International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:9, No:6, 2015

Enumerative Search for Crane Schedule in Anodizing Operations

Kanate Pantusavase, Jaramporn Hassamontr

Abstract**—**This research aims to develop an algorithm to generate a schedule of multiple cranes that will maximize load throughputs in anodizing operation. The algorithm proposed utilizes an enumerative strategy to search for constant time between successive loads and crane covering range over baths. The computer program developed is able to generate a near-optimal crane schedule within reasonable times, i.e. within 10 minutes. Its results are compared with existing solutions from an aluminum extrusion industry. The program can be used to generate crane schedules for mixed products, thus allowing mixed-model line balancing to improve overall cycle times.

Keywords**—**Crane scheduling, anodizing operations, cycle time minimization.

I. Introduction

ONE of the key competitive edges for manufacturing industry is speed. More and more companies now rank speed as a competitive advantage over cost. To improve speed of delivery, production lead time must be reduced. While we can focus our efforts on various components of lead time, the most effective strategy would be to reduce production cycle time itself.

This paper focuses on minimizing cycle time in a chemical process where loads of products are transferred from one bath to another using a set of cranes. Typically there are several baths, or process steps, for each load to go through. A specific example of such operation is anodizing process for aluminum profiles. Anodizing operation is a finishing process aiming to put protective film on metal parts. The most popular film colors are clear, brown and black.

Aluminum profiles are usually clamped to a rack or work holding device such as that shown in Fig. 1. A load of aluminum profiles is dipped in a number of tanks of acids, clean water, dyes or other types of chemicals for varying lengths of time and for varying amperage of electricity flow from tank walls through liquids into aluminum profiles. Each load has a predetermined process time for each process step. Process time at a tank includes (1) time duration for the load to be immersed in the liquids and (2) dripping time after the load is lifted from liquid before transporting to the next tank.

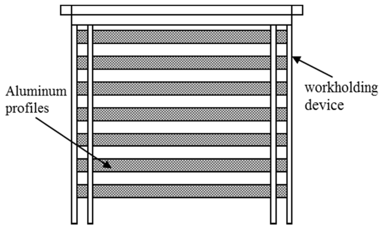


Fig. 1 Aluminium profiles in work holding device for anodizing operations

In practice, tanks are constructed next to one another in sequence and sometimes called bath. Fig. 2 illustrates an example of baths allocated for each process step. In order to improve bottleneck in the operation, some process steps have more than one bath so that several loads can be processed in parallel. Five cranes are on the same rail, each covering a range of baths. Overlaps between cranes are necessary to transport loads from the beginning to the end of the anodizing line. Each crane operates in similar fashion. It transports a load from one bath to another, then it leaves the load in the bath until process time is reached, during which it can be used to transport another load. It should be noted that there are several baths designated for clean water rinsing, denoted by CW rinsing. This is to ensure that cranes are moving all the loads forward, no backward movement is allowed.

Since aluminum profiles within a load undergo the same process condition throughout the anodizing line, they are usually of the same product, i.e. aluminum profiles of the same cross section, same length and same surface specification. However, an anodizing line may handle several products at the same time, much the same way as mixed-model assembly line in automotive industry. This can achieved by setting process routing and process conditions for each individual load. Different film colors require different electro-coloring times. Load sequence must be pre-determined and managed by human operators clamping aluminum profiles to the rack at the beginning of the anodizing line, denoted by area 43 in Fig. 2. Moving loads into and out of anodizing line is done by transfer cranes, which are separated from the five cranes used in the anodizing line.

This paper addresses an enumerative search approach to generate a near-optimal crane schedule for both single and mixed products in anodizing operation. The time interval to introduce a load into the anodizing line is minimized. An object-oriented program is developed to illustrate the approach. Actual data from an aluminum anodizing plant are used to validate its feasibility and effectiveness.

K. Pantusavase is with the Department of Industrial Engineering and Management, Silpakorn University, Sanamchandra Palace Campus, Nakornpathom 80305 Thailand.

J. Hassamontr is with the Department of Production Engineering, King Mongkut's University of Technology North Bangkok, Bangkok 10800 Thailand (phone +6681-989-0705; e-mail: jph@kmutnb.ac.th).

