \ldots, m)$ to be $m_{(i_1^k)}, m_{(i_2^k)}, \ldots, m_{(i_{n_k}^k)}$ where $i_1^k < i_2^k < \cdots < i_{n_k}^k$. For each hoist k, we need to prove that there is a feasible route for it to travel for a complete cycle, including the link that crosses the cycle boundary (starts in one cycle and completes in the next) (e.g., from $m_{i_{n_k}^k}$ in cycle $[-T, 0]$ to $m_{i_1^k}$ in cycle $[0, T]$). For convenience, we refer to the move $m_{i_{n_k}^k}$ in cycle $[-T, 0]$ as $m_{i_0^k}$ in the current cycle $[0, T]$.

We first construct a route for H_1. To construct the route, we only need to construct a link between each adjacent pair of moves $m_{\left(i_j^1\right)}$ and $m_{\left(i_{j+1}^1\right)}$, $j = 0, 1, \ldots, n_1 - 1$. We can construct the link in the following way (see Fig. 7 for illustrations). In the time-way diagram, starting from the ending point of $m_{\left(i_j^1\right)}$ (point A), draw a line p_1 downwards with slope $-\lambda$, which intersects the horizontal line $w = w_l$ at point B. From the starting point of $m_{\left(i_{j+1}^1\right)}$ (point D), draw a line p_2 downwards with slope λ, which intersects the horizontal line $w = w_l$ at point C. If the lines p_1 and p_2 do not intersect above or on the horizontal line $w = w_l$ as shown in Fig. 7(a), the link is constructed using the three line segments AB, BC, and CD. If the lines p_1 and p_2

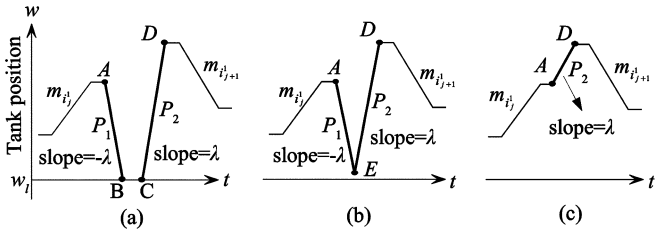

Fig. 7. Hoist route between $m_{(i_j^1)}$ and $m_{(i_{j+1}^1)}$.

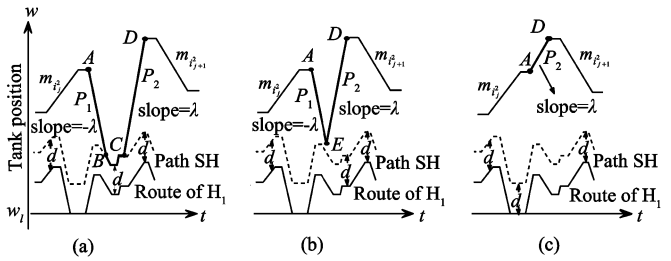

Fig. 8. Hoist route between $m_{(i_j^k)}$ and $m_{(i_{j+1}^k)}$.

intersect at a point E above or on the horizontal line $w = w_l$ as shown in Fig. 7(b), the link is constructed using the two line segments AE and ED. If point E is overlapped with point A or D as shown in Fig. 7(c), the link is constructed by connecting the two points A and D directly. In all of the cases, the link is feasible (i.e., the hoist can feasibly travel from $m_{(i_j^1)}$ to $m_{(i_{j+1}^1)}$ along the link), because the hoist speeds required on the link segments are all within the maximum empty hoist speed λ.

To show that this route of H_1 does not conflict with the moves assigned to other hoists, we consider any move $m_{(i_j^k)}$ which is assigned to a hoist H_k $(k \neq 1)$. If any part of $m_{(i_j^k)}$ is in a time interval that contains a move or the sloped link segments on the route of H_1, then from the above route construction method, we know that the move–pair constraints ensure that $m_{(i_j^k)}$ is at least $(k-1)d$ distance vertically above the route of H_1 in this interval. If any part on $m_{(i_j^k)}$ is in a time interval that contains a horizontal link segment $(w = w_l)$ on the route of H_1, then from the individual move constraints, we know that $m_{(i_j^k)}$ is also at least $(k-1)d$ distance vertically above the route of H_1 in this interval. In summary, any move $m_{(i_j^k)}$ $(k \neq 1)$ is always at least $(k-1)d$ distance vertically above the route of H_1 (i.e., the route of H_1 does not conflict with the moves assigned to other hoists).

Now, we imagine a path SH that is exactly d distance above the route of H_1 and then construct the route for H_2 by linking each pair of moves $m_{(i_j^2)}$ and $m_{(i_{j+1}^2)}$, $j = 0, 1, \ldots, n_2 - 1$ (see Fig. 8 for illustrations). Starting from the ending point of $m_{(i_j^2)}$ (point A), draw a line p_1 downwards with slope $-\lambda$, which intersects the path SH at point B. From the starting point of $m_{(i_{j+1}^2)}$ (point D), draw a line p_2 downwards with slope λ, which intersects the path SH at point C. If the lines p_1 and p_2 do not intersect above or on SH as shown in Fig. 8(a), the link is constructed using the line segment AB, path BC and line segment CD. If the lines p_1 and p_2 intersect at a point E above or on SH as shown in Fig. 8(b), the link is constructed using the two line segments AE and ED. If point E is overlapped with point A or D, as shown in Fig. 8(c), the link is constructed by connecting the two points A and D directly. Clearly, the route of H_2 constructed this way keeps at least d distance from the route of H_1 because it never goes below path SH. Similar to the situation for H_1, it can be shown that the route of H_2 does not conflict with moves assigned to hoists $H_3, \ldots, H_m$.

Imagining a path SH that is exactly d distance above the route of H_2, we can construct the route for H_3 in the same way as for H_2. Similarly, we can construct the routes for other hoists, one-by-one, and show that the routes for any adjacent hoists always keep at least a distance of d. Now consider the route of H_m, the individual move constraints guaranteed that the moves on the route are all below the line $w = w_r$. From the route construction method, we can see that the link between any two moves is always lower than the highest point of the two moves. Thus, the whole route of H_m is below the line $w = w_r$. In summary, the routes of the hoists constructed above are all within the vertical range $[w_l, w_r]$ and always keep at least a distance d from each other. Therefore, the hoist routes are feasible (i.e., the corresponding cycle length T is feasible). ■

The proof of this proposition provides a method to construct a feasible schedule. Since each move needs to be linked to only one other move on its right in the route of the hoist assigned to them, we only need to construct $n+1$ move links. The number of line segments for each link is proportional to the number of moves $(\leq n)$ of the hoist routes below it. Therefore, the computational complexity for constructing a feasible schedule is $O(n^2)$.

Note that the route construction method given in the above proof is mainly for showing the existence of a feasible schedule when the group of constraints (7) and (11) has a feasible integer solution. In the resulting hoist routes, however, the empty hoist moves may be unnecessarily long as can be seen from Figs. 7 and 8. With proven existence of a feasible schedule, schedules can actually be constructed differently, in which most links between the loaded moves in the hoist routes can be straight lines. For example, we can modified the schedule constructed in the proof in the following way. Modify the routes of the hoists, one by one from Hoist m to Hoist 1: between each pair of adjacent moves assigned to the hoist, let the empty hoist move along a straight line on the time-way diagram; whenever it reaches a point that has a distance of d to the route of an adjacent hoist, detour along a path that keeps the safety distance of d to the route of that adjacent hoist until it comes back to the straight line. It can be seen that with this modification step, the complexity for constructing the schedule is still $O(n^2)$.

B. Problem Solution

For a given cycle length, we can construct constraints (7) and (11). By introducing an auxiliary variable $x_s \equiv 0$, we can transform constraint (7) to the following difference constraint:

$$l_i \leq x_i - x_s \leq u_i \quad i = 0, \ldots, n. \tag{12}$$

According to Proposition 3, the hoist scheduling problem with a given cycle length becomes a problem to find whether there exists an integral solution to the corresponding system of difference constraints (11) and (12).

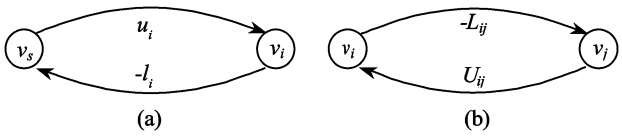

Fig. 9. Constructing a graph for the system of difference constraints.

Such a system of difference constraints can be represented by a weighted directed graph such that the system is feasible if and only if there exists no negative weight cycle in the graph ([20, pp. 602–605]).

The graph $G(V,E)$, corresponding to constraints (11) and (12), is as follows. There are $n+2$ nodes in the graph $v_s, v_0, v_1, \ldots, v_n$, where v_s is the source node corresponding to x_s and node v_i corresponds to the decision variable x_i $(i=0,\ldots,n)$. The graph contains $(n+1)(n+2)$ arcs, each corresponding to one bound of a difference constraint. Between v_s and v_i $(i=0,\ldots,n)$, there are two directed arcs e_{si} from v_s to v_i and e_{is} from v_i to v_s, as shown in Fig. 9(a). The weights of e_{si} and e_{is} are $\pi_{si}=u_i$ and $\pi_{is}=-l_i$, respectively. Between any v_i and v_j $(i=0,\ldots,n-1;\ j=i+1,\ldots,n)$, there are also two directed arcs e_{ij} from v_i to v_j and e_{ji} from v_j to v_i, as shown in Fig. 9(b). The weights of arcs e_{ij} and e_{ji} are $\pi_{ij}=-L_{ij}$ and $\pi_{ji}=U_{ij}$, respectively. If a negative weight cycle exists in the graph $G(V,E)$, there is no feasible solution for constraints (11) and (12). Otherwise, the total weight of the shortest path from v_s to v_i $(i=0,\ldots,n)$ is the solution of x_i. Since all π_{ij}'s are integer, it is guaranteed that the solution is integer.

The problem of determining whether graph $G(V,E)$ has a negative weight cycle can be solved using the Bellman–Ford algorithm (see [20]–[22]). Systems of difference constraints and the Bellman–Ford algorithm have been used to solve some other scheduling problems. For example, [16] used a system of difference constraints, with move starting times as variables, to check the feasibility of a given sequence of moves for a single hoist. Reference [23] used a modified Bellman–Ford algorithm to minimize the cycle length for a given robot route.

The following is a procedure, based on the Bellman–Ford algorithm, for solving the system of constraints (11) and (12).

Procedure 1: Feasibility Checking

Set $x_s=0$ and $k=0$.

Set $x_i=x_s+\pi_{si}$ for $i=1,\ldots,n$.

While $k<n+2$: {

If $x_i+\pi_{ij}<x_j$ holds for some arcs $e_{ij}\in E$, then {

For each arc $e_{ij}\in E$: {

Correct and update $x_j=x_i+\pi_{ij}$ if $x_i+\pi_{ij}<x_j$. }

Set $i=i+1$. }

Else, {

Stop. x_i's are a feasible solution. }}

There is a negative cycle and the constraint system is infeasible.

The computational complexity of the algorithm is $O(|V||E|)$. Since $|V|=n+2$, $|E|=(n+1)(n+2)$, the complexity of this procedure is $O(n^3)$.

V. Special Values of the Cycle Length

Although the cycle length may take any real value between T_0 and T^0, some of the values are infeasible (the corresponding schedule is not feasible) and the feasibility changes only at some special values when T increases from T_0 to T^0. In this section, we identify all of the special values at which the feasibility may potentially change.

Proposition 4: When the cycle length increases from T_0 to T^0, the feasibility may change (from infeasible to feasible or from feasible to infeasible) only if there exists a pair of i and j $(i=0,1,\ldots,n-1;\ j=i+1,\ldots,n)$ such that the value of L_{ij} or U_{ij} corresponding to the cycle length changes.

Proof: For a given cycle length, the feasibility can be checked by solving constraints (11) and (12). As discussed in Section III-A, constraints (12) are independent of cycle length and are kept unchanged when cycle length changes. Therefore, the feasibility may change only if there is at least one constraint in constraint set (11) changes (i.e., one of the bounds in the constraint changes). Therefore, feasibility may change only if there exists at least one pair of i and j such that L_{ij} or U_{ij} changes. ■

Recall that L_{ij} and U_{ij} are obtained by checking the Cases 1)–h) in Section III-C. We first discuss the changes of L_{ij} and U_{ij} for Case 1) (assume $i<j$).

If Case 1) in Section III-C dominates the other cases for L_{ij}, we have

$$L_{ij}=\min\left\{1,\left\lceil\frac{\left(w_{s_{i+1}}-w_{s_j}-\left(t_j^s-t_i^e\right)\lambda\right)}{d}\right\rceil\right\}$$

and L_{ij} may change only when

$$\frac{\left(w_{s_{i+1}}-w_{s_j}-\left(t_j^s-t_i^e\right)\lambda\right)}{d}=-p,\quad \text{where } p=0,1,\ldots,m-1.$$

Since $t_i^e\le t_j^s$, $t_i^e=z_i^e-\theta_i T$ and $t_j^s=z_j^s-\theta_j T$, the cycle lengths that may cause changes of L_{ij} are

$$T=\frac{\left[z_j^s-z_i^e-\frac{(w_{s_{i+1}}-w_{s_j}+pd)}{\lambda}\right]}{(\theta_j-\theta_i)},\ \text{if } w_{s_j}<w_{s_{i+1}}+pd.$$

Since the cycle length cannot be shorter than the processing time at a station plus the move time to the next station, we have $\theta_j-\theta_i=1,\ldots,j-i$ and the above cycle lengths can be expressed as

$$T=\frac{\left[z_j^s-z_i^e-\frac{(w_{s_{i+1}}-w_{s_j}+pd)}{\lambda}\right]}{k},\ \text{if } w_{s_j}<w_{s_{i+1}}+pd$$

where $p=0,1,\ldots,m-1$ and $k=1,\ldots,j-i$.

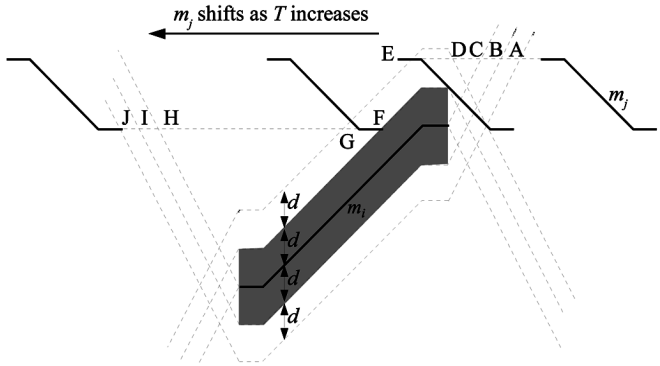

Fig. 10. Special T values corresponding to the changes of L_{ij} and U_{ij} as an example.

Similarly, if Case 1) in Section III-C dominates the other cases for U_{ij}, we have

$$U_{ij} = \max\left\{-1, \left\lfloor \frac{\left(w_{s_{i+1}} + \left(t_j^s - t_i^e\right)\lambda - w_{s_j}\right)}{d} \right\rfloor\right\}$$

and U_{ij} may change only when

$$\frac{\left(w_{s_{i+1}} + \left(t_j^s - t_i^e\right)\lambda - w_{s_j}\right)}{d} = p, \text{ where } p = 0, \ldots, m-1.$$

Since $t_i^e = z_i^e - \theta_i T$, $t_j^s = z_j^s - \theta_j T$ and $t_i^e \le t_j^s$, the cycle lengths corresponding to the changes of U_{ij} are

$$T = \frac{\left[z_j^s - z_i^e - \frac{(w_{s_j} + pd - w_{s_{i+1}})}{\lambda}\right]}{k}, \text{ if } w_{s_j} > w_{s_{i+1}} - pd$$

where $p = 0, 1, \ldots, m-1$ and $k = 1, \ldots, j-i$.

In a similar way, for a pair of moves i and j with $i < j$, we can find special values of the cycle length corresponding to possible changes of L_{ij} and U_{ij} for all Cases 1)–h). Detailed formulas for these special values are listed in Appendix B.

For each class of the above special cycle lengths, the value of k can be $1, 2, \ldots, j-i$. Therefore, for a pair i and j, the total number of these special cycle lengths corresponding to the changes of L_{ij} and U_{ij} is bounded by $O(mn)$.

