Journal of the Korean
OR/ MS Society
Vol. 18, No. 2, August 1993

An Approach to Double Hoist Scheduling in the Chemical Processes

Joon-Mook Lim* and Hark Hwang*

Abstract

This paper deals with scheduling problem of the chemical process system where aircraft parts go through a given sequence of tanks filled with chemical solutions. The system has two hoists which move carriers holding the parts between tanks. A mixed integer programming model is developed from which a maximum throughput schedule can be found for the hoist movements. To show the validity of the model, a real world problem is solved and the results are compared with those with an existing approach.

1. Introduction

During their manufacture, certain aircraft parts go through a given sequence of chemical treatments. Treatments are applied by immersing the parts sequentially into a series of tanks filled with chemical solutions. The parts are held in carriers which are moved by one or more hoists. Hoists are programmed to handle the inter-tank moves of the parts, where each ***move*** consists of three simple hoist operations : (1) lift a carrier from a tank ; (2) move to the next tank ; and(3) submerge the parts in that tank. Upon completion of a move, a hoist travels to another tank for the next scheduled move. Both the hoist traveling times and times to perform each move are given constants. The parts must proceed from tank 0(loading tank) to tank 1, to tank 2, ⋯, to tank n and finally to tank $n+1$ (unloading tank). There is a minimum amount of time the parts have to remain within each tank and, in most cases there is also a maximum allowed time. Many settings of such lines also require the hoists to travel along a common track, where avoidance of hoist collisions must be considered. Figure 1 shows an arrangement of a double hoist system.

* Department of Industrial Engineering, Korea Advanced Institute of Science and Technology.

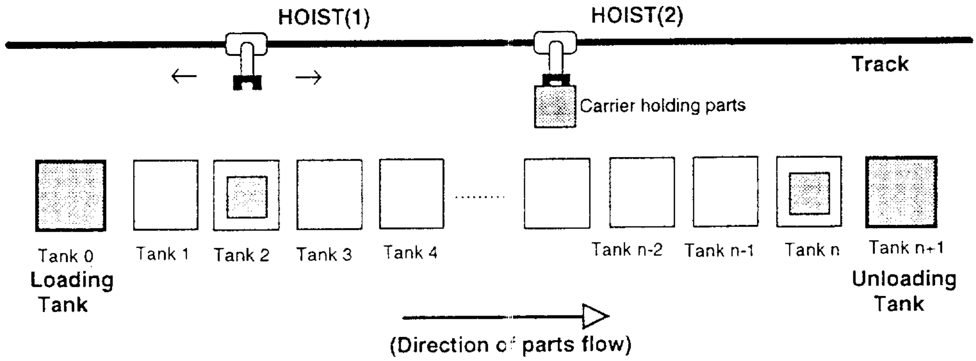

Figure 1. Hoist setup

The amount of time between successive loadings of carriers into the system (departures from tank 0)will be taken to be a *cycle*. Each cycle must be identical, which means the configuration of carriers in tanks at the end of a cycle must be the same as that of at the beginning. The configuration includes (a) the location of the hoists, (b) the number and locations of the carriers in the system, and (c) the elapsed processing time of each carrier in its current process. It follows that during each cycle, each tank has one carrier dropped in, and one removed, not necessarily in that order. That is, at the beginning of a cycle, a tank may have a carrier in it. During the cycle, that carrier is removed, and some time later, another carrier is dropped into the tank. For the case when a tank is empty at the beginning of the cycle, a carrier will eventually be enterd into the tank, and removed later in the cycle.

The fixed sequence of moves that hoists perform in each cycle is defined by a *cyclic schedule*. Cycles may be distinguished by the number of carriers which are introduced into the system in a period. In an *n−cycle*, n($n \geq 1$) carriers are introduced each period. In this paper, we limit our study to 1−cycle schedule(simple cyclic schedule) following Phillips and Unger[4] and Shapiro and Nuttle[5].

The number of carriers which can be served simultaneously by the hoists depends on the relative magnitudes of the minimum and maximum allowed time in each tank and the hoist travel times. In the hoist scheduling problem, the objective is to maximize the throughput of carriers per hour which is equivalent to minimizing the cycle length.

