

XXI International Conference on Industrial Engineering and Operations Management
9th International Conference on Industrial Engineering and Industrial Management
International IIE Conference 2015
Aveiro, Portugal. July 6-8, 2015

A procedure based on branch-and-bound for the Cyclic Hoist Scheduling Problem with n types of product

Mateo M¹, Manier M-A², Companys R³

Abstract When various kinds of products must receive the same treatments in a production line of tanks and the size of batches is high, a cyclic manufacturing composed of a job from each batch can be scheduled. A hoist ensures the automated transfer of the jobs between tanks. The problem consists in the scheduling of repetitive hoist movements, which is known as CHSP (Cyclic Hoist Scheduling Problem). The objective is to find a sequence which minimizes the cycle time for jobs from different products. We consider the problem where n types of products must be treated and we search an n -cyclic schedule. The algorithm is based on the resolution of different sequences of products. For each one, a branch-and-bound is solved which considers only coherent subsequences. It enables to reduce the computational times most of the time for instances with 5 tanks and 4 product types.

Keywords: Scheduling, branch-and-bound, n -cycle, Hoist Scheduling Problem.

1 Introduction

In some manufacturing systems, transportation resources are the most critical ones and cannot be neglected in the related models. In surface treatment facilities,

¹ Manuel Mateo (e-mail: manel.mateo@upc.edu)
Management Department. ETSEIB Engineering School. Universitat Politècnica de Catalunya.
Av. Diagonal 647, 7th, 08028 Barcelona.

² Marie-Ange Manier (✉ e-mail: marie-ange.manier@utbm.fr)
OPERA - Université de Technologie de Belfort-Montbéliard. Rue Thierry Mieg, 90010 Belfort
cedex

³ Ramon Companys (e-mail: ramon.companys@upc.edu)
Management Department. EPSEB-UPC. Av. Doctor Marañón, 44-50, 3rd, 08028 Barcelona.

chemical processes are performed into tanks. A variety of jobs may be processed, and material handling can be ensured by hoists. This work can be included in the frame of the well-known Cyclic Hoist Scheduling Problem (CHSP).

A set of jobs in a limited number of products ($j=1, \dots, n$) is to be produced and receive one treatment in each one of a set of tanks ($i=1, \dots, m$). Each job j has its own processing time windows, a minimum value $a_{i,j}$ and a maximum value $b_{i,j}$, associated to operation i (in tank i). Jobs are moved along the production line by a hoist, which picks up a job from a tank at any feasible time and transfers it to the next one. This is done from a loading station (tank 0) to an unloading station (tank $m+1$), going through the m tanks. When the hoist arrives at the loading station, it can immediately load a job ($a_{0,j}=0$; $b_{0,j}=\infty$). At the unloading station, similarly the hoist only leaves a job and moves away ($a_{m+1,j}=0$; $b_{m+1,j}=\infty$). A hoist spends idle time above a tank if it arrives earlier than the scheduled instant for getting the job. It cannot pause while moving a job.

We consider the CHSP for lines with one hoist and tanks with capacity for only one job. The loading and unloading stations are considered dissociated; otherwise the model could be easily adapted. In the 1-cycle, the most studied case, one job is introduced into and removed from each tank during each cycle. So, each operation and each hoist move is performed exactly once (n times for an n -cycle). If n jobs are introduced into and n jobs are removed from the line during a cycle, we search for an n -cyclic schedule, where n is the cycle degree. An n -product problem means that n different products are alternatively produced on the line. The purpose of our study is to offer a model for n kinds of products, particularly tested for $n=4$.

Section 2 describes the state of the art for the CHSP. Section 3 presents the n -product n -cyclic problem. Section 4 develops the procedure to solve the problem and Section 5 shows the computational results. Finally, the conclusions of the research are discussed in Section 6.

2 State of the art

The first model for the CHSP was developed by Phillips & Unger (1976). Manier & Bloch (2003) grouped the different works on HSP and classified them. According to their notation, the case dealt can be classified as: CHSP $|m//diss|/n,m+2|C$. It means there is a single hoist; m tanks; *diss* as the loading and unloading stations are dissociated; n kinds of products; $m+2$ is the number of operations of the longest processing sequence; and C is the cycle time to be minimized. Generally, CHSP is NP-complete (Lei and Wang, 1989), even for the simplest variant.