The example shown in Fig. 10 gives an explanation of the meanings of these special values. The example shows a pair of moves m_i and m_j $(i < j)$ for a problem with three hoists. Imagine we increase the cycle length continuously, m_j will move to the left with respect to m_i. At the beginning, the starting point of m_j is to the right of point A. When the cycle length increases, m_j passes through points A, B, C D, and E one by one. U_{ij} changes from 2 to 1 when the starting point of m_j passes through points A. Similarly, U_{ij} changes from 1 to 0 and 0 to -1 when the starting point of m_j passes through points B and C, respectively. When the starting point of m_j passes through points D and E, L_{ij} changes from -2 to -1 and -1 to 0, respectively. After its starting point passes through point E, m_j enters the shaded region and the corresponding cycle length becomes infeasible. At this time, we have $\left|w_{s_{i+1}} - f_j\left(t_i^e\right)\right| < d$ to reflect the infeasibility.

When cycle length increases further, m_j will leave the shaded region and its ending point passes point F. At this time, when L_{ij} changes from 0 to -1, we have $L_{ij} = U_{ij} = -1$. L_{ij} changes from -1 to -2 when the ending point of m_j passes point G. Similarly, U_{ij} changes from -1 to 0, 0 to 1, and 1 to 2 when the ending point of m_j passes points H, I, and J, respectively.

From the above analysis, we know that for this example, the pair of moves m_i and m_j defines a total of 10 special cycle length values, corresponding to points A to J, respectively.

VI. Complete Solution to the No-Wait Problem

With all of the results of the previous sections, we know that the no-wait multihoist scheduling problem under study can be solved by generating the lowerbound, the upperbound, and all of the special values of the cycle length, and then checking the feasibility of these special values in ascending order starting from the lowerbound. As the feasibility may only change at these values, the first feasible value will be the optimal cycle length and the corresponding hoist schedule is an optimal schedule. The overall algorithm to obtain the optimal solution is summarized as follows:

Procedure 2: Algorithm to solve the no-wait problem

Read problem data and calculate T_0 and T^0.

For $i = 0, \ldots, n$: {

$l_i = \lfloor(\min(w_{s_i}, w_{s_{i+1}}) - w_l)/d\rfloor + 1$.

$u_i = m - \lfloor(w_r - \max(w_{s_i}, w_{s_{i+1}}))/d\rfloor$.

If $l_i > u_i$, the problem is infeasible. stop. }

Calculate the special values for each pair of moves according to the formulas in **Section V**.

Order all of the special values in ascending order and get rid of the duplicates and the values outside the interval $\left[T_0, T^0\right]$. Call the resulting ordered values "checking points" (cycle lengths) and name them $\gamma_1, \gamma_2, \ldots, \gamma_N$.

Set $k = 1$.

$(*)$ While $k < N + 1$: {

Let $T = \gamma_k$ and calculate t_i^s and t_i^e for $i = 0, \ldots, n$.

For $i = 0, \ldots, n-1$; $j = i+1, \ldots, n$: {

Construct L_{ij} and U_{ij} by checking Cases 1)–h) in **Section III-C**. If $L_{ij} > U_{ij}$, γ_k is infeasible, set $k = k + 1$, go back to step $(*)$. }

Apply Procedure 1 to determine whether there is a feasible solution to the constraints **(11)** and **(12)**. If it is infeasible, set $k = k + 1$, go back to step $(*)$. Otherwise, γ_k is the optimal cycle length; construct an optimal schedule using the method described in **Section IV-A**; stop. }

All of the checking points are infeasible. Stop; the problem is infeasible.

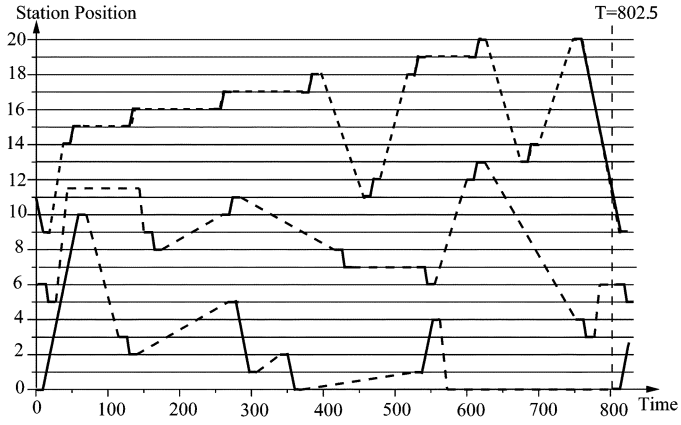

Fig. 11. No-wait solution for the example.

Proposition 5: The no-wait multihoist cyclic scheduling problem with fixed processing and transfer times is polynomially solvable.

Proof: Procedure 2 solves the problem because it generates an optimal hoist schedule when the problem is feasible and it indicates infeasibility otherwise.

In the procedure, for each pair of moves, m_i and m_j, the number of checking points is bounded by $O(mn)$. The total number of checking points is then $O(mn^3)$. The complexity of Procedure 1 for checking feasibility of a checking point is $O(n^3)$. Hence, the complexity to check all of the checking points is $O(mn^6)$. Ordering the special values is completed with the complexity of $O(mn^3 \log mn)$ before the checking process. The optimal schedule is constructed with the complexity of $O(n^2)$ after the checking process. Therefore, the computational complexity of the complete algorithm is $O(mn^6)$. ■

Illustrative example

As an illustrative example, we apply the algorithm to the following three-hoist problem ($m = 3$). The system has 20 processing stations ($n = 20$) and a further station (station 0) for both loading and unloading. The positions of the stations are $w_i = i$, $i = 0, 1, \ldots, n$. The position limits for the hoists on the track are $w_l = 0$, $w_r = 20$. The safety distance between hoists is 1.5. The loaded and empty hoist speeds are $\nu = 0.2$ and $\lambda = 0.4$, respectively. The time for lifting up or dropping off a part at a station is $\mu = 10$. The order of stations (the s_i's) that the parts visit is $[0, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 9, 8, 7, 6, 5, 1, 4, 3, 2, 0]$. The processing times at the stations, in the above order, are $[190, 170, 120, 50, 140, 60, 100, 100, 120, 60, 120, 130, 240, 90, 250, 240, 220, 190, 140, 200]$.

The algorithm takes only 0.025 s (on a Pentium III PC with 750 MHz CPU) to obtain an optimal no-wait schedule with $T = 802.5$ (see Fig. 11). From the results, we can see that the algorithm is very efficient.

To observe the impact of system settings on the optimal cycle length and to demonstrate the efficency of our algorithm in different situations, we change the above example data to generate more instances in the following two ways: 1) try different number of hoists from 1 to 5; 2) change the range (the leftmost and rightmost positions) that hoists can travel on the track. The combination of these provides 25 different instances including the original one shown above. Table I shows the results of applying the algorithm to these instances. Two numbers are shown for each problem instance. The first number is the resulting optimal cycle length for that case and the second number (in brackets) is the computation time used (in seconds on a Pentium III PC with 750-MHz CPU).

TABLE I
RESULTS FOR THE EXAMPLE WITH CHANGED PARAMETERS

w_l	w_r	1 hoist	2 hoists	3 hoists	4 hoists	5 hoists
0	20	2775	1227.5	802.5	802.5	805
		(0.001)	(0.013)	(0.025)	(0.048)	(0.081)
0	21.5	2775	1227.5	757.5	683.75	556.25
		(0.001)	(0.014)	(0.024)	(0.042)	(0.056)
0	∞	2775	1227.5	757.5	683.75	556.25
		(0.001)	(0.014)	(0.025)	(0.041)	(0.056)
-1.5	21.5	2775	1227.5	757.5	547.5	547.5
		(0.001)	(0.013)	(0.023)	(0.027)	(0.047)
$-\infty$	∞	2775	1227.5	757.5	547.5	547.5
		(0.001)	(0.013)	(0.023)	(0.027)	(0.047)

From the results, we can see that, in general, the optimal cycle length becomes shorter (e.g., the productivity of the electroplating line becomes higher), when the number of hoists increases. This is because more hoists mean more resource capacity while the material handling workload in a cycle is fixed. However, when the system has adequate hoists (3 or 4 for this example), adding more hoists will not reduce the optimal cycle length further. In fact, too many hoists may increase the optimal cycle length, as can be seen in the case of 5 hoists with a traveling range from 0 to 20, because of the increased interference among them.

The change of the hoist traveling range does not affect the optimal cycle length when the number of hoists is small. With a larger number of hoists, increasing the traveling range by a safety distance (1.5) reduces the optimal cycle length due to relaxed constraints. Furthermore, increasing the range at both ends of the track results in more reduction in the optimal cycle length compared to increasing at one end. The results also indicate that increasing the traveling range by one safety distance seems sufficient and a further increase does not affect the optimal cycle length for this example.

The computation time required increases as the number of hoists increases due to the increasing number of hoist assignment options. The change in the hoist traveling range has little impact on the computation time when the number of hoists is not large (up to 3 in this example). With a larger number of hoists, the computation time decreases when the range increases. In any of these cases, however, the computation time never exceeds 0.1 s, confirming that our algorithm is very efficient.

VII. CONCLUSION

In this paper, we have studied the multihoist no-wait cyclic scheduling problem in which the tank processing times and the part move times are fixed parameters, and developed a polynomial algorithm to obtain an optimal schedule with minimum cycle length. The polynomial solution is based on the following two developments: 1) For the problem with a fixed cycle length, we analyzed the relationship between the part moves and identified a group of hoist assignment constraints. The problem with the fixed cycle length was then transformed to a system of difference constraints on the hoist assignment

variables. The constraint system can then be solved efficiently. 2) a set of special values of the cycle length that might cause feasibility change was derived. The whole problem was then solved by checking the feasibility of these special values in ascending order. The computational complexity of the complete algorithm is $O(\mathrm{mn}^6)$. An example was given demonstrating that the algorithm is very efficient.

APPENDIX A
CONSTRAINTS FOR THE RESTRICTIONS BETWEEN m_j IN ONE CYCLE AND m_i IN THE NEXT CYCLE (DENOTED AS m_i')

Case a) Between m_j and the starting point of m_i', if $t_j^e \le T + t_i^s$

$$\min\left\{1, \left\lceil \frac{\left(w_{s_i} - w_{s_{j+1}} - \left(t_i^s + T - t_j^e\right)\lambda\right)}{d} \right\rceil\right\} \le x_i - x_j \le \max\left\{-1, \left\lfloor \frac{\left(w_{s_i} + \left(t_i^s + T - t_j^e\right)\lambda - w_{s_{j+1}}\right)}{d} \right\rfloor\right\}.$$

Case b) Between m_i' and the corner point $\left(t_j^s + \mu, w_{s_j}\right)$ of m_j, if $t_i^s + T \le t_j^s + \mu \le T + t_i^e$, as shown in the first equation at the bottom of the page.