A few studies on the cyclic scheduling problem with a single hoist have been reported in the literature. Phillips and Unger[4] developed a mixed integer programming model to determine the minimal cyclic time. Shapiro and Nuttle[5] proposed a branch and bound procedure based on linear programming. Lei and Wang[1] introduced an interval−cutting algorithm that is able to find the optimal cyclic schedule of Q−degree.

For the cyclic scheduling problem with two or more hoists, Thesen and Lei[6] asserted that different decision rules should be used to control hoist movements responding to various situations. However, the approach has the disadvantage in that decision rules for all possible situations must be developed in advance. Recently, Lei and Wang[2] proposed a Minimum Common Cycle(MCC) algorithm. According to MCC algotithm, a given double hoist system is partitioned into two single hoist systems of ***contiguous*** tanks i.e., single hoist system with the first $k(<n+1)$ tanks and another with remaining tanks. The two systems are solved independently with an existing solution procedure to an optimal solution and then (minimal) common-cycle time that is acceptable to both subsystems is determined through an iterative process. It can be recognized that MCC algorithm contains the following shortcomings : (1) the movements of each hoist are confined to each corresponding set of tanks in which two neighboring tanks of a tank must belong to the same set of tanks except the right most tank. (2) the effects on the throughput of the second hoist position at the beginning of cycle is not considered.

In this regard, we are going to find a double hoist system schedule which relaxes the restriction on the hoist movement and also treats the second hoist position as a decision variable. In the following, a Mixed Integer Programming formulation is described for the double hoist system. Computational experiences with data from a real world system are presented to verify the formulation developed.

2. Mathematical Programming Formulation

In this section, a mathematical programming model is developed for finding an optimal cycle for the double hoidt system as shown in Figure 1. The system is operated by two hoists, HOIST(1) and HOIST(2). Unlike single hoist system, the type of hoist must be determined for each tank in addition to the time at which a carrier is removed from the tank. We assume that two subcycles, $cycle^1$ and $cycle^2$, can be generated by each hoist's movement and the length of each subcycle is the same. Note that these subcycles are imbedded in the system cycle whose length is the time interval between two successive departure times of carriers into tank 0.

For the simplicity of modeling, the following notations and assumptions are introduced.

Notation

(a) ST : set of tank numbers, $ST=\{0,1,2,\cdots,n,n+1\}$.

(b) $h(i)$: $h(i)=h$ if HOIST(h) removes carrier from tank i.

(c) $G(h)$: set of tank numbers associated with $h(i)$, and $G(h)=\{i\mid h(i)=h,\ i\in ST\}$.

(d) T_i^h : time at which HOIST(h) removes a carrier from tank i.

With i given, either T_i^1 or T_i^2 is defined but not both.

(e) T^h_{max} : $T^h_{max} = \underset{i \in G(h)}{Max} T^h_i$ for $h=1,2$.

(f) X_{ij} : zero-one integer variable defined only for $i > j$, $i,j \in ST$.
$X_{ij}=1$ iff $T^{h(i)}_i > T^{h(j)}_j$ and 0, otherwise.

(g) Z^h_i : zero-one integer variable and $Z^h_i=1$ iff $T^h_i = T^h_{max}$.

(h) $Lcycle$: length of the system cycle.

(i) $Lcycle^h$: length of $cycle^h$.

(j) C^h_{ij} : travel time of the empty HOIST(h) from tank i to tank j.
$C^h_{i,i}=0$ and $C^h_{j,i}=C^h_{i,j}$.

(k) CM^h_{ij} : travel time of HOST(h) moving a carrier from tank i to tank j.

(l) A_i : minimum required processing time in tank i.

(m) B_i : maximum allowed processing time in tank i.

(n) M : very large real number.

Assumption

(1) Indentical parts are introduced into the system.

(2) Tanks are indexed by their order in the given process sequence.

(3) At no time can two carriers occupy the same tank.

(4) Two hoists cannot change their relative positions, i.e., HOIST(1) is always located in front of HOIST(2).

(5) Carriers are introduced into the system only by HOIST(1) and removed from the system only by HOIST(2). It is defined that $T^1_0=0$ and $T^2_{n+1}=\infty$.

(6) HOIST(2) is located at tank $K (2 \le K \le n)$ at the moment a system cycle begins.

A cycle is feasible if the associated sequence of hoist movements is executable and the associated processing time in each tank is within the allowed range. In other words, any schedule is feasible if and only if the following conditions are satisfied(Shapiro and Nuttle[5]).

(F1) No two carriers occupies the same process tank at the same time.

(F2) No two moves must be made simultaneously for individual hoist.

(F3) There is sufficient time between moves for each hoist to travel from where it was last used to where it is next needed.

(F4) The processing time in the tank i $(i=1,2,\cdots,n)$ must lie within $[A_i, B_i]$.

The constraints representing the above feasibility conditions and the *cycle* that has to be minimized can be described as follows.

Objective function

Minimize $Lcycle$ (1)

Constraints

(a) The constraints setting the cycle time.

(i) to define the subcycle length.

For each h,

$$Lcycle^h = T^h_{max} + \sum_{i \in G(h)} \{CM^h_{i,i+1} + C^h_{i+1,k}\} Z^h_i \tag{2}$$

$$\text{where } k = \begin{cases} 0 & \text{for } h=1 \\ K & \text{for } h=2. \end{cases}$$

Cycle length is the sum of the following three components : (i) T^h_{max} (the last time HOIST(h) departs from a tank with a carrier), (ii) the hoist travel time to the next tank to drop off the carrier, (iii) the travel time for the hoist to return to the position where it was at time 0.

(ii) to define T^h_{max}.

For each h,

$$T^h_{max} \geq T^h_i \tag{3}$$

$$T^h_{max} \leq T^h_i - (Z^h_i - 1)M \text{ for } i \in G(h) \tag{4}$$

$$\sum_{i \in G(h)} Z^h_i = 1 \tag{5}$$

The relationships in (ii) force T^h_{max} to be equal to the maximum T^h_i and force Z^h_i to be 1 for i which satisfies $T^h_i = T^h_{max}$ and Z^h_i to be 0 for all the other i.

(iii) to satisfy assumption(6)

$$T^h_i \geq C^2_{K,i} \text{ for } i \in G(2) \tag{6}$$

Due to the assumption that HOIST(2) starts at tank K, $2 \leq K \leq n$, at the beginning of a cycle, T^2_i equals or larger than $C^2_{K,i}$ which is the time for the empty hoist to move from tank K to tank i. Note that if $i=K$ then $C^2_{K,K}=0$ and (6) also holds.

(b) to provide hoists with sufficient travel time between the tanks of the same group.

For each h and $i, j \in G(h)$,

$$T_j^h - T_i^h \geq CM_{i,i+1}^h + C_{i+1,j}^h - M \cdot X_{i,j} \qquad i>j \tag{7}$$

$$T_i^h - T_j^h \geq CM_{j,j+1}^h + A_{j+1} + M(X_{i,j}-1) \qquad i=j+1 \tag{8}$$

$$T_i^h - T_j^h \geq CM_{j,j+1}^h + C_{j+1,i}^h + M(X_{i,j}-1) \qquad i>j+1 \tag{9}$$

Case 1. $T_i^h > T_j^h$ (A carrier is removed from tank j before a carrier is removed from tank i).
The left hand side(LHS) of (7) is negative, while the first two terms in the right hand side (RHS) of (7) are positive, therefore the last term on the RHS of (7) must be negative, implying $X_{i,j}=1$.

Case 1-1. $i=j+1$
Constraint (8) says that the time between removing a carrier from tank j and removing a carrier from tank $j+1$ must be at least time enough to take a carrier from tank j to tank $j+1$ ($CM_{j,j+1}^h$) plus the minimum time required in tank $j+1$(A_{j+1}).

Case 1-2. $i>j+1$
Constraint (9) says that the time between removing a carrier from tank j and removing a carrier from tank i must be at least large enough to take a carrier from tank j to tank $j+1$($CM_{j,j+1}^h$) plus the time for the empty HOIST(h) to move from tank $j+1$ to tank i($C_{j+1,i}^h$) to pick up another carrier.

Case 2. $T_j^h > T_i^h$ (A carrier is removed from tank i before a carrier is removed from tank j).
The LHS of (8) or (9) is negative, while the first two terms in the RHS of (8) or (9) are positive. Hence the last term of (8) or (9) must be negative, implying $X_{i,j}=0$. Constraint (7) then says that the time between removing a carrier from tank i and removing a carrier from tank j must be at least large enough to take a carrier from tank i to rank $i+1$ ($CM_{i,i+1}^h$) plus the time for the empty HOIST(h) to get from tank $i+1$ to tank j ($C_{i+1,j}^h$).

(c) to satisfy assumption (4).

Let $l \in G(1)$ and $m \in G(2)$.

For $m+1<l$

$$T_l^1 - T_m^2 \geq CM_{m,m+1}^2 + C_{m+1,l}^2 - M(1-X_{l,m}) \tag{10}$$

$$T^2_m - T^1_l \geq CM^1_{l,l+1} + C^1_{l+1,m} - M \cdot X_{l,m} \tag{11}$$

For $m+1=l$

$$T^1_l - T^2_m \geq CM^2_{m,m+1} + A_l - M(1-X_{l,m}) \tag{12}$$

$$T^2_m - T^1_l \geq CM^1_{l,l+1} + C^1_{l+1,m} - M \cdot X_{l,m} \tag{13}$$

For $m=l+1$

$$T^1_l - T^2_m \geq -M \cdot X_{m,l} \tag{14}$$

$$T^2_m - T^1_l \geq CM^1_{l,l+1} + A_{l+1} - M(1-X_{m,l}) \tag{15}$$

Case 1. (Tank m is located to the left side of tank l.)

Case 1-1. $T^1_l \geq T^2_m$

The LHS of (11) or (13) is negative ,while the first two terms on RHS of (11) and (13) are positive. This implies that $X_{l,m}=1$.

Cace 1-1-1. $m+1<l$

Constraint (10) says that the time between removing a carrier from tank m by HOIST(2) and removing a carrier from tank l by HOIST(1) must be at least large enough for HOIST(2) to to take a carrier from tank m to tank $m+1$ plus the time for empty HOIST(2) to move from tank $m+1$ to tank l because at the instant when HOIST(1) removes a carrier from tank l, HOIST(2) must be located to the right of HOIST(1).

Case 1-1-2. $m+1=l$

Constraint (12) says that the time between removing a carrier from tank m by HOIST(2) and removing a carrier from tank $l(=m+1)$ by HOIST(1) must be at least large enough for HOIST(2) to take a carrier from tank m to tank $m+1$ plus the minimum requried processing time A_l in tank l because the carrier which is placed into tank l by HOIST(2) at $T^2_m + CM^2_{m,m+1}$ will be removed from tank l by HOIST(1) during the same cycle.

Case 1-2. $T^1_l < T^2_m$

The LHS of (10) or (12) is negative, while the first two terms on RHS of (10) and (12) are positive. This implies $X_{l,m} = 0$.

Case 1-2-1. $m+1<l$

Constraint (11) now says that the time between removing a carrier from tank l by HOIST(1) and removing a carrier from tank m by HOIST(2) must be at least large enough for HOIST(1) to take a carrier from tank l to tank $l+1$ plus the time for empty HOIST(1) to move from tank $l+1$ to tank m because at the time, T_m^2, when HOIST(2) removes a carrier from tank m, HOIST(1) must be located to the left of HOIST(2).

Case 1-2-2. $m+1=l$

Constraint (13) has the same interpretation as case 1-2-1.

Case 2. (Tank m is located to the right side of tank l.)

Case 2-1. $T_l^1 \geq T_m^2$

Case 2-1-1. $m=l+1$

The LHS of (15) is negative, while the first two terms on RHS of (15) are positive. This implies $X_{m,l}=0$. Constraint (14) says that the time between removing a carrier from tank m by HOIST(2) and removing a carrier from tank l by HOIST(1) has no restriction.

Case 2-1-2. $m>l+1$ No restriction.

Case 2-2. $T_l^1 < T_m^2$

Case 2-2-1. $m=l+1$

The LHS of (14) is negative. This implies $X_{m,l}=1$. Constraint (15) says that the time between removing a carrier from tank l by HOIST(1) and removing a carrier from tank m by HOIST(2) must be at least large enough for HOIST(1) to take a carrier from tank l to tank $l+1$ plus the minimum time required in tank $l+1$ because the carrier which is placed into tank $l+1$ by HOIST(1) at $T_l^1+CM_{l,l+1}^1$ will be removed from tank $l+1$ by HOIST(2) during the same cycle.

Case 2-2-2. $m>l+1$

No retriction.

(d) to guarantee that carriers are kept in tank i for an amount of time lying between A_i, and B_i.

For each h and $i \in G(h)$,

$$T_i^h - (T_{i-1}^h + CM_{i-1,i}^h) \leq B_i + M(1 - X_{i,i-1}) \tag{16}$$

$$A_i - M \cdot X_{i,i-1} \leq (T_i^h + Lcycle^h) - (T_{i-1}^h + CM_{i-1,i}^h) \tag{17}$$

$$(T_i^h + Lcycle^h) - (T_{i-1}^h + CM_{i-1,i}^h) \leq B_i + M \cdot X_{i,i-1} \tag{18}$$

Case 1. (Thers is a carrier in tank i at time 0)

The carrier in tank i must be removed before any other carrier can be removed from tank $i-1$. This is so because once a carrier is removed from tank $i-1$ and taken to tank i, tank i must be empty. Therefore $T_i^h < T_{i-1}^h$, and by the type(b) and(c) constraints, $X_{i,i-1} = 0$. Constraint(16) is now satisfied automatically. Constraints (17) and (18) now force $(T_i^h + Lcycle^h) - (T_{i-1}^h + CM_{i-1i}^h)$ to lie between A_i and B_i. But $(T_i^h + Lcycle^h)$ is just the time(during the following cycle) when a carrier is removed from tank i, and this carrier is placed into tank i at time $(T_{i-1}^h + CM_{i-1i}^h)$(during the present cycle). It is clear then that what is being constrained to lie between A_i and B_i is the time a carrier remains in tank i.

Case 2. (There is no carrier in tank i at time 0).

In this case, a carrier(not necessarily in tank $i-1$ at time 0) must be removed from tank $i-1$ and placed into tank i before it can be removed from tank i. That is, $T_i^h > T_{i-1}^h$, which implies (by the type (b) and (c) constraints) that $X_{i,i-1}=1$. (17) and (18) are then automatically satisfied. Constraint (16) then says that the time a carrier spends in tank i [the time it leaves, T_i^h, less the time it arrives, $(T_{i-1}^h + CM_{i-1i}^h)$] must be no greater than B_i. The fact that the time a carrier spends in tank i must be at least A_i is covered in constraints (8), (12) and (15).

Suppose $|G(1)|=n_1$ and $|G(2)|=n_2=n-n_1$, excluding the loading and unloading tanks. Then the model involves $n+4$ continuous variables, i.e., the n T_i^h's, 2 T_{max}^h and 2 $Lcycle^h$, and at most $\frac{n(n+1)}{2}$ zero one integer variables, that is, the n Z_i^h's and the α number of X_{ij}'s where α depends on n_1, n_2 and the type (c) constraint. There are $(2n+n_2+4)$ constraints of type(a), $(n_1(n_1-1)+n_2(n_2-1))$ constraints of type(b), β number of constraints of type (c) where β depends on $G(i)$ and $3n$ constraints of type(d), with the total being not more than (n^2+4n+n_2+4).

Now, we have the following double hoist scheduling model.

Minimize $\quad Lcycle$

Subject to

$Lcycle^h = Lcycle \quad$ for $h=1,2$

Constraint(2) through (18)

$T_i^h,\ T_{max}^h \geq 0$ for each h and $i \in G(h)$

$X_{ij},\ Z_i^h \in \{0,\ 1\}$ for each h and $i, j \in G(h)$

Once $G(1)$ and $G(2)$ known, the above formulation, constructed by zero one integer and non-negative real variables, can be transformed into Mixed Integer Programming(MIP).

We can schedule the double hoist system by interpreting the solution, T_i^h's and $X_{i,j}$'s of the MIP as the following: At time T_i^h, HOIST(h) raises the carrier in tank i and moves the carrier to tank $i+1$ and lowers it into the tank at time $T_i^h - CM_{i,i-1}^h$. If T_i^h is less than or equal to T_{i+1}^h, then $X_{i+1,i}$ is forced to be 1. This means that the carrier, in the tank i, has to be removed by HOIST(h) before(or at least at the same time) the carrier in tank $i+1$ is removed. Therefore tank $i+1$ does not have a carrier at time 0 (initial state). On the contrary, if T_i^h is greater than T_{i-1}^h, then $X_{i+1,i}$ is equal to zero. This implies that tank $i+1$ has a carrier at the initial state because the removing time of a carrier in tank $i+1$ is less than the entering time of its carrier.

In each tank, the actual processing time is determined by the time difference between the time when a carrier is removed from the tank i and the time for a carrier to be entered. Thus, with $T_i^h \leq T_{i+1}^h$, the ***actual processing time*** becomes $T_{i+1}^h - (T_i^h + CM_{i,i-1}^h)$ and with $T_i^h > T_{i+1}^h$, the processing time becomes $Lcycle + T_{i+1}^h - (T_i^h + CM_{i,i+1}^h)$.

HOIST(1) starts to lift a new carrier from tank 0 (loading tank) at time zero. After performing a sequence of operation given by T_i^1's $i \in G(1)$, it terminates the works of one cycle and returns to tank 0. Likewise, HOIST(2) moves from tank K at time zero, and then returns to the same tank at the end of the cycle.

3. Computational Experiences

We have applied the proposed formulation to a set of data from an actual hoist set up at an aircraft parts manufacturing company(we are requested not to reveal the company's name). This setup has 7 tanks with maximum and minimum processing times listed in table 1 in seconds. Table 2 and 3 show the empty hoist travel times and the carrier moving times in seconds between tanks, respectively.

The example was formulated by MIP and solved by LINDO on Personal Computer(486 DX2). The results are summarized in Table 4 and 5. Note that according to MCC algorithm the minimum cycle time becomes 1158 seconds while the proposed model generates the cycle time of 932 seconds, resulting in 20% of improvement. Suppose the system is operated by a single hoist, then the minimum cycle time is found to be 1414 seconds. Figure 2 illustrates the movements schedule of each hoist on time horizon. It can be read as the following : At the beginning of cycle, HOIST

(1) moves a carrier to tank 1 with the allowed time of 73 seconds and then goes to tank 5 to pick up a carrier. At tank 5 the hoist lifts a carrier, moves it to tank 6 and lowers it into the tank with the allowed time of 81 seconds and so on. Figure 3 shows the locations of carriers, the status of tanks and the movements of hoists at the time when placement of a carrier into some tank is completed.

For the problem with $|G(1)|=n_1$ and $|G(2)|=n_2=n-n_1$, the formulation presented involves $\frac{1}{2}(n^2+5n+12)$ number of decision variables and (n^2+4n+n_2+4) constraints. This implies that the problem size becomes larger quite rapidly as n increases. For a moderate size problem, an optimal solution can be found within a reasonable computing time. Note that in Table 4, approximately 40 seconds of CPU time was required to solve each problem with $n=7$. We found that our approach becomes computationally infeasible on personal computer if n, the number of tanks, is much more than 10.

Table 1. A_i and B_i in seconds

i	0	1	2	3	4	5	6	7	8
A_i	0	240	720	180	540	420	180	600	0
B_i	∞	1500	780	300	600	960	240	720	∞

Table 2. Empty hoist moving time($C^1_{i,j}=C^2_{i,j}$)

$i \setminus j$	0	1	2	3	4	5	6	7	8
0	0	45	53	58	63	83	98	111	137
1	45	0	46	51	56	76	91	104	111
2	53	46	0	43	48	68	83	96	123
3	58	51	43	0	43	63	78	91	118
4	63	56	48	43	0	58	73	86	113
5	83	76	68	63	58	0	53	66	93
6	98	91	83	78	73	53	0	51	78
7	111	104	96	91	86	66	51	0	65
8	137	111	123	118	113	93	78	65	0

Table 3. Carrier moving time of hoist($CM^1_{i,i+1}=CM^2_{i,i+1}$)

i	0	1	2	3	4	5	6	7
$CM_{i,i+1}$	73	74	71	71	86	81	79	93

Table 4. Result summary for given example data

	Model	Cycle time (sec)	K	G(1)	G(2)	CPU time (sec)
Single Hoist System		1414		{0, 1,2,3,4,5,6,7,8}	ϕ	272.0
Double Hoist System	MCC algorithm	1332	2	{0, 1}	{2, 3, 4, 5, 6, 7, 8}	32.6
		1158	3	{0, 1, 2}	{3, 4, 5, 6, 7, 8}	39.7
		1158	4	"	"	36.4
		1158	5	"	"	40.5
		1158	6	"	"	41.7
		1158	7	"	"	32.0
		1172	4	{0, 1, 2, 3}	{4, 5, 6, 7, 8}	29.5
		1172	5	{0, 1, 2, 3, 4}	{5, 6, 7, 8}	32.9
		1172	6	{0, 1, 2, 3, 4 5}	{6, 7, 8}	44.0
		1236	7	{0, 1, 2, 3, 4, 5, 6}	{7, 8}	54.0
	Proposed formulation	932	2	{0, 1, 2, 5}	{3, 4, 5, 6, 7, 8}	37.5
		932	3	"	"	34.5
		932	4	"	"	38.0
		932	5	"	"	43.9
		947	6	"	"	29.8
		932	7	"	"	52.0

Table 5. Results for example data
(for G(1)={0, 1, 2, 5}, G(2)={3, 4, 6, 7, 8} and K=4, Cycle time=932)

tank	0	1	2	3	4	5	6	7	8
entering time	*	73	879	754	188	874	443	702	463
removing time	0	805	683	117	788	362	623	370	*
elapsed time	*	732	736	295	600	420	180	600	*

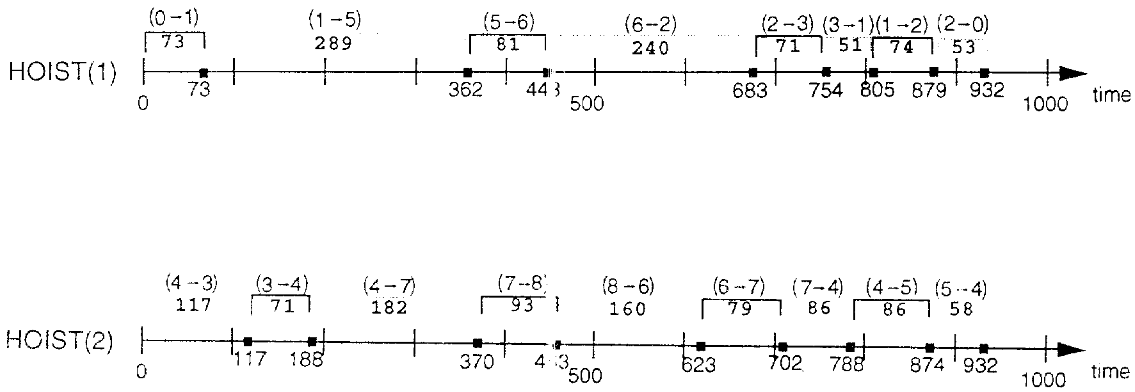

Figure 2. The cyclic schedule for double hoist system

$(i \rightarrow j)$	: empty or loaded hoist movement from tank i to j
$\boxed{M}$	: loaded hoist movement
M	: empty hoist movement
M	: time allowed

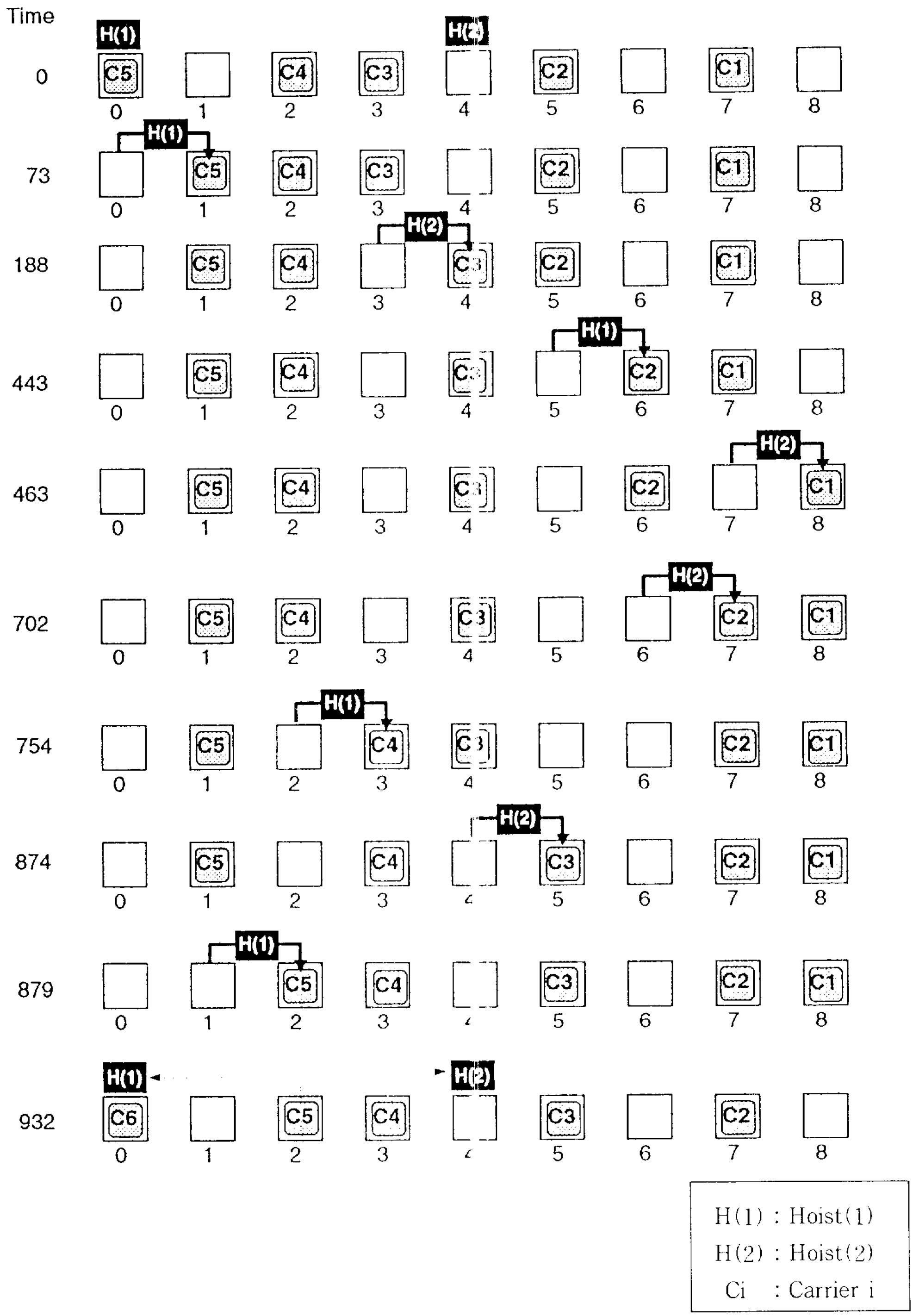

Figure 3. Configurations of tanks and carriers

4. Conclusion

In double hoist system, the requirement of avoiding hoist interference causes much complexities in formulating an optimum hoists schedule compared to single hoist system. In an effort to eliminate some shortcomings of an existing mathematical programming approach, this paper presented another formulation for the double hoist schedule problem. Even though the formulation was shown to generate a better schedule in terms of the system throughput with a case problem, we observe that for the problem with $n>10$, the formulation becomes computationally infeasible on personal computer. The problems remained for further studies include finding an efficient way to partition tanks into two disjoint subsets of tanks and determination of K.

References

[1] Lei, L. and T. J. Wang, "On the Optimal Cyclic Schedules of Single Hoist Electroplating Processes," Working Paper #89-0006, Rutgers University, April, 1989.

[2] Lei, L. and T. J. Wang, "The Minimum Common-Cyclic Algorithm for Cyclic Scheduling of Two Material Handling Hoists with Time Window Constraints," *Management Science*, Vol. 37, No. 12(1991), pp.1629-1639.

[3] Murty, K. G., *Linear programming*, John Wiley and Son, New York, 1983.

[4] Phillips, L. W. and P. S. Unger, "Mathematical Programming Solution of a Hoist Scheduling Program," *AIIE Transactions*, Vol. 8, No. 2(1976), pp.219-225.

[5] Shapiro, G. W. and H. L. Nuttle, "Hoist Scheduling for a PCB Electroplating Facility," *IIE Transactions*, Vol. 20, No. 2(1988), pp.157-16[illegible].

[6] Thesen, A. and L. Lei, "An Expert Scheduling System for Material Handling Hoists," *Journal of Manufacturing System*, Vol. 9, No. 3(1990), pp.247-252.