The cyclic sequence is introduced to produce identical jobs (all the jobs of the same product), as Shapiro & Nuttle (1988) described. The hoist infinitely repeats

cyclic movements to treat the input of jobs. The objective is to minimize the cycle time, i.e. the time consumed by the hoist to carry out a complete sequence of movements. Chen et al (1998), among others, have used graphs to solve it.

The simultaneous production of multiple products has been treated by means of several kinds of procedures. For non-cyclic scheduling, Lei and Liu (2001) presented a formal analysis of the HSP with two different products and developed a branch-and-bound procedure to find the optimal schedules. El Amraoui et al (2008) studied the 2-product 2-cyclic HSP and proposed a mixed integer linear program (MILP), which was extended in El Amraoui et al (2012). Three methods (exact and approach ones) are proposed to solve the n -cyclic ($n \geq 2$) HSP for heterogeneous jobs: in El Amraoui et al (2013a), the initial MILP is extended to n -cyclic problems. All the exact models are solved using commercial software CPLEX. El Amraoui et al (2011) proposed a heuristic, in which the cycle degree is a variable. And in El Amraoui et al (2013b), a genetic algorithm is used for the n -cyclic problem if $n \leq 10$. Nevertheless, those two approached methods cannot ensure to find the optimal solution.

This work develops an exact algorithm to solve the n -product n -cyclic problem, taking advantage of coherent constraints to evaluate only feasible sequences and reduce the node generation in a branch-and-bound.

3 The n -product n -cyclic problem and model

Considering n different products, the problem can be solved converting easily the usual m operations of a product (1-cyclic model) into $n \cdot m$ operations (n -cycle), as the tank virtual capacity is multiplied by n . First the concept of stage is defined. A stage in the model is each one of the treatments received by any product in any tank of the line (Mateo and Companys, 2006). Only one of the stages associated to a given tank can be carried out simultaneously. In an n -product n -cyclic schedule, each stage is performed exactly once during a period. In the model, the sequence for product 1 is formed by stages 0, n , $2 \cdot n \dots$; for product 2 formed by stages 1, $n+1$, $2 \cdot n+1 \dots$ and so on, up to product n (stages $n-1$, $2 \cdot n-1, \dots$). Equation (1.1) expresses the index k of any stage according to the corresponding couples of indices (i, j) (tank i and product j):

$$k = i \cdot n + (j-1) \quad k=0, \dots, n \cdot (m+1) + (n-1) \quad \text{for } i=0, \dots, m+1; j=1, \dots, n \quad (1.1)$$

For instance, for $n=2$, the third operation of product 1 corresponds to stage $k=6$. Then using this expression, the model of the problem, in particular the constraints of time windows and hoist movements, can be expressed according to the defined

stages k rather than to product j and tank i , whatever a number n of products (see (1.2) to 1.7)). For this goal, we introduce the following notations:

- k index for stages ($k=0, \dots, n \cdot (m+1) + (n-1)$)
 a_k, b_k minimal and maximum values for the time window at stage k ($0 \leq k \leq n \cdot (m+1) + (n-1)$); $a_k=0$ and $b_k=0$ ($k \geq n \cdot (m+1)$)
 f_k transportation time between stages k and $k+n$ ($k=0, \dots, n \cdot (m+1) - 1$)
 $e_{k,k'}$ empty hoist time from stage k to stage k' ($k, k'=0, \dots, n \cdot (m+1) + (n-1)$)
 c_k 1, if a job arrives and leaves the stage k in different cycles ($0 \leq k \leq n \cdot (m+1) + (n-1)$); 0, otherwise.

Let $\mathbf{H}=(h_{[0]}, h_{[1]}, \dots, h_{[n \cdot (m+1) - 1]})$ be a circular permutation of the full-hoist movements, which indicates the origin stage. Vector $\mathbf{T}=(t_{[0]}, \dots, t_{[n \cdot (m+1) - 1]})$ contains the starting times associated to vector \mathbf{H} . Without loss of generality, $h_{[0]}=h_0$ and $t_{[0]}=0$ and $t_{[0]} < t_{[1]} < \dots < t_{[n \cdot (m+1) - 1]}$. A cyclic sequence (\mathbf{H}, \mathbf{T}) will be feasible if and only if there is a job in the tank to be taken, the destination tank for the job is empty and the processing times fall within the limits of the time windows. C is the time to complete the $n \cdot (m+1)$ movements. A set of $n \cdot (m+1) - 1$ variables come from the vector \mathbf{T} , because t_0 is supposed to be fixed, and an additional variable is the cycle time C . Therefore, the variables are:

- h_k transportation move of a job from stage k ($k=0, \dots, n \cdot (m+1) - 1$)
 t_k ending time of stage k or starting time of movement of a job to next stage ($0 \leq k \leq n \cdot (m+1) + (n-1)$)
 C cycle time to be minimized

And the model is:

$$[\text{MIN}] C \quad (1.2)$$

subject to

$$t_k - t_{k-n} \geq a_k + f_{k-n} - c_k \cdot C \quad k=n, \dots, n \cdot m + (n-1) \quad (1.3)$$

$$t_{k-n} - t_k \geq -b_k - f_{k-n} + c_k \cdot C \quad k=n, \dots, n \cdot m + (n-1) \quad (1.4)$$

$$t_{[k+1]} - t_{[k]} \geq f_{[k]} + e_{[k]+n, [k+1]} \quad k=0, 1, \dots, n \cdot m + (n-2) \quad (1.5)$$

$$t_0 - t_{[n \cdot m + (n-1)]} \geq f_{[n \cdot m + (n-1)]} + e_{[n \cdot m + (n-1)]+n, 0} - C \quad (1.6)$$

$$C \geq 0; t_k \geq 0 \quad k=0, \dots, n \cdot (m+1) - 1 \quad (1.7)$$

The length of the cycle C is to be minimized in (1.2). Constraints (1.3) and (1.4) ensure that the minimal and maximal processing times are respected at each stage k . Constraints (1.5) and (1.6) impose that the hoist movement time. Con-

straint (1.7) indicates that variables are non-negative. We use a graph approach of Chen et al (1998) to represent the values of the variables from vector \mathbf{T} in vertices and arcs for the constraints. For each constraint (1.3 to 1.6), an arc is defined between two of the $(m+1) \cdot n$ vertices, the initial stage and final stage, with has a positive or negative value and sometimes adding or subtracting the variable C .

4 The procedure to solve the n-product n-cyclic problem

4.1 Branch-and-bound procedure given a sequence of products

For the cyclic problem, a branch and bound procedure inspired by Shapiro & Nuttle (1988) is developed. The solutions in the research tree are constructed by adding a new tank i at each level and the associated stages in vector \mathbf{H} .

Nodes. The number of levels in the tree is equal to the number of tanks m . The sequence in the root node is automatically determined. A node v at a level r ($2 \leq r \leq m$) is characterized by a permutation of movements assigned to the first $n \cdot (r+1)$ stages. $\mathbf{H}_{r,[v]} = (h_{[0]}, \dots, h_{[n \cdot (r+1) - 1]})$ corresponds to a permutation such that $h_{[0]} = h_0$ and any $h_w \in \mathbf{H}_{r,[v]}$ must respect the constraints on coherent subsequences. If we consider two consecutive tanks (i and $i+1$) and the hoist movements associated to the stages (from k to $k+m \cdot n - 1$), the order of movements is prefixed due to the general hypothesis of the model. Given a tank i in which stages k and $k+1$ are performed, and the new tank $i+1$ associated with stages $k+n$ and $k+n+1$, they follow the circular permutation $(k, k+n, k+1, k+n+1)$. The relation of hoist movements between these four stages, and the constraint satisfaction on the associated variables, depend on the occupation of tank i by jobs at the beginning of the cycle (Mateo and Amorós, 2002). This can be applied on two consecutive products. This is a condition that ensures the feasibility of the sequence (Lei and Liu, 2001).

For the sub-problem at each node, a graph is developed and solved with a shortest path algorithm. If the node is a leaf and $C(\mathbf{H}_{r,[v]})$ is lower than the cycle time of the best known solution, $\mathbf{H}_{r,[v]}$ and its cycle time will be the new best one.

Branching. If v is a non-leaf node of level $r-1$, the vectors in descendant vertices at level r are obtained by adding n new stages, which correspond to the next tank r . For any value n , n groups of four stages are considered to satisfy the coherent conditions and the new potential descendant vertices are accepted or rejected.