Case c) Between m_i' and the corner point $\left(t_j^e - \mu, w_{s_{j+1}}\right)$ of m_j, if $T + t_i^s \le t_j^e - \mu \le T + t_i^e$, as shown in the second equation at the bottom of the page.

Case d) Between m_i' and the ending point of m_j, if $T + t_i^s \le t_j^e \le T + t_i^e$, as shown in the third equation at the bottom of the page.

Case e) Between m_j and the starting point of m_i', if $T + t_i^s \le t_j^e$, as shown in the fourth equation at the bottom of the page.

Case f) Between m_j and the corner point $(t_i^s + \mu + T, w_{s_i})$ of m_i', if $T + t_i^s + \mu \le t_j^s$, as shown in the first equation at the bottom of the next page.

Case g) Between m_j and the corner point $\left(t_i^e - \mu + T, w_{s_{i+1}}\right)$ of m_i', if $T + t_i^e - \mu \le t_j^e$, as shown in the second equation at the bottom of the next page.

$$\begin{cases} \left\lceil \frac{\left(f_i\left(t_j^s+\mu-T\right)-w_{s_j}\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } f_i\left(t_j^s + \mu - T\right) - w_{s_j} \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(f_i\left(t_j^s+\mu-T\right)-w_{s_j}\right)}{d} \right\rfloor, & \text{if } f_i\left(t_j^s + \mu - T\right) - w_{s_j} \ge d \\ \text{infeasible}, & \text{if } \left|f_i\left(t_j^s + \mu - T\right) - w_{s_j}\right| < d. \end{cases}$$

$$\begin{cases} \left\lceil \frac{\left(f_i\left(t_j^e-\mu-T\right)-w_{s_{j+1}}\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } f_i\left(t_j^e - \mu - T\right) - w_{s_{j+1}} \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(f_i\left(t_j^e-\mu-T\right)-w_{s_{j+1}}\right)}{d} \right\rfloor, & \text{if } f_i\left(t_j^e - \mu - T\right) - w_{s_{j+1}} \ge d \\ \text{infeasible}, & \text{if } \left|f_i\left(t_j^e - \mu - T\right) - w_{s_{j+1}}\right| < d. \end{cases}$$

$$\begin{cases} \left\lceil \frac{\left(f_i\left(t_j^e-T\right)-w_{s_{j+1}}\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } f_i\left(t_j^e - T\right) - w_{s_{j+1}} \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(f_i\left(t_j^e-T\right)-w_{s_{j+1}}\right)}{d} \right\rfloor, & \text{if } f_i\left(t_j^e - T\right) - w_{s_{j+1}} \ge d \\ \text{infeasible}, & \text{if } \left|f_i\left(t_j^e - T\right) - w_{s_{j+1}}\right| < d. \end{cases}$$

$$\begin{cases} \left\lceil \frac{\left(w_{s_i}-f_j\left(t_i^s+T\right)\right)}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } w_{s_i} - f_j\left(t_i^s + T\right) \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{\left(w_{s_i}-f_j\left(t_i^s+T\right)\right)}{d} \right\rfloor, & \text{if } w_{s_i} - f_j\left(t_i^s + T\right) \ge d \\ \text{infeasible}, & \text{if } \left|w_{s_i} - f_j\left(t_i^s + T\right)\right| < d. \end{cases}$$

Case h) Between m_j and the ending point of m_i', if $T+t_i^e \le t_j^e$ as shown in the third equation at the bottom of the page.

APPENDIX B
SPECIAL VALUES OF THE CYCLE LENGTH THAT MAY CAUSE CHANGES OF L_{ij} AND U_{ij}

1) For Case 1)

$$\frac{\left[z_j^s - z_i^e - \frac{(w_{s_{i+1}} - w_{s_j} + pd)}{\lambda}\right]}{k}, \text{ if } w_{s_j} \le w_{s_{i+1}} + pd \text{ where } p = 0, \ldots, m-1$$

$$\frac{\left[z_j^s - z_i^e - \frac{(w_{s_j} + pd - w_{s_{i+1}})}{\lambda}\right]}{k}, \text{ if } w_{s_j} \ge w_{s_{i+1}} - qd \text{ where } p = 0. \ldots, m-1.$$

2) For Case a)

$$\frac{\left[z_j^e - z_i^s + \frac{(w_{s_{j+1}} - w_{s_i} + pd)}{\lambda}\right]}{k}, \text{ if } w_{s_{j+1}} \ge w_{s_i} - pd \text{ where } p = 0, \ldots, m-1$$

$$\frac{\left[z_j^e - z_i^s + \frac{(w_{s_i} + pd - w_{s_{j+1}})}{\lambda}\right]}{k}, \text{ if } w_{s_{j+1}} \le w_{s_i} + pd \text{ where } p = 0, \ldots, m-1.$$

3) For Case 2)

$$\frac{\left[z_j^s - z_i^s - \mu - \frac{(w_{s_j} - w_{s_i} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_i} + pd \le w_{s_j} \le w_{s_{i+1}} + pd$$

$$\frac{\left[z_j^s - z_i^s - \mu + \frac{(w_{s_j} - w_{s_i} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_{i+1}} + pd \le w_{s_j} \le w_{s_i} + pd$$

where $p = -m+1, -m+2, \ldots, -1, 1, 2, \ldots, m-1$.

