

# Collision-Free Based Model for the Cyclic Multi-Hoist Scheduling Problem

Laajili E.<sup>1</sup>, Lamrous S.<sup>1</sup>, Manier M.-A.<sup>1</sup> and Nicod J.-M.<sup>2</sup>

**Abstract**—In this paper, we propose a Mixed Integer Linear Programming model for solving a hoist scheduling problem with several transportation resources. This model complements initial work that neglected the risk of collisions between hoists. This new model identifies and manages the various possible collision situations, and it is intended to be integrated as a solution evaluation module in a hybrid algorithm addressing the broader and more complex joint problem of sizing transport resources and scheduling surface treatment workshops. In this global approach, an evolutionary algorithm first generates partially feasible solutions, whose total feasibility is then verified a posteriori, thanks to the proposed new model. This model is validated through tests performed on instances of the literature.

## I. INTRODUCTION

Surface treatment lines (Fig. 1) can be found in many industrial sectors (automotive, household appliances, jewellery, etc.). They make it possible to modify the mechanical properties and/or external appearance of many types of parts. In such systems, the movement of parts between processing resources is ensured by automated handling hoists, which most often perform cyclic sequences of alternated loaded moves and empty repositioning. In this case, the challenge is to optimize the line productivity by minimizing the cycle time. However, the number and location of resources is also a lever for improving workshop performance and optimizing costs. The underlying idea is to be able to reconfigure the physical system in the event of series' change, in order to maintain optimal steady-state operation for each new production. In particular, our work focused on transport resources and we have previously studied joint optimization (minimization) of the couple number of hoists and cycle time [1], [2]. To the best of our knowledge, the joint optimization problem was not studied with both latter objectives together, and the idea is to provide a decision support system that enables to choose the best couple of parameters regarding productivity and cost objectives. Nonetheless, the simultaneous use of transport resources leads to a risk of collisions. We initially neglected this risk [1], [2], [3], as have done most studies in the literature. In this paper, we propose a mathematical model integrating the management of collisions between hoists. This model was developed as part of a hybridization with a metaheuristic approach, that is dedicated in particular to

solve the hoist assignment problem. It is tested and proves its effectiveness on partially feasible solutions (without collision treatment) for known benchmarks of the literature.

The organization of this paper is as follows: Section II details the addressed problem. In Section III, a synthetic state of the art situates our work within the literature. Section IV is dedicated to the mathematical model that we develop. The experimental environment and comparative test results are provided in Section V. An analysis of these results is also given in this section. Finally, Section VI concludes this paper and opens perspectives for future work.

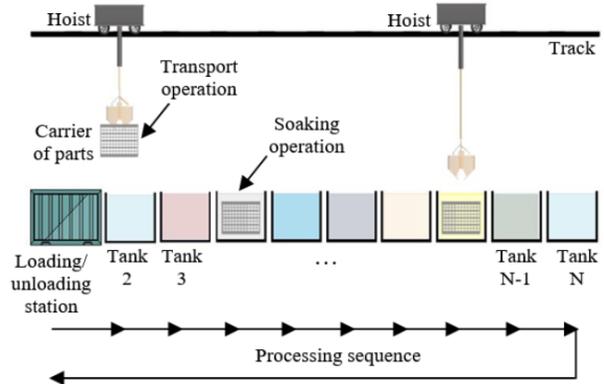


Fig. 1. Example of a treatment surface line [3]

## II. PROBLEM DESCRIPTION

The studied system is an electroplating facility in which parts are processed according to a single sequence of soaking operations in several tanks. Material handling resources called hoists generally move along a single track above the line to transfer parts from tank to tank. In this case, to optimize the performance of such a workshop, the hoists are programmed to achieve cyclic sequences of alternate loaded and empty moves. Each empty move is a repositioning operation between two loaded movements. The problem is known as the Cyclic Hoist Scheduling Problem (CHSP) whose solving aims at minimizing the cycle time (period). A production line is composed of  $n$  tanks (Fig. 1), including a loading and unloading station (tank 1). The tanks from 2 to  $n$  are the soaking tanks. Then a processing sequence is composed of  $n + 1$  operations. In the workshop,  $H$  hoists perform the transfer of parts between processing tanks (from tank  $i$  to tank  $i + 1$ ). In our work, the number of hoists is a variable of the problem that we have a priori determined

<sup>1</sup>Laajili E., Lamrous S. and Manier M.-A. are with Univ. Bourgogne Franche-Comté, FEMTO-ST Institute/CNRS, UTBM, rue Thierry-Mieg, 90010 Belfort Cedex, France (emna.laajili, sid.lamrous, marie-ange.manier)@utbm.fr

<sup>2</sup>Nicod J.-M. is with Univ. Bourgogne Franche-Comté, FEMTO-ST Institute/CNRS, ENSMM, 25000 Besançon, France Jean-Marc.Nicod@ens2m.fr

by an assignment procedure [3] (not detailed here). This one also provides the cyclic move sequence of each assigned hoist, alternating loaded and empty moves. Then, at this stage, those parameters are input data of our problem to solve. The latter consists in determining the starting times of each move, and the minimal common cycle time of the hoists, while respecting various constraints, such as:

- (a) Unitary capacity of tanks and hoists (both are disjunctive resources, they process only one carrier at a time);
- (b) The soaking time in each tank  $i$  is bounded by a minimal and a maximal duration. Then after being processed, a part must leave the tank without any delay to not exceed the maximal time limit;
- (c) It is a no-wait, no-storage problem. First, neither soaking nor transfer operations can be interrupted; second, no intermediate buffer is allowed;
- (d) The hoists must have enough time to travel empty between two successive transport tasks (loaded moves).

A schedule which would not satisfy any of those constraints would be considered as a non-feasible solution.

### III. STATE OF THE ART

The Cyclic Hoist Scheduling Problem (CHSP) [4] was widely investigated in the literature, in both single and multiple hoist cases. Works belonging to the first category are numerous where first, Phillips and Unger [5] had proposed a Mixed Integer Programming Model to maximize the production throughput, and later, many researchers dealt with the problem, using either exact [6], [7], [8], [9] or approximate solving approaches [10]. In the multiple hoist case, obviously, particular constraints must be considered to avoid collisions between the hoists. The problem gets then more complex. Researchers having studied this category, have chosen different methods to solve the problem. Many have opted for the so called “partitioning” or “zoning” approach [11], [12], [13], [14]. It is a heuristic that partitions the line into disjunctive sets of tanks, each one being assigned to a single hoist. This portioning heuristic enables to solve the problem while guaranteeing non-collision between hoists. This assignment heuristic is followed by a resolution of the scheduling problem. Here, Lei and Wang [11] have proposed a “minimum common cycle” algorithm to find the optimal common cycle for both hoists of the line, whereas, Zhou and Li [14] have developed a mixed integer linear programming model. Manier et al. [15] have also investigated a static partitioning approach to solve the CHSP in near real industrial lines (i.e., with multi-function and duplicated tanks). However, their partitioning method may accept that the divided zones overlap. They then identified all possible configurations of collision and applied a branch and bound like algorithm to solve the problem. Others studies have differently coped with the multi-hoist CHSP. Che and Chu [16] have written disjunctive inequalities to formulate the collision-free constraints and used a branch and bound procedure. Leung et al. [17] and Che et al. [18] built up a mixed integer programming formulation to model the problem. In a bi-hoist system, Chtourou et al. [19] have

suggested a collision test procedure to check collisions. Other studies [20], [21] have focused on  $k$ -degree CHSP, where in one cycle,  $k$  identical parts join and leave the production line. Both of them developed a MILP model to solve the problem for the multi-hoist case.

Still in the multi-hoist case, Manier and Lamrous [1], [2] have considered both objectives of minimizing the cycle time and the number of hoists  $H$  assigned to the line. They have proposed a new assignment procedure based on empty moves to generate move sequences for the hoists. They have applied a mixed integer linear programming model to solve the scheduling dimension of the problem, without considering the collision avoidance constraints. In a previous work [3], we have improved this assignment procedure to reach more solutions and broaden the search space. In this paper, we propose to deal with the different scenarios of collision avoidance and incorporate collision-free constraints to the model previously proposed in [1], [2].

## IV. MATHEMATICAL MODEL

### A. Preliminary remarks

In [3], we have developed a hybrid evolutionary algorithm to solve the Cyclic Hoist Design and Scheduling Problem. In this approach, the chromosomes are based on the hoists’ empty moves. Each one represents one solution of the problem and its decoding recognizes the hoists empty moves, the associated move sequence and the number of transportation resources used. Each generated solution is then evaluated. The evaluation consists in solving a Mixed Integer Linear Program (MILP) which provides both the optimal cycle time of the considered scheduling and the associated starting times of each move [2]. In case of non-compliance with model constraints, the evaluation returns the value 0 for the cycle time, which means that the solution is not feasible.

In our initial algorithm, the MILