مؤتمر الأزهر الهندسي الدولي الحادي عشر

AL-AZHAR ENGINEERING
ELEVENTH INTERNATIONAL CONFERENCE
December 21 - 23, 2010

Code: E 51

A HEURISTIC ALGORITHM FOR HOIST SCHEDULING IN ELECTROPLATING LINES

Tamer A.Ismail , Amin K.El Kharbotly and Nahid H.Afia
Faculty of Engineering, Ain Shams University

ABSTRACT
Hoist scheduling has been considered a very important factor in improving the productivity of production lines which uses a transporter for handling jobs between successive processing stations.

In this paper a heuristic algorithm for scheduling of hoist moves in a no-wait system is developed, where the processing time of the jobs in the processing stations is prescribed between a lower and upper bound known as time window. The system can process multiple job types in which each type differs in its processing sequence as well as the immersion time in the different processing stages.

The considered system may contain multi tank processing stages where a number of duplicate tanks can perform the same process, as well as multi function tanks where a single tank can be visited by a job more than once.

The algorithm is developed and tested on three bench mark problems of a single product type, and one problem with two product types. The results showed that the proposed algorithm gives comparable results with those obtained by exact mathematical methods.

Keywords: Hoist scheduling, No wait system, Time window, Electroplating

1. INTRODUCTION
Automated manufacturing systems have been widely spread in industry. One class of automated manufacturing systems is represented by production lines, which are provided with integrated computer controlled material handling servers known as hoists.

Many production systems such as electroplating lines and printed circuit boards(PCB) production lines employ hoists to transport jobs between different stations represented by the processing tanks. The processing times in the different processing stations can be either described by a fixed value or an interval value prescribed between a lower bound and an upper bound known as time window.

Hoist scheduling problem has received the attention of many researchers over the past decades either in fixed processing time environments Ada Che (2002) and V.Kats (1997),or in time window environments H.Chen (1998) and L.Lei (1994,2002). Comprehensive overviews and classification of the general hoist scheduling problem are given by Crama et al (2000).

The basic system has been studied extensively by Lei & Wang (1994), Lim (1997) and Chen et al (1998). Philips & Unger (1976) presented the first linear programming model for the system with multi function tanks.

Zhili Zhou (2003) and Che & Chu (2005) addressed the problem of a single hoist multiple tanks, where parts' processing times are given constants (zero width time window). A linear programming model was developed and the problem was proved to be polynomially solvable.

Hoist scheduling problem with time windows has been proved to be NP-hard in a very strong sense L.Lei (1989). L.Lei (1994), G.W Shaprio (1998) and Ada Che (2007) proposed branch and bound algorithms for the system with multi-tank stages and time window processing times. Joon-Mook Lim (1997) introduced a genetic algorithm based approach for a single hoist simple cyclic scheduling. The proposed algorithm is shown to be more efficient than the previous mathematical programming based algorithm. An artificial intelligence approach using constraint programming was proposed by Mak et al (1998) for the generation of cyclic schedules.

Other approaches extend simple cycles towards the generation of higher degree cycles. Che et al (2002) proved that higher degree cycles allows more than one part being loaded into the line while performing the cycle, thus increasing the chance of achieving an improved schedule. Grunder (1997) searched for the best layout of tanks for a single hoist and the schedule corresponding to a saturated line.

Multi hoist scheduling problem is much less studied. V.Kats (1997) and Karzanov (1978) addressed this problem of multiple hoists running on parallel tracks. Recently, research on cyclic scheduling problem for multi hoist systems has been reported. Lie & Wang (1991 & 1994) considered the cyclic two hoist problem without overlapping where a hoist is assigned to a zone corresponding to a set of tanks. Varnier (1997), Leung & Zhang (2003), Che & Chu (2004) and Zhili Zhou (2008) addressed the problem of multi hoist scheduling without overlapping. L.Lei (1993),V.Kats (2002), Leung (2006) considered the problem of minimizing the number of hoists running on a single track. Yun Jiang (2007) studied the no-wait multi hoist scheduling problem. The processing times in the tanks and the transfer times were considered constant parameters and a polynomial optimal solution to minimize the cycle length was developed. David Alcaide (2007) addressed the problem of an automated production system in which several robots were used for transporting parts between work stations in a carousel mode. All the processing, setup and transportation operations were assumed to be flexible and lying in prescribed time windows. The problem is shown to be polynomially solvable since the robot moves are known in advance. Manier (2008) used an evolutionary algorithms to minimize the cycle time while minimizing the number of hoists used where the assignment of the transport operation are not priori fixed.