4) For Cases 3) and b)

$$\frac{\left[z_j^s - z_i^s - \frac{(w_{s_j} - w_{s_i} - pd)}{\nu}\right]}{k}, \text{if } w_{s_i} + pd \le w_{s_j} \le w_{s_{i+1}} + pd$$

$$\frac{\left[z_j^s - z_i^s + \frac{(w_{s_j} - w_{s_i} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_{i+1}} + pd \le w_{s_j} \le w_{s_i} + pd$$

where $p = -m+1, -m+2, \ldots, -1, 1, 2, \ldots, m-1$.

5) For Cases 4) and c)

$$\frac{\left[z_j^e - z_i^s - 2\mu - \frac{(w_{s_{j+1}} - w_{s_i} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_i} + pd$$

$$\begin{cases} \left\lceil \frac{(w_{s_i} - f_j(t_i^s + \mu + T))}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } w_{s_i} - f_j(t_i^s + \mu + T) \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{(w_{s_i} - f_j(t_i^s + \mu + T))}{d} \right\rfloor, & \text{if } w_{s_i} - f_j(t_i^s + \mu + T) \ge d \\ \text{infeasible}, & \text{if } \left|w_{s_i} - f_j(t_i^s + \mu + T)\right| < d. \end{cases}$$

$$\begin{cases} \left\lceil \frac{(w_{s_{i+1}} - f_j(t_i^e - \mu + T))}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } w_{s_{i+1}} - f_j(t_i^e - \mu + T) \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{(w_{s_{i+1}} - f_j(t_i^e - \mu + T))}{d} \right\rfloor, & \text{if } w_{s_{i+1}} - f_j(t_i^e - \mu + T) \ge d \\ \text{infeasible}, & \text{if } \left|w_{s_{i+1}} - f_j(t_i^e - \mu + T)\right| < d. \end{cases}$$

$$\begin{cases} \left\lceil \frac{(w_{s_{i+1}} - f_j(t_i^e + T))}{d} \right\rceil \le x_i - x_j \le -1, & \text{if } w_{s_{i+1}} - f_j(t_i^e + T) \le -d \\ 1 \le x_i - x_j \le \left\lfloor \frac{(w_{s_{i+1}} - f_j(t_i^e + T))}{d} \right\rfloor, & \text{if } w_{s_{i+1}} - f_j(t_i^e + T) \ge d \\ \text{infeasible}, & \text{if } \left|w_{s_{i+1}} - f_j(t_i^e + T)\right| < d. \end{cases}$$

$$\leq w_{s_{j+1}} \leq w_{s_{i+1}} + pd$$
$$\frac{\left[z_j^e - z_i^s - 2\mu + \frac{(w_{s_{j+1}} - w_{s_i} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_{i+1}} + pd$$
$$\leq w_{s_{j+1}} \leq w_{s_i} + pd$$
where $p = -m+1, -m+2, \ldots, -1, 1, 2, \ldots, m-1$.

6) For Cases 5) and d)

$$\frac{\left[z_j^e - z_i^s - \mu - \frac{(w_{s_{j+1}} - w_{s_i} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_i} + pd$$
$$\leq w_{s_{j+1}} \leq w_{s_{i+1}} + pd$$
$$\frac{\left[z_j^e - z_i^s - \mu + \frac{(w_{s_{j+1}} - w_{s_i} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_{i+1}} + pd$$
$$\leq w_{s_{j+1}} \leq w_{s_i} + pd$$
where $p = -m+1, -m+2, \ldots, -1, 1, 2, \ldots, m-1$.

7) For Case e)

$$\frac{\left[z_j^s - z_i^s + \mu + \frac{(w_{s_i} - w_{s_j} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_j} + pd$$
$$\leq w_{s_i} \leq w_{s_{j+1}} + pd$$
$$\frac{\left[z_j^s - z_i^s + \mu - \frac{(w_{s_i} - w_{s_j} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_{j+1}} + pd$$
$$\leq w_{s_i} \leq w_{s_j} + pd$$
where $p = -m+1, -m+2, \ldots, -1, 1, 2, \ldots, m-1$.

8) For Cases 6) and f)

$$\frac{\left[z_j^s - z_i^s + \frac{(w_{s_i} - w_{s_j} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_j} + pd$$
$$\leq w_{s_i} \leq w_{s_{j+1}} + pd$$
$$\frac{\left[z_j^s - z_i^s - \frac{(w_{s_i} - w_{s_j} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_{j+1}} + pd$$
$$\leq w_{s_i} \leq w_{s_j} + pd$$
where $p = -m+1, -m+2, \ldots, -1, 1, 2, \ldots, m-1$.

9) For Cases 7) and g)

$$\frac{\left[z_j^s - z_i^e + 2\mu + \frac{(w_{s_{i+1}} - w_{s_j} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_j} + pd$$
$$\leq w_{s_{i+1}} \leq w_{s_{j+1}} + pd$$
$$\frac{\left[z_j^s - z_i^e + 2\mu - \frac{(w_{s_{i+1}} - w_{s_j} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_{j+1}} + pd \leq w_{s_{i+1}}$$
$$\leq w_{s_j} + pd$$
where $p = -m+1, -m+2, \ldots, -1, 1, 2, \ldots, m-1$.

10) For Cases 8) and h)

$$\frac{\left[z_j^s - z_i^e + \mu + \frac{(w_{s_{i+1}} - w_{s_j} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_j} + pd \leq w_{s_{i+1}}$$
$$\leq w_{s_{j+1}} + pd$$
$$\frac{\left[z_j^s - z_i^e + \mu - \frac{(w_{s_{i+1}} - w_{s_j} - pd)}{\nu}\right]}{k}, \text{ if } w_{s_{j+1}} + pd$$
$$\leq w_{s_{i+1}} \leq w_{s_j} + pd$$
where $p = -m+1, -m+2, \ldots, -1, 1, 2, \ldots, m-1$.