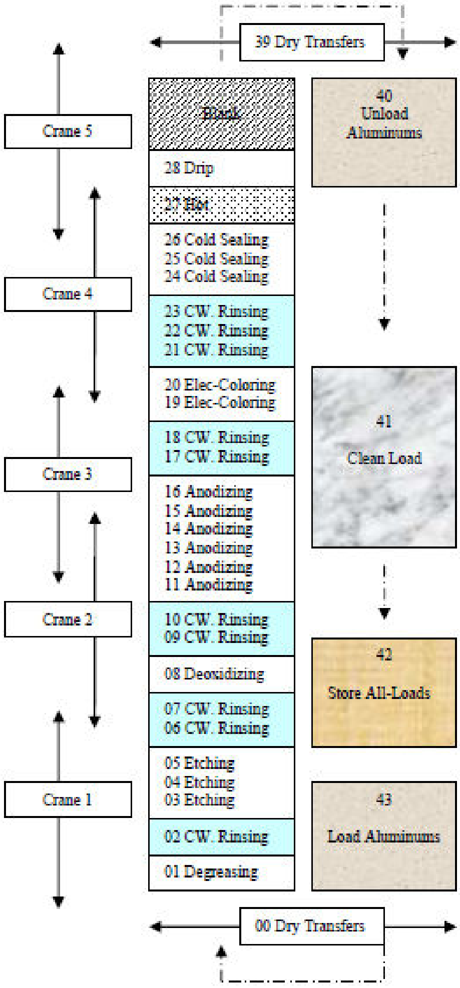


Fig. 2 Example layout of a typical anodizing operation line

II. Literature Review

Crane scheduling was introduced first by [1] in 1981. Reference [1] had proposed a general model for multiple cranes sharing the same gantry. Several jobs were handled simultaneously. The model was inspired by a smelter workshop. Algorithms to generate a feasible solution were proposed for a single operation problem where each job has identical process routing and processing times. Later, the model was extended to 2-operation crane scheduling problems in [2].

A similar class of problem can be found in hoist scheduling. Reference [3] studied cyclic scheduling of 2 hoists. Jobs were identical. The problem was formulated as an assignment problem. Reference [4] extended the study to hoist scheduling in an ECB plating line where no buffering is allowed. Processing times must vary within certain ranges. Reference [5] investigated scheduling problem of 2 factory cranes on the same track. The problem was formulated as a mixed-integer programming. A heuristic approach was proposed.

Another class of related research involved scheduling of chemical process. Reference [6] provided a good overview of discrete-time and continuous time model for scheduling chemical process. Reference [7] developed mixed integer programming model for multi-product batch process scheduling. Linear programming based heuristics were proposed. Extensive overview of scheduling chemical process with no immediate buffers between machine and no wait time between machines allowed was provided by [8].

Lastly, another class of research for crane scheduling involves port terminal. Reference [9] modeled the crane scheduling for port terminal. For one crane case, an approximation method was proposed. Then later [10] provided a more rigorous treatment for the optimal solution. Reference [11] extended crane scheduling problem to include berth scheduling at port terminal. There, simulated annealing technique was used. Reference [12] formulated stacker crane scheduling problem where one crane is used to transport bins of materials to various work stations.

Research works in crane scheduling are vast. Most of them focused on maximizing throughput or minimizing make span. This research attempts to utilize object-oriented programming approach to search for minimal cycle time. As in several researches, we considered multiple cranes on the same track, with blocking and no-wait property. Several products are also handled simultaneously. As computing power from a personal computer increases the enumerative search has proved useful and efficient to search for a near-optimal solution in such a complex problem.

III. Methodology

As the number of cranes and loads increases, searching for a feasible crane schedule can be the most effective way to find a solution. Moreover, object-oriented programming property is very useful to model interactions between loads, cranes and baths. The objective of the search is to minimize time between successive load inputs into the anodizing line with blocking and no-wait property. Here, we will use the terms, time between successive loads and cycle times interchangeably. The time between successive load inputs, regardless of whether they are similar or different products, will be assumed constant. The following are user-defined parameters for the anodizing process and search parameters in the program. These must be known before the search starts.