In the single item case, a fixed sequence of hoist operations is repeated and referred to as a cycle. Turning to the multi-item problems the difficulty increases drastically and only a very few heuristic approaches are known since the problem with multiple item has been proven to be NP hard.

Yih (1994) and Cheng & Smith (1997) were the first who addressed the problem of hoist scheduling with multiple product types. S.Hindi (2004) and Henrick J.paul (2007) introduced a heuristic algorithm for solving single hoist multiple products scheduling problem. V.Kats (2008) considered the problem of multiple part types and setup time requirements between the processing steps for different parts at the shared stations. He showed that the problem is equivalent to the parametric critical path problem.

This paper addresses hoist scheduling problem for a system of multiple part types.Where each job type has its own processing sequence and may have different processing time in the same processing stage. The processing time of the jobs is prescribed between a minimum and maximum value. The considered system consists of multi-function tanks and multi-tank stages. All the jobs to be processed are ready at the input storage buffer at a given input sequence. A heuristic procedure is introduced for scheduling hoist moves. A simulation model is developed to apply the proposed heuristic procedure with the objective of minimizing the average cycle time.

2. PROBLEM DEFINITION

The problem under consideration addresses a system with successive processing stations represented by tanks and a hoist handling system which transports jobs between processing stations. The processing time of the jobs lies between a minimum and maximum values.

Such systems is known as no wait systems since the job has a maximum soaking time in each tank which it must not exceed. Thus the hoist must be ready to transfer the job to its next processing station before exceeding the permissible soaking time. A tank conflict occurs when a job has to be moved to a tank that is already occupied since there are no intermediate storage buffers between processing tanks. A hoist availability conflict occurs when a job has to move while the hoist is busy transporting another job. As a consequence, the scheduling of the hoist moves is required to avoid such resource conflicts.

Production stages with long processing times may exist. Such stages may create severe bottlenecks which greatly affect system's throughput and utilization. Therefore a group of duplicate tanks can be added to eliminate such type of bottleneck. This stage is denoted by multi tank stage. Also a number of processing stages may share one physical tank to reduce facility associated costs. This tank is denoted by multi function tank. The systems with multi function tanks and multi tank stages are known as extended systems.

Efficient scheduling of such hoists is evaluated by the degree of improvement of the production throughput.

3. NOMENCLATURE

M : Number of tanks in the system including input station and finished products station, The tanks are designated as: tank(i) where i=1……………M , tank (1) represents the input station while tank (M) is the finished products station.

N : Total number of jobs to be processed ; job(j) where j = 1,…………….N.

TPS(i) : Total processing stages of certain job 'i'.

OP(i,j) : Returns the process designated of job 'i' at the processing stage 'j' ,
Where i=1……………..N & j=1…………TPS(i).

LB(i,j) : Returns the lower bound of the processing time for job 'i' at the processing stage 'j' .
Where i=1……………..N & j=1…………TPS(i).

UB(i,j) : Returns the upper bound of the processing time for job 'i' at the processing stage 'j' .
Where i=1……………..N & j=1…………TPS(i).

AVT(op) : The set of available tanks that can perform process 'op', AVT(op)={tnk(k)} where k ∈ {2,………M}.

ME=[Me(i,j)] : Symmetric squared matrix representing time taken by the hoist to move empty from tank i, to tank j Where i & j = 1 , 2 , 3 ,…………… M-1, M.

MO=[Mo(i,j)] : Symmetric squared matrix representing time of a hoist loaded move from tank ' i' , to tank j Where i & j = 1 , 2 , 3 ,…………… M-1, M.

TL: Time taken by the hoist to perform its loading actions from any tank, (assumed constant for all tanks).

TU: Time taken by the hoist to perform its unloading actions at any tank, (assumed constant for all tanks).

h_t: Current position of the hoist at time instant t , where $t \geq 0$ & h=1,…………..M

$Tstart_{i,j,k}$: Starting time of processing job 'i' at the processing stage 'j' in tank 'k' .
Where i=1,……….N , j=1,……………TPS(i) & k $\in$ {1,…………….M}

$Tmove_{i,j,k}$: Starting time of moving job ' i' at the processing stage 'j' in tank 'k'.
Where i=1,……….N , j=1,……………TPS(i) & k $\in$ {1,…………….M}

$TS_{i,t}$: Returns the status of tank ' i' at time instant t; where $t \geq 0$, i=1,………N-1 & $TS_{i,t}$=1,……….N. For $TS_{i,t}=0$ then tank(i) is empty, for $TS_{i,t} \neq 0$ then tank(i) is occupied.

$JS_{j,t}$: Returns the current processing stage of job ' j' at time instant t; where $t \geq 0$, j =1,……….N & $JS_{j,t}$=1……..TPS(j).

$Lef_{i,t}$: The time left for the job currently immersed in tank 'i' to reach its lower bound of processing time.

$Res_{i,t}$: The time left for the job currently immersed in tank 'i' to reach its upper bound of processing time.

Ø : Represents the set of occupied tanks at any time.

IPJ : Represents the set of in process jobs(immersed in the processing tanks only) at any time.

IM : Represents the set of in process jobs and the ready job at the input buffer at any time.

TJ : Set of transferable jobs at any time.

CT : Set of tanks containing transferable jobs at any time.

CM : Represents the set of tanks containing candidate jobs to be moved.

NP_i : Represents the next process of the job 'i', where i $\in$ IM .

ST_i : The set of tanks that job 'i' can be transferred to, for the next process 'NP_i' , ST_i = AVT(NP_i).

ET_i : Represents the earliest time to move the job currently immersed in tank 'i'.

r_i : Represents the next processing tank for the job currently immersed in tank 'i' .

CM1: Represents a set of tank indices containing jobs that exceeded their lower bound time of processing at the time of observation (named: " Immersed post-lower bound jobs").

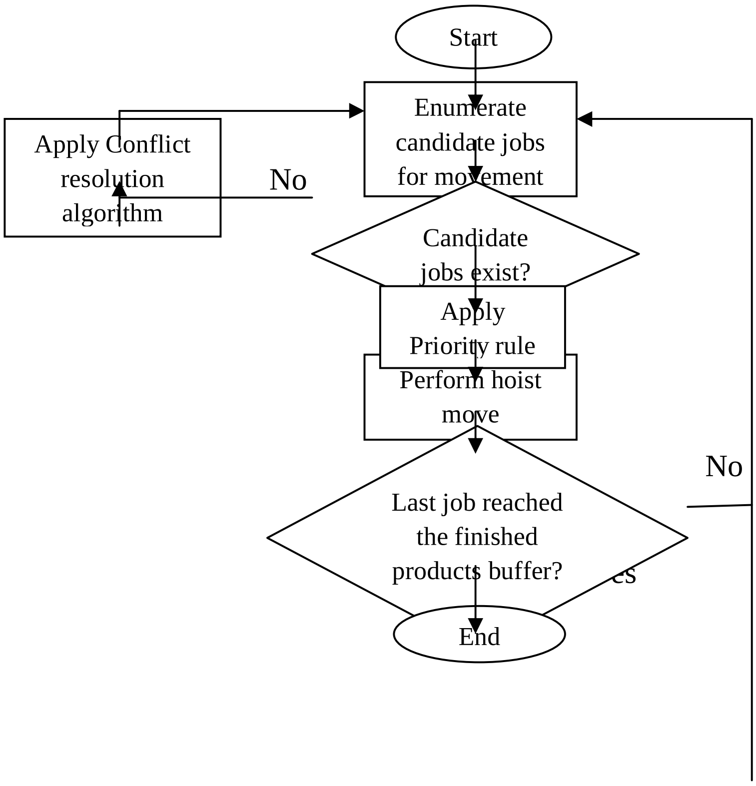
CM2: Represents a set of tank indices containing jobs that are below their lower bound time of processing at the time of observation (named: " Immersed pre-lower bound jobs").

Stank: Index of the selected tank from which the immersed job will be moved as decided by the applied priority rules.

TOS(i,j) : The index of the tank that processed job 'i' at the processing stage 'j'.

4. THE PROPOSED ALGORITHM

A heuristic algorithm is developed based on the observation for the system status (before every hoist move) within time domain in order to enumerate all candidate jobs to be moved from one tank to the next in processing sequence without spoiling the remaining immersed jobs at other processing tanks. A nested priority rule is then applied to determine which job to be moved among all candidate jobs. As a conflict in hoist actions may arise while advancing the solution with time, a conflict resolution algorithm is developed. The algorithm resolves the conflict utilizing the slack time in the processing at the prior stations. The following are the steps of the developed heuristic algorithm:



4.1-ENUMERATING ALL CANDIDATE JOBS FOR MOVEMENT.

At any time t, enumerate all candidate jobs to be moved by determining the tanks' occupancy status including the input buffer. If $TS_{i,t} \neq 0$, then tank(i) is assigned to the set of occupied tanks Ø . Consider all occupied tanks (i.e: $\forall$ tank (i) where i □ Ø) , investigate the possibility of moving any job from its current tank to next processing tank, this check is made through the following procedure:

(a) Determine the set of immersed jobs IM , IM = $\{TS_{i,t}\}$ where $\forall$ i □ Ø.

(b) Determine the current processing stage $JS_{i,t}$ for all immersed jobs.

(c) Determine the next process NP_j for all immersed jobs , $NP_j = OP(j, JS_{j,t}+1)$ $\forall j \in IM$.

(d) Determine the available tanks ST_j for the next process NP_j $\forall j \in IM$, $ST_j = AVT(NP_j)$.

(e) Check the availability of a vacant tank at the next processing stage for all immersed jobs.

(f) Assign any job with the next processing tank(s) vacant to the set of transferable jobs TJ.

(g) Determine the nearest vacant tank at the next processing operation for all tanks containing transferable job,Such that $r_i = \{ l \mid Mo(i,l) = \min(Mo(i,k)) \ \& \ TS_{k,t} = 0 \}$ i □ CT, k □ ST_j.

(h) Determine the earliest time that the hoist can load any transferable job from its current tank, Such that $ET_i = \max\{ Me(h_t,i) , Tstart_{j,seq,i} + LB_{j,seq} \}$ i □ CT , j = $TS_{i,t}$ & seq=$JS_{i,t}$

(i) Check the possibility of moving any transferable job from its current tank to next processing tank in sequence without exceeding the upper bound time of the remaining immersed job. If this condition is satisfied for any tank containing transferable job, then assign the job to the set CM :
$\forall$ i □ CT & q ={tank(k) | k □ {Ø} & $k \neq i$ & $k \neq 1$ } , where j = $TS_{q,t}$, seq=$JS_{j,t}$

If $ET_i + T L + Mo(i,r_i) + TU + Me(r_i,q) \leq tstart_{j,seq,q} + UB_{j,seq}$

Then set i □ CM

(j) If $\{CM\} \neq \Phi$, then apply the priority rule to select a tank where its immersed jobs is to be moved. Otherwise, no hoist actions can be performed and the conflict resolution algorithm should be applied.

4.2-PRIORITY RULE

The application of the proposed priority rule requires the following:

4.2.1 Determining immersed jobs status:

Determine the status of the immersed jobs either Post-Lower Bound Job (POLB) or Pre-Lower Bound Job (PRLB). In case of the (POLB) job calculate the residual length of time from the observation time t up to the upper bound of their time window. For the jobs designated as (PRLB) calculate the required time to reach their lower bound processing time.

Compose a set CM1 to include all tank indices containing the (POLB) jobs. Similarly, compose another set CM2 including all tank indices of the (PRLB) jobs.

$\forall\ i \in \{CM\}$ & $i \neq 1$, $j = TS_{i,t}$ & $seq = JS_{j,t}$

if $Tstart_{j,seq,i} + LB_{j,seq} \leq t$

$Res_i = Tstart_{j,seq,i} + UB_{j,seq} - t$ set $i \in CM1$

otherwise $Lef_i = Tstart_{j,seq,i} + LB_{j,seq} - t$ set $i \in CM2$

4.2.2 APPLYING THE DEVELOPED PRIORITY RULE HEURISTIC

If $CM1 \neq \Phi$, then the job with minimum time left to reach the upper bound will be considered for movement to its next processing tank in sequence. i.e;

$Stank = \{ m \mid Res_m = \min(Res_i) \}$ $i \in CM1$

Otherwise, if the input buffer belongs to the set CM, then the ready job at the input buffer will be considered for movement i.e: Stank = 1.

Otherwise , the job with minimum time left to reach the lower bound will be considered for movement $Stank = \{ m \mid Lef_m = \min(Lef_i) \}$ $i \in CM2$

4.3-PERFORMING THE SELECTED HOIST ACTION AND UPDATING VARIABLES

The job'j' currently immersed in tank 'Stank' is selected to move to its next process in sequence ' NP_j' represented by tank 'r' (where $r = r_j$) . All values must be updated considering the time that the job being picked from the tank, the time taken by the hoist to move to next tank and the time during which the job is loaded in the next tank, the new position of the hoist, the new status of the tanks, the new status of the jobs, and the time t.

Set $Tmove_{j,seq,Stank} = ET_{Stank}$ where $j=TS_{Stank,t}$, $seq= JS_{j,t}$

$Tstart_{j,seq+1,r} = Tmove_{j,seq,Stank} + TL + Mo(Stank,r) + TU$

$t = Tstart_{j,seq+1,r}$

$TS_{r,t} = TS_{Stank,t}$

$TS_{Stank} = 0$ for $s \neq 1$ i.e: 's' is a processing tank

$TS_{Stank} = TS_{Stank} + 1$ for $s=1$ i.e: 's' represents the input buffer

$JS_{j,t} = JS_{j,t} + 1$

$h_t = r$

$TOS(j,seq+1) = r$

If all the required jobs are processed (i.e : the last job reached the output buffer) then the problem is terminated, Otherwise a new enumeration is performed and the above mentioned sequence is repeated.

4.4-CONFLICT RESOLUTION ALGORITHM

Two types of conflict may occur, the first is called tank conflict and the second is called hoist conflict. A tank conflict occurs when the job is required to move to a tank which is already occupied with another job for time duration that exceeds the upper bound of the prior job. A hoist conflict occurs when the job is required to move, but if the hoist moved such job to its next processing tank, another in-process job will exceed its upper bound time of processing. If the enumeration procedure doesn't yield any possible moves i.e: CM=Φ as one of the above mentioned cases, then a conflict resolution algorithm has to be applied with the following steps:

(a) Determine the first job among all in process jobs that will exceed its upper bound time of processing due to resource conflict, this can be represented mathematically as follows;

Set $t = t + 1$

$\forall$ tank(i) where $TS_{i,t}$ □ IPJ , $j = TS_{i,t}$, $seq = JS_{i,t}$

If $t > Tstart_{j,seq,i} + UB_{j,seq}$

Then $b= i$

Otherwise $t= t + 1$ & the condition is rechecked within the present step

(a) Determine the set of available tanks 'ST' for the next processing stage of the job 'j' immersed in tank 'b'; such that $ST_j = AVT(OP(j,JS_{j,t}+1))$

(b) Check the availability of any vacant tank in the set of tanks determined in the previous step. If no vacant tank is available, this represents a case of a tank conflict and the conflict resolution procedure is to be applied to this job.

$\forall i \in ST_j$ if $TS_i \neq 0$ tank conflict case, go to step(e)

(c) If a vacant tank exists for the next processing stage (which represents a case of hoist conflict), determine the second job that is expected to be spoiled. The conflict resolution procedure is to be applied to this job, as this job is the cause of the hoist conflict.

$\forall$ tank(i) where $TS_{i,t}$ □ IPJ & $i \neq b$, $j = TS_{i,t}$, $seq = JS_{i,t}$.

If $t > Tstart_{j,seq,i} + UB_{j,seq}$

Then $b = i$

Otherwise $t = t+1$ & the condition is rechecked within the present step

(d) After determining the job on which the conflict resolution will be applied, consider the following procedure: Backtrack the job processing by one stage and check the availability of residual time up to the upper bound of the time window. Prolong the processing of the job at that stage by delaying its transfer to next processing tank by one unit time. If no residual time is available at the previous processing stage, consider the preceding stage and apply the same check. Adjust clock time and resume the solution procedure.

Consider job 'd' immersed in tank 'b', where $d=TS_{b,t}$ & $seq=JS_{d,t}$

If $Tmove_{d,seq-1,TOS(d,seq-1)} - Tstart_{d,seq-1,TOS(d,seq-1)} \leq UB_{d,seq-1}$

Then $t = Tmove_{d,seq-1,TOS(d,seq-1)} + 1$ & Start enumeration

Otherwise $seq = seq -1$ & the check is repeated

5. SOLUTION OF BENCH MARK PROBLEMS

The proposed algorithm was tested on three bench mark problems. The data of these problems can be found in Che & Chu (2007). Two of the test problems are real life instances from actual industrial applications where the electroplating line processes a single type of product. The results obtained from the application of the present algorithm to the three problems compared to that given in the literature are shown in Table (1). The results show that the proposed algorithm gives higher average cycle time values ranging between 2.5 and 14 % compared to that obtained by exact mathematical solutions. Such differences go back to the fact that the proposed algorithm is based on heuristics which employ priority rules and the conflict resolution algorithm. The differences can be attributed to the fact that the algorithm simulates the hoist actions starting from the initial loading of the system where all jobs are at the input storage buffer and all the tanks are empty. The results of ramping up the system to the steady state further affect the obtained results, since neither the optimum values of processing times nor the hoist moves sequence are priori known. This is in contrast with the mathematical methods in which the sequence of hoist actions is predetermined ensuring optimal solution.

Also the algorithm was tested on a problem that includes two different products being processed. The problem is found in S.Hindi(2004). The results in Fig (1) show that the solution obtained is compatible with that given in the literature. Although the size of the problem is small, the proposed algorithm is capable of solving multi product types with considerable efficiency. Other exact solutions, which depend on enumerating possible feasible cycles by applying B&B algorithm and further using LP to obtain the optimal processing time at each process, will not be capable of solving multi product problems where each product type has different value of processing times (lower bound and upper bound processing times).

Table 1: Comparison of the results of applying the present algorithm with Bench mark problems of single product type.

problem	number of tanks	number of stages	average cycle time (present work)	Optimum cycle time (others)	Percentage deviation
Philips	12	12	534.5	521	2.59%
Ligne 1	12	12	457.5	418	9.44%
Ligne 2	14	14	810	712	13.76%