Bounding. Let v be a node at level r (with r tanks) defined by $\mathbf{H}_{r,[v]}$. A lower bound for C , trying to use the idle time for the hoist, can be determined considering the moves included in vector $\mathbf{H}_{r,[v]}$ and, if necessary, an additional time for not assigned moves ($h_i \in \mathbf{U}$). Given $C(\mathbf{H}_{r,[v]})$, the cycle time for $\mathbf{H}_{r,[v]}$; \mathbf{U} , the set of un-

assigned loaded moves in $\mathbf{H}_{r,[v]}$ and w_k , the waiting time for hoist above the stage k , besides more lower bounds obtained from the graph (Chen et al, 1998):

$$\text{LB}(\mathbf{H}_{r,[v]}) = C(\mathbf{H}_{r,[v]}) + \max \left\{ \left[\sum_{h_i \in \mathbf{U}} (f_i + e_{i,i+1}) - \sum_{h_k \in \mathbf{H}} (w_k) \right], 0 \right\} \quad (1.9)$$

4.2 General algorithm

If a 1-product n-cyclic sequence is going to be determined, one option is the addition of n 1-cyclic sequences. Also, if a sequence is searched for an n-product n-cyclic sequence, the initial solution may be obtained solving the 1-product 1-cycle, for a fictitious product with the most restricted time windows at each tank.

For $n > 2$, the first decision is the evaluation of each possible sequence of products \mathbf{J}_{sp} entering the line. Let sp an index for the $(n-1)!$ different sequences in a cycle. For example, 4 products (A,B,C,D) must be produced simultaneously through the line. The first level consists in determining each possible sequence \mathbf{J}_{sp} (ABCD, ABDC, ACBD, ACDB, ADBC, ADCB) and for each of the six sequences, the corresponding branch-and-bound procedure is developed. Moreover, the optimum cycle time for the second sequence could be computed with an upper bound, the optimal cycle time for the first one, and so on for the other sequences.

Algorithm

- Step 1. Obtain data (n ; time windows: a_k, b_k ; hoist times: $f_k, e_{k,k'}$)
- Step 2. 2.1. Define a single virtual product vp with time windows:
 $[a_i, b_i] = [\max_j \{a_{i,j}\}, \min_j \{b_{i,j}\}]$:
 2.2. Solve a branch-and-bound for the this single virtual product
 → Optimal sequence \mathbf{H}^*_0 and optimal cycle time C^*_0
- Step 3. 3.1. $sp=1$; $\mathbf{J}_1=(1, 2, \dots, n)$
 3.2. Given $UB_1=C^*_0$, solve a branch-and-bound for the sequence of products \mathbf{J}_1
 → Optimal sequence \mathbf{H}^*_1 and optimal cycle time C^*_1
 3.3. $\mathbf{H}^* = \mathbf{H}^*_1$; $C^* = C^*_1$
- Step 4. While not all the product permutations are evaluated
 4.1. $sp=sp+1$; $\mathbf{J}_{sp}=(1, [2], \dots, [n])$
 4.2. Given $UB_{sp} = C^*$, solve a branch-and-bound for another sequence \mathbf{J}_{sp}
 → Optimal sequence \mathbf{H}^*_{sp} and optimal cycle time C^*_{sp}
 4.3. If $C^*_{sp} < C^*$
 $\mathbf{H}^* = \mathbf{H}^*_{sp}$; $C^* = C^*_{sp}$
-

5. Computational results

The computational experiments are based on a set of 30 instances corresponding to 4 products and 5 tanks (Mateo and Companys, 2012) and various ranges in the width of time windows and the hoist speed, or the relation between loaded and empty hoist moves. We focused here both on the cycle time and the computing time. The algorithm is written in Visual C++ and run in a Pentium Core2 Quad 2.40 Ghz, 2 Gb RAM. The results on the instances are given in Table 1. Each line provides: the reference number for the instance, the computational time (in seconds) and the optimal value found in El-Amraoui et al (2013a), called as EA13, and then the computational time and optimal cycle time we obtained.

For an instance (209080305), we reach a different solution (ours is 718) as there is a mistake in El-Amraoui et al (2013a); indeed for their announced results, the time windows are violated. We discovered they really solved a mix of two instances (time windows were from instance 207080605). We obtain the optimum (679) of that “mixed” instance in 32 seconds. For the remaining instances, we almost always reduce the computing time and always obtain the optimum. For the four instances with computational time equal to 24 hours, we can guarantee that the best results obtained are optimal ones, in a mean time of 8.4 seconds. Moreover, the mean reduction in time is 291 seconds (nearly 5 minutes).