REFERENCES

[1] L. Lei and T. J. Wang, "A proof: The cyclic hoist scheduling problem is NP-complete," in *Working Paper 89-0016*. Piscataway, NJ: Rutgers Univ., 1989.

[2] L. W. Phillips and P. S. Unger, "Mathematical programming solution of a hoist scheduling program," *AIIE Trans.*, vol. 28, pp. 219–225, 1976.

[3] W. Song, Z. B. Zabinsky, and R. L. Storch, "An algorithm for scheduling a chemical processing tank line," *Prod. Planning Control*, vol. 4, pp. 323–332, 1993.

[4] J. Liu, Y. Jiang, and Z. Zhou, "Cyclic scheduling of a single hoist in extended electroplating lines: A comprehensive integer programming solution," *IIE Trans.*, vol. 34, pp. 905–914, 2002.

[5] R. Armstrong, L. Lei, and S. Gu, "A bounding scheme for deriving the minimal cycle time of a single-transporter n-stage process with time-window constraints," *Eur. J. Oper. Res.*, vol. 78, pp. 130–140, 1994.

[6] L. Lei and T. J. Wang, "Determining optimal cyclic hoist schedules in a single-hoist electroplating line," *IIE Trans.*, vol. 26, pp. 25–33, 1994.

[7] W. C. Ng, "A branch and bound algorithm for hoist scheduling of a circuit board production line," *Int. J. Flexible Manuf. Syst.*, vol. 8, pp. 45–65, 1996.

[8] H. Chen, C. Chu, and J. M. Proth, "Cyclic scheduling of a hoist with time window constraints," *IEEE Trans. Robot. Autom.*, vol. 14, no. 1, pp. 144–152, Feb. 1998.

[9] L. Lei, "Determining the optimal starting time in a cyclic schedule with a given route," *Comput. Oper. Res.*, vol. 20, pp. 807–816, 1993.

[10] V. Kats and E. Levner, "A strongly polynomial algorithm for no-wait cyclic robotic flowshop scheduling," *Oper. Res. Lett.*, vol. 21, pp. 171–179, 1997.

[11] Y. Crama, V. Kats, V. Van de Klundert, and E. Levner, "Cyclic scheduling in robotic flowshops," *Ann. Oper. Res.*, vol. 96, pp. 97–124, 2000.

[12] P. Serafini and W. Ukowich, "A mathematical model for periodic scheduling problems," *SIAM J. Discrete Math.*, vol. 2, pp. 550–581, 1989.

[13] L. Lei and T. J. Wang, "The minimum common-cycle algorithm for cyclic scheduling of two material handling hoists with time window constraints," *Manage. Sci.*, vol. 37, pp. 1629–1639, 1991.

[14] A. Che and C. Chu, "Multi-degree cyclic scheduling of two robots in a no-wait flowshop," *IEEE Trans. Autom. Sci. Eng.*, vol. 2, no. 2, pp. 173–183, Apr. 2005.

[15] L. Lei, R. Armstrong, and S. Gu, "Minimizing the fleet size with dependent time-window and single-track constraints," *Oper. Res. Lett.*, vol. 14, pp. 91–98, 1993.

[16] R. Armstrong, S. Gu, and L. Lei, "A greedy algorithm to determine the number of transporters in a cyclic electroplating process," *IIE Trans.*, vol. 28, pp. 347–355, 1996.

[17] J. B. Orlin, "Minimizing the number of vehicles to meet a fixed periodic schedule," *Oper. Res.*, vol. 30, pp. 760–776, 1982.

[18] V. Kats and E. Levner, "Minimizing the number of robots to meet a given cyclic schedule," *Ann. Oper. Res.*, vol. 69, pp. 209–226, 1997.

[19] J. Liu and Y. Jiang, "An efficient optimal solution to the two-hoist no-wait cyclic scheduling problem," *Oper. Res.*, vol. 53, pp. 313–327, 2005.

[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.

[21] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.

[22] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[23] V. Kats and E. Levner, "Polynomial algorithms for scheduling of robots," in *Intelligent Scheduling of Robots and FMS*, E. Levner, Ed. Holon, Israel: CTEH, 1996, pp. 77–100.

Yun Jiang received the Ph.D. degree in industrial engineering from Hong Kong University of Science and Technology (HKUST), Hong Kong, China, in 2003 and the B.S. degree in automatic control from Hua Zhong University of Science and Technology, Wuhan, China, in 1998.

He was a Research Associate and Research Fellow with HKUST and the National University of Singapore (NUS), respectively. Currently, he is an Assistant Professor with Bilkent University, Ankara, Turkey. His research interests include scheduling and planning in production and logistics.

Jiyin Liu received the B.Eng. degree in industrial automation and the M.Eng. degree in systems engineering from Northeastern University, Shenyang, China, in 1982 and 1985, respectively, and the Ph.D. degree in manufacturing engineering and operations management from the University of Nottingham, Nottingham, U.K., in 1993.

Currently he is Professor of Operations Management in the Business School at Loughborough University, Leicestershire, U.K. Prior to joining Loughborough University, he was with Northeastern University and Hong Kong University of Science and Technology, Hong Kong, China. His research interests are in operations planning, scheduling, and supply chain and logistics management. He has published papers in journals such as the *European Journal of Operational Research*, *IIE Transactions*, *International Journal of Production Research*, *Journal of the Operational Research Society*, *Naval Research Logistics*, *Operations Research*, and *Transportations Research*.