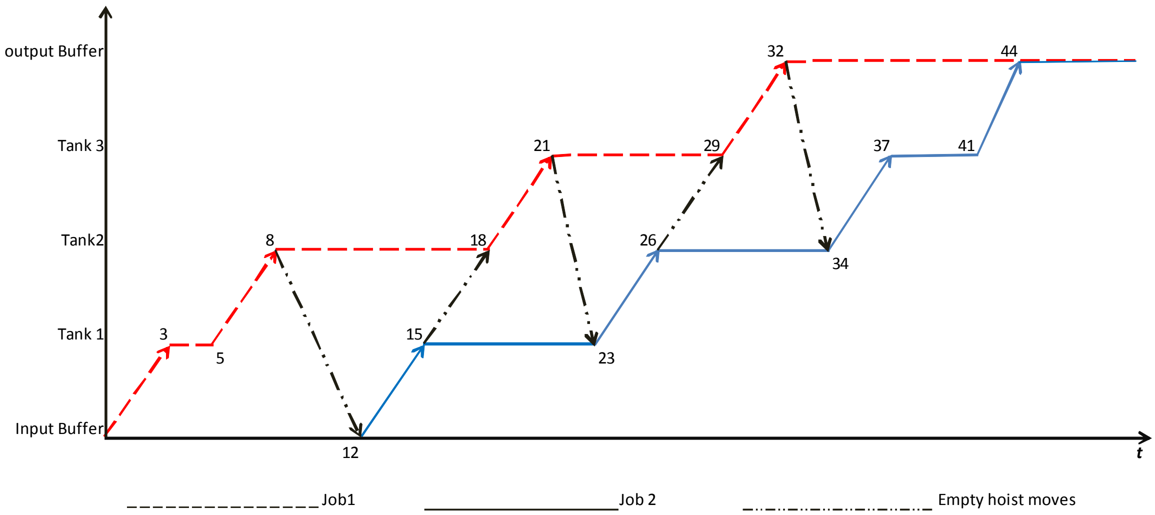


Fig .1. Hoist schedule for the Khalil S.Hindi (2004) problem (obtained from the present work)

6. CONCLUSION

The proposed algorithm addresses the problem of a system for processing multiple product types where the processing time of the jobs is prescribed between minimum and maximum values. Extended systems were considered including multi tank stages that assume the addition of a group of duplicate tanks at same stages of long processing time, and the case of multi function tanks where more than one processing stage may share a single tank.

The results obtained for single type of product by the present heuristic were comparable to that obtained by exact methods with some deviations. The difference between the obtained and bench mark problems cycle time is 14 % for industrially long processing lines consisting of 12 and 14 processing tanks. Also the algorithm was tested to schedule the system of two different types of products. The results prove that the proposed heuristic can be used efficiently for single hoist scheduling either for a single product type or multiple types of products.

Future work may include extending this work to a multi hoist problem which may improve system's throughput and utilization considerably.

ACKNOWLEDGMENT

The authors would like to thank Dr. M.El Beheiry;Assistant lecturer at Ain Shams University for his help in the problem formulation.

REFERENCES

1) Zhili Zhou., Ling Li , (2003). Single hoist cyclic scheduling with multiple tanks: a material handling solution. Computers and operation research, 30 , 811-819 .

2) Ada Che., C. Chu, (2005). A polynomial algorithm for no-wait cyclic hoist scheduling in an extended electroplating line. Operations research letters, 33 , 274-284.

3) Ada Che., C. Chu, (2007). Cyclic hoist scheduling in large real- life electroplating lines. OR Spectrum, 29 , 445-470.

4) Zhili Zhou., Ling Li , (2009). A solution for cyclic hoist scheduling of multi-hoists without overlapping. Ann Oper Res, 168, 5-21.

5) David Alcaide, C.Chu. , V.Kats,(2007). Cyclic multiple-robot scheduling with time-window constraints using a critical path approach. European Journal of operational research, 177, 147-162.

6) Marie-Ange Manier., S. Lamrous.,(2008). An evolutionary approach for the design and scheduling of electroplating facilities. J Math Model Algor, 7, 197-215.

7) Khalil.S.Hindi., K.Fleszar., (2004). A constraint propagation heuristic for the single-hoist, multiple-products scheduling problem. Computers and industrial engineering, 47, 91-101.

8) Henrick. J.Paul., C.Bierwirth., H.Kopfer.,(2007). A heuristic scheduling procedure for multi-item hoist production lines. Int. J . Production economics, 105 , 54-69.

9) V.Kats., L.Lei., E.Levner.,(2008). Minimizing the cycle time of multiple-product processing networks with a fixed operation sequence, setup, and time-window constraints. European Journal of operational research, 187, 1196-1211.

10) Yun Jiang., J.Liu.,(2007). Multihoist cyclic scheduling with fixed processing and transfer times. IEEE Transactions on Automation Science and Engineering. Vol 4.No 3.July2007 .

11) Janny M.Y.Leung., Eugene Levener.,(2006).An efficient algorithm for multi-hoist cyclic scheduling with fixed processing times. Operations Research Letters, 34 , 465-472.