- Load sequence can be of identical products or different products, here it is assumed that users are responsible to plan product mixes to balance average processing time at bottleneck process.
- Each crane c has a starting operating range to cover baths from L_c to R_c. For example, in Fig. 2, crane 1 covers bath number 00 to 07. The algorithm will fine-tune these ranges further so that workload among cranes is balanced.
- Each process step is denoted by starting bath number, B_p and the number of baths, B_p’. For example, from Fig. 2, anodizing process starts from bath number 11 and it has 6 baths;
- Process time is determined from 2 components-the time required for a load to remain in liquid in the bath and the delayed time to allow liquid from a load to drip before

moving to the next bath.

- Baths are constructed in linear sequence, as shown in Fig. 2, and in equal-spaced. Each bath can take only one load at a time.
- Crane's transportation time is assumed to be dependent upon the difference between bath numbers, regardless of whether it carries a load.

A. Modeling of Crane Behavior

In order to calculate the time it takes for a crane to move a load from bath b_1 to b_2, we note that crane must accelerate from zero speed to certain value then its speed remains constant until it almost reaches the terminal bath. Then it decelerates and stops at the terminal bath. It is useful to denote the following time elements:

- t_1 time taken for a crane to move a load smoothly for 1 bath length, i.e. $|b_2\text{-}b_1|=1$;
- t_2 time taken for a crane to move a load smoothly for 2 bath lengths, i.e. $|b_2\text{-}b_1|=2$;
- t_0 time taken for a crane to move a load at constant speed for 1 bath length.

All of these can be obtained and fine-tuned from the experiments at actual process setting. The time required for a crane to move a load from bath b_1 to b_2 can be calculated by

$$T_{b_1,b_2} = \begin{cases} t_1 & d = 1 \\ t_2 & d = 2 \\ t_0(d-2) + t_2 & otherwise \end{cases} , \tag{1}$$

where $d = |b_2\text{-}b_1|$.

The time to occupy a crane can be determined from other time constants such as (1) time to lower a load to the bath, T_d, (2) time to lift a load from the bath, T_u and (3) time to let liquid drips from the load before moving to another bath, T_w. These time elements are used to reserve operation time's for crane c, from Tc_s to Tc_f.

$$Tc_f = Tc_s + T_u + T_w + T_{b_1,b_2} + T_d \tag{2}$$

This also leads to reserving times for the bath, from T_{bs} to T_{bf} for a particular load. Let $T_{a,p}$ be the time required for load a to be processed at process step p. We can reserve times for the bath from T_{bs} to T_{bf} as:

$$Tb_s = Tc_s + T_u + T_w + T_{b_1,b_2} \tag{3}$$

$$Tb_f = Tb_s + T_d + T_{a,p} + T_u + T_w. \tag{4}$$

Once a process is finished, a crane is supposed to come and pick up the load immediately, according to no-wait property. Thus, the next start time for this crane is identified.

$$Tc_s = Tb_s + T_d + T_{a,p} \tag{5}$$

As the crane will move to another bath, we can identify crane finish time by resuming to (2). Fig. 3 illustrates an example of how crane c is scheduled to pick up a load a, move to bath b, then come back to pick up the load to move to another bath. In many cases, the crane that drops off a load at a bath may not be the same one coming to pick up the load to move to another bath. Fig. 3 illustrates an example where the same crane is used to put a load in a bath and to pick up the load to move to another bath.

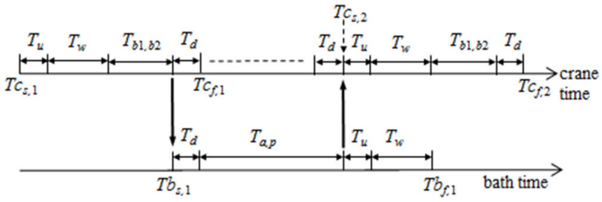


Fig. 3 Reserving times between crane and bath

Classes of crane and baths are developed to maintain the current status of their occupied times. As we continue to introduce a load into the anodizing line, new time slots for crane movements must be added to the current crane schedule. Fig. 4 demonstrates how a new time interval, denoted by Tc_s and Tc_f, for a crane is checked with its current schedule. If the new time slot cannot fit in the crane's available time, schedule of the next crane is searched. There will be at most 2 cranes covering the same bath. Consequently, if the next crane cannot find available time, the time slot is deemed infeasible; the search for next cycle time will begin. The same analysis can be done for the bath, with a finite number of baths available for each process step. If no bath is available to take the loads, the schedule is once again deemed infeasible as blocking occurs.

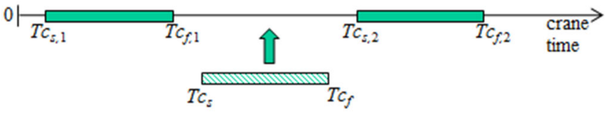


Fig. 4 Adding a new time slot to current crane schedule

B. Exploring Crane Moving Range

As discussed previously, users must specify the largest possible ranges to cover, from bath L_c to R_c, for each crane c. It is obvious that workloads among cranes must be balanced so that the line cycle time is improved. The overlaps between each crane should be maximized so that cranes can help one another. However, as crane covering ranges increase, so do the chance for crane collision. To manage crane covering range more effectively, three strategies are added to the algorithm.

First, we used mixed-base enumeration algorithm to generate all possible crane covering ranges. This can be done from, say, crane 5 to crane 1. Fig. 5 shows a snapshot of alternatives generated by the algorithm. Five shaded areas represent the range of baths for 5 cranes to cover. Each crane covering range overlaps the ones next to it, but not shown in the figure for simplicity and clarity. Two cranes next to each other will have at least 1 overlapping process step. That is, crane 1 has some overlaps over crane 2, whereas crane 2 has overlaps with both cranes 1 and 3. In short, the algorithm will start from having cranes 1-4 covering the largest possible range while crane 5 has the smallest one. The algorithm generates the next alternative by expanding range for crane 5, reducing range of crane 4. This is done until crane 4 has the lowest range, then the algorithm starts to lower the range for crane 3, while resetting cranes 4 to the largest range and crane 5 to the lowest range. The algorithm enumerates all crane covering ranges possible. If there is any alternative where more than two cranes cover the same bath range, i.e. when a crane is assigned to cover only one process step that is not in the beginning or the end of the line, it is pruned out of consideration. Each remaining alternative for crane coverage will be explored to search for the best possible cycle time. This will be explained in details in the next section.

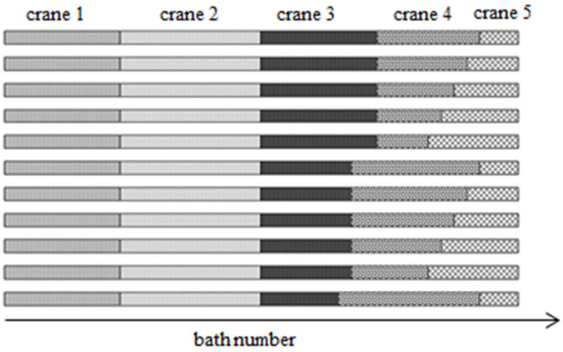


Fig. 5 Enumerative search of range where each of 5 cranes can cover

The second strategy is used to avoid crane collision at each bath. The algorithm will continuously monitor each individual bath that is covered by two cranes. Fig. 6 illustrates 4 steps to avoid the risk of crane collision. Whenever two pairs of bath times are reserved close to each other, say at bath b_2, and are from two different cranes, say from crane c and c+1, the previous bath that crane c+1 took the load from, bath b_1, is identified. Delay time, T_D, is added to the dripping time at bath b_1, thus shifting the bath time for the load at b_2 to the right.

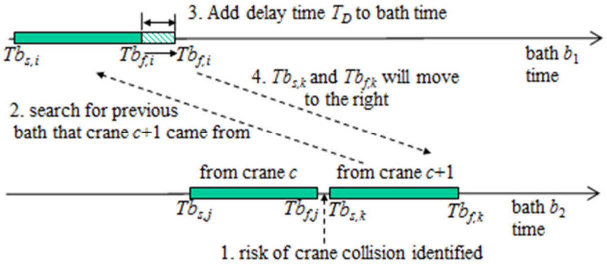
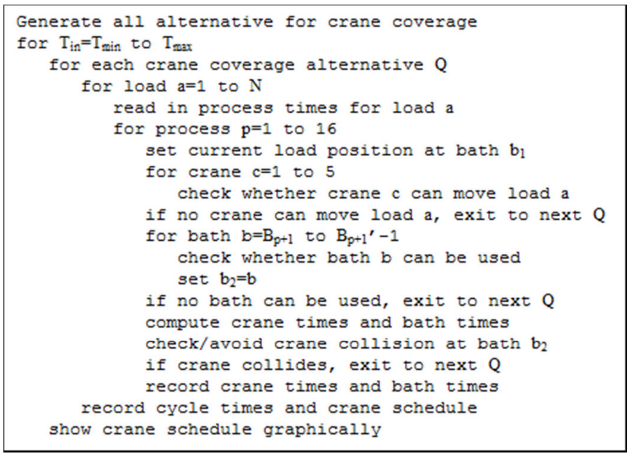

3. Add delay time T_D to bath time
$Tb_{s,i}$
$Tb_{f,i}$
$Tb_{f,i}$
bath b_1 time
4. $Tb_{s,k}$ and $Tb_{f,k}$ will move to the right
2. search for previous bath that crane c+1 came from
from crane c
from crane c+1
$Tb_{s,j}$
$Tb_{f,j}$
$Tb_{s,k}$
$Tb_{f,k}$
bath b_2 time
1. risk of crane collision identified

Fig. 6 Strategy to avoid crane collision at a bath

The third strategy is to check for crane collision itself. Cranes may collide anywhere while they are moving, not only when lowering or lifting a load at a bath. To check whether cranes collide into each other, crane position in terms of bath number is denoted byline segments over time. All 5 crane schedules can be presented in a graph as shown in Fig. 8. If line segments from 2 different cranes intersect each other, those two cranes collide. When this is observed, crane schedule is considered infeasible; the next cycle time will be investigated. In our algorithm, whenever a crane is determined to move from one bath to another, the corresponding line segment will be generated and checked for intersection with other pre-existing lines.

C. Crane Scheduling Strategy

Once details from section A and B are worked out, it is obvious how crane schedule can be iteratively searched for the lowest cycle time. The algorithm is summarized in Fig. 7. First all the crane covering range alternatives are generated and saved. Without loss of generality, the number of crane is set at 5 and there are 16 process steps in the anodizing process, such as that shown in Fig. 2. The time between successive loads, T_{in}, is defined incrementally. For the benefit of computational time required, T_{in} is searched at a step of 1 s. Successive loads are not necessarily the same products as the previous ones. Thus, they may not have the same process routing and times. At each value of T_{in}, all crane covering alternatives are explored. Crane schedule and bath times for all process steps are generated one load at a time until all the loads are successfully planned. If no crane is available to move the load, or if no bath is available to process the load at any point, the algorithm resets both crane and bath schedule in order to start all over at the next crane covering alternative. After passing availability test for crane and bath, process times may be adjusted to prevent cranes from running into each other at a bath. Every added crane movement must be checked for possible collision. If this condition is passed for all loads, the solution is found. As the algorithm increments cycle time at a step of 1 s., the first feasible solution found will become a reasonably solution, if not near optimal. While setting cycle time search finer than 1 s. can be done computationally, it is difficult to control cranes in practice, and thus not done here.



```
Generate all alternative for crane coverage
for $T_{in}=T_{min}$ to $T_{max}$
   for each crane coverage alternative Q
      for load a=1 to N
         read in process times for load a
         for process p=1 to 16
            set current load position at bath $b_1$
            for crane c=1 to 5
               check whether crane c can move load a
            if no crane can move load a, exit to next Q
            for bath b=$B_{p+1}$ to $B_{p+1}$'-1
               check whether bath b can be used
               set $b_2$=b
            if no bath can be used, exit to next Q
            compute crane times and bath times
            check/avoid crane collision at bath $b_2$
            if crane collides, exit to next Q
            record crane times and bath times
      record cycle times and crane schedule
   show crane schedule graphically
```

Fig. 7 Algorithm to search for crane schedule

IV. CASE STUDIES

Based on the strategy developed in previous section, a computer program was developed using Visual Basic Application, interfacing with MS Excel. Actual process parameter data were obtained from an aluminum profile plant where its anodizing processes are automated by pre-installed software. The software that accompanies the system is capable of generating crane schedule mainly for single product. The crane schedules generated in this research will be verified and compared with those obtained from the pre-installed software.

Table I shows process assignment for each bath that the company usually uses. Process times at most of the process steps are similar, except for electro-coloring process, which depends on the coating color. White color actually means no color coating, thus loads with white specification will not need to go through electro-coloring process. Table II shows process times for all 16 process steps. Process times in electro-coloring process also vary with film thickness. However, for simplicity, only standard film thickness is used here.

TABLE I
BATHS FOR EACH PROCESS STEP

Process step	Starting bath no.	Ending bath no.
Dry transfer	0	0
Degreasing	1	1
CW rinsing	2	2
Etching	3	5
CW rinsing	6	7
Desmut	8	8
CW rinsing	9	10
Anodizing	11	16
CW rinsing	17	18
Electro-coloring	19	20
CW rinsing	21	23
Cold sealing	24	26
Hot water	27	27
Dripping	28	29
Blanking	30	38
Dry transfer	39	39

TABLE II
PROCESS TIMES IN S. FOR CRITICAL PROCESSES

Process step	white	brown	black
1 Dry transfer	0	0	0
2 Degreasing	180	180	180
3 CW rinsing	120	120	120
4 Etching	480	480	480
5 CW rinsing	120	120	120
6 Desmut	240	240	240
7 CW rinsing	120	120	120
8 Anodizing	1800	1800	1800
9 CW rinsing	120	120	120
10 Electro-coloring		***230***	***900***
11 CW rinsing	120	120	120
12 Cold sealing	420	420	420
13 Hot water	180	180	180
14 Dripping	180	180	180
15 Blanking			
16 Dry transfer	120	120	120

TABLE III
STARTING CRANES' OPERATING RANGE

Crane no.	Starting bath no.	Ending bath no.
1	0	10
2	2	18
3	8	20
4	17	32
5	21	39

TABLE IV
COMPARISONS OF CRANES' OPERATING RANGE FOR BROWN COLOR

Crane no.	Current use(plant)	Suggested by the program
1	0-7	0-4
2	6-10	3-8
3	9-20	8-19
4	19-27	19-25
5	21-39	24-39
cycle time(s.)	348	323

Without loss of generality, process times in Table II have already included dripping time, T_w, and crane lowering time, T_d. The time required for a crane to move from bath b_1 to b_2, according to (1) can be simplified to:

$$T_{b_1,b_2} = \begin{cases} 4 & d = 1 \\ 8 & d = 2 \\ 3(d-2)+8 & otherwise \end{cases} , \qquad (6)$$

Table III presents the starting covering range for each crane, as specified by users. Incremental time search for cycle time is with a step of 1 s. Computational time, in most cases, are within 10 minutes using a typical personal computer.

There are 5 cases considered in this paper. First, crane schedule for brown-color loads, currently used by the plant, is compared with the result from our program. Table IV compares crane covering ranges from that of the plant versus that offered by our program. The cycle time from the plant is 348 s., while that achieved by our program is 323 s., roughly 7% lower. Fig. 8 shows the crane schedule for the brown color, currently used by the plant. Crane schedule suggested by our program is shown in Fig. 9. It should be noted that without searching for crane covering range extensively, it would be difficult to find substantial improvement on the cycle times.

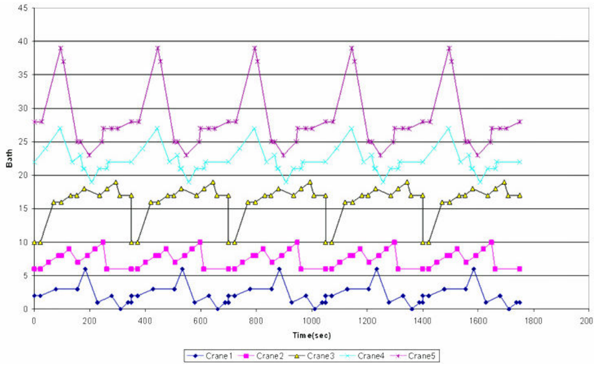


Fig. 8 Crane schedule for brown color (used by the plant)

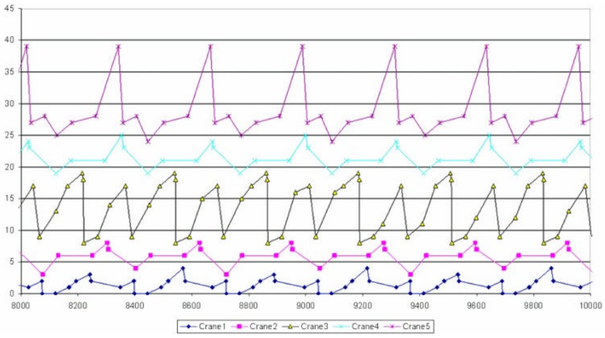


Fig. 9 Crane schedule for brown color (calculated by the program)

The second case considered is white color loads where electro-coloring time is not needed. The crane covering range used by the plant is compared with that suggested by the program in Table V. Both crane covering ranges are very similar, yet the cycle times achieved are identical to those from brown color, suggesting that electro-coloring time for brown color has negligible effect of the overall system cycle times. Fig. 10 demonstrates the crane schedule developed by the program.

TABLE V
COMPARISONS OF CRANES' OPERATING RANGE FOR WHITE COLOR

Crane no.	Current use(plant)	Suggested by the program
1	0-4	0-4
2	3-8	3-8
3	8-19	8-17
4	19-25	17-25
5	24-39	24-39
cycle time(s.)	348	323

The third case involves mixed colors where loads requiring white, brown and black colors are sequenced into the anodizing line. The software used by the plant does not allow this mixed-model type of input, thus there is no comparison available for all mixed color cases. Fig. 11 shows the crane schedule generated by the program. The cycle time obtained from the schedule is 538 s.

The fourth case is used to compare with the third case where loads with white, dark brown and black colors are input to the anodizing line. Crane schedule generated is shown in Fig. 12. Dark brown color requires electro-coloring time of 420 s. The cycle time for the crane is 590 s. Table VI summarizes crane assignment for each case along with resulting cycle times.

The fifth case investigates the case where only dark color coating are mixed. Loads with black, brown and dark brown are sequenced into the production line. As crane schedule is shown in Fig. 13, its resulting cycle time is 954 s., suggesting that all dark colors should not be put into the anodizing line together. Mixing with white color will significantly improve line's cycle time.

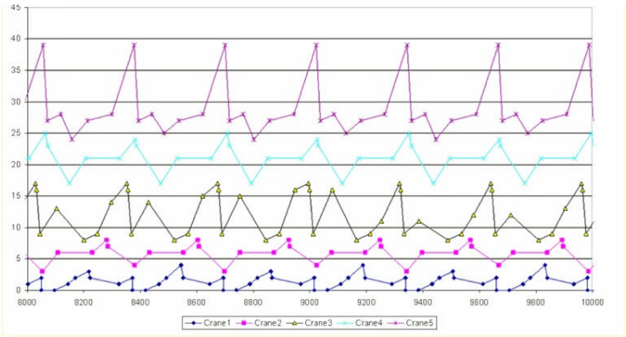


Fig. 10 Crane schedule for white color

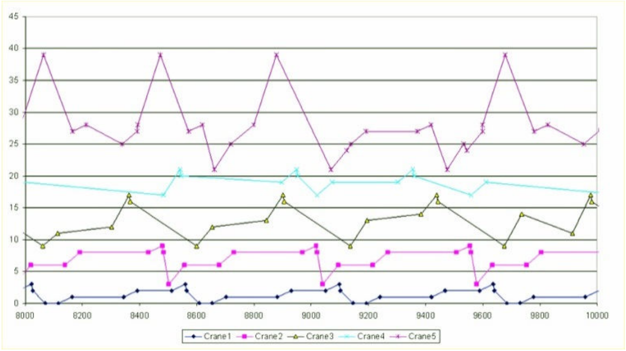


Fig. 11 Crane schedule for white-brown-black colors in sequence

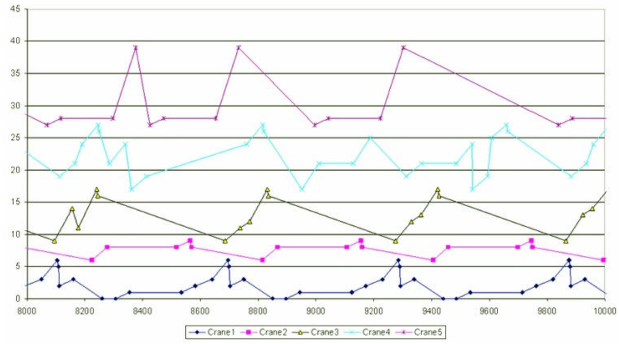


Fig. 12 Crane schedule for white-dark brown-black colors in sequence

TABLE VI
CRANES' OPERATING RANGE FOR MIXED COLORS (SUGGESTED BY PROGRAM)

Crane no.	white-brown-black	white-dark brown-black	black-brown-dark brown
1	0-3	0-6	0-3
2	3-8	6-9	3-12
3	8-17	9-17	11-19
4	17-21	17-27	19-28
5	21-39	27-39	28-39
cycle time(s.)	538	590	954

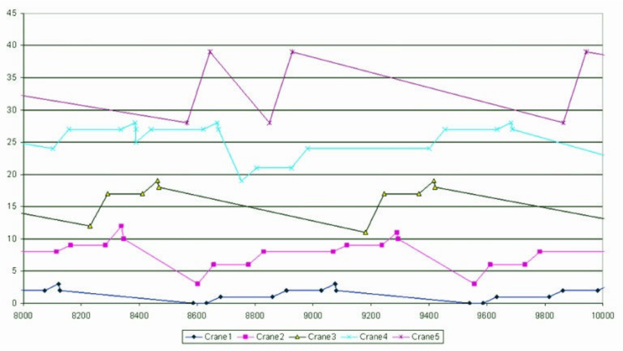


Fig. 13 Crane schedule for black-brown-dark brown colors in sequence

V. Conclusion

In this research, an algorithm to perform an enumerative search for crane schedule was developed. The algorithm searches for individual crane covering range as well as crane schedule. The search assumes time constant between introducing successive loads into the anodizing line. The solution generated is near-optimal, considering 1 second cycle time step search. It has shown great potential to allow mixed model load sequencing in order to identify minimum cycle time.

The algorithm developed can be improved in several ways. Future works include separating search strategy into two steps-rough and fine cycle time search; different bath setups to help balance workload requirements at each process step. Moreover, time between successive load inputs can be non-uniform, allowing loads to be introduced to match crane schedule, thus fully utilizing them.

Acknowledgment

Authors would like to thank Thai Metal Company Limited for providing tremendous supports to this research work.

References

[1] R. W. Lieberman and I. B. Turksen, “Crane scheduling problem,” *AIIE Transaction*, vol. 13, pp. 304-311, 1981.

[2] R. W. Lieberman and I. B. Turksen, “Two-operation crane scheduling problems,” *IIE Transactions*, vol. 14, pp. 146-155, 1982.

[3] L. Lei, and T.-J. Wang, “The minimum-common cycle algorithm for cyclic scheduling of two material handling hoists with time window constraints,” Management Sciences, vol. 37(12), pp. 1629-1639, 1991.

[4] Y. Yih, “An algorithm for hoist scheduling problems,” International Journal of Production Research, vol. 32(3), pp. 501-516, 1994.

[5] P. GE, J. Wang, M. Jin, P. Ren, and H. Gao, “An efficient heuristic algorithm for overhead crane scheduling operations in workshop,” *Applied Mathematics & Information Sciences*, vol. 6-3S, pp. 1087-1094, 2012.

[6] C. A. Floudas, and X. Lin, “Continuous-time versus discrete-time approaches for scheduling of chemical process: a review,” *Computer and Chemical Engineering*, vol. 28, pp. 2109-2129, 2004.

[7] F. Blomer, and H. O. Gunther, “Scheduling of a multi-product batch process in the chemical process industry,” Computer in Industry, vol. 36, pp. 245-259, 2004.

[8] N. G. Hall, C. Sriskandarajah, “A survey of machine scheduling problems with blocking and no wait in process,” Operations Research, vol. 44, pp. 510-525, 1996.

[9] C. F. Daganzo, “The crane scheduling problem,” *Transportation Research*, vol. 23B, pp. 159-175, 1989.

[10] R. I. Peterkofsky, and C. F. Daganzo, “A branch and bound solution method for the crane scheduling problem,” *Transportation Research*, vol. 24B, pp. 159-172, 1990.

[11] K. H. Kim, and K. C. Moon, “Berth scheduling by simulated annealing,” *Transportation Research Part B*, vol. 37, pp. 541–560, 2003.

[12] H. Matsuo, J. S. Shang, and R. S. Sullivan “A crane scheduling problem in a computer integrated manufacturing environment,” Management Science, vol. 37, pp. 587-606, 1991.