Table 1 Results (for $n=4$ and $m=5$) obtained with our algorithm compared with El-Amraoui et al (2013)

Instance	Time(s) EA13	Opt EA13	Time(s)	Opt	Instance	Time(s) EA13	Opt EA13	Time(s)	Opt
104040705	690	736	105	736	201080405	29	988	1	988
105040905	785	622	9	622	207080605	279	768	32	768
109040505	235	978	2	978	209080305	371	678	283	718
102040605	59	637	4	637	205080205	341	826	181	826
103040805	178	810	7	810	204080905	624	646	35	646
107070405	440	708	618	708	308060205	821	784	77	784
109070605	271	904	9	904	301060405	86400	1188	7	1188
102070205	98	642	60	642	304060405	86400	1152	2	1152
102070705	140	645	15	645	303060905	990	646	22	646
101070405	154	718	55	718	309060805	86400	1092	4	1092
205050605	171	842	58	842	309090705	86400	1024	79	1024
203050905	29	520	4	520	308090205	977	772	183	772
208050705	55	700	2	700	303090905	192	650	54	650
209050705	100	721	47	721	302090205	309	756	160	756
204050805	91	638	15	638	304090405	86400	1116	2	1116

6. Conclusions

The proposed model relies on the stage definition, the operation performed in any given tank according to the part type. Consequently, multiple stages are assigned to each tank. This model opens new paths to study more complex cyclic sequences, with multi-part types, reduced to procedures considered for single part jobs. The conditions for coherent subsequences limit in a smart way the node generation in the described branch-and-bound. Finally, the study on the n -product n -cycle, with $n=4$, shows the advantage of the use of bounds. As future research lines, the algorithm should be extended to $n>4$ and $m>5$ and evaluate if it can be used to confirm and obtain optimal values for greater size instances.

7 References

- Chen H, Chu C, Proth JM (1998) Cyclic Scheduling of a Hoist with Time Window Constraints. *IEEE Transactions on Robotics and Automation*. 14: 144-152
- El Amraoui A, Manier MA, El Moudni A, Benrejeb M (2008) A mixed linear program for a multi-part cyclic hoist scheduling problem. *International Journal of Sciences and Techniques of Automatic control and computer engineering*, 2: 612-623
- El Amraoui A, Manier MA, El Moudni A, Benrejeb M (2011) Heuristics for the resolution of the cyclic hoist scheduling problem with multi-items. 18th IFAC World Congress, Milano: 8195-8200
- El Amraoui A, Manier MA, El Moudni A, Benrejeb M (2012) Resolution of the Two-Part Cyclic Hoist Scheduling Problem with bounded Processing Times in Complex Lines' Configuration, *European Journal of Industrial Engineering*, 6: 454-473
- El Amraoui A, Manier MA, El Moudni A, Benrejeb M (2013a) A linear optimization approach to the heterogeneous r -cyclic hoist scheduling problem. *Computers & Industrial Engineering*, 65: 360-369
- El Amraoui A, Manier MA, El Moudni A, Benrejeb M (2013b) A genetic algorithm approach for a single hoist scheduling problem with time window constraints. *Engineering Applications of Artificial Intelligence*, 26: 1761-1771
- Lei L, Liu Q (2001) Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints, *INFOR*, 39: 185-199
- Lei L, Wang TJ (1989) A proof: the cyclic hoist scheduling problem is NP-complete. Working paper #89-0016, Rutgers University.
- Manier MA, Bloch C (2003) A classification for Hoist Scheduling Problems. *International Journal of Flexible Manufacturing Systems*. 15: 37-55
- Mateo M, Amorós A (2002) Estudio del caso 2-cíclico para el Hoist Scheduling Problem. Document Intern de Treball 2002/14. Dep. Organització d'Empreses. UPC, Barcelona, Spain
- Mateo M, Companys R (2006) Hoist scheduling in a chemical line to produce batches with identical sizes of different products. 6ème Conférence Francophone MOSIM, Rabat: 677-684
- Mateo M, Companys R (2012) n -Cyclic Hoist Scheduling Problem to manufacture more than 2 products. Research Report. <http://hdl.handle.net/2117/16384>
- Phillips LW, Unger PS (1976) Mathematical Programming Solution of a Hoist Scheduling Program. *AIIE Transactions*. 8: 219-225
- Shapiro GW, Nuttle HW (1988) Hoist Scheduling For A PCB Electroplating Facility. *IEE Transactions*. 20: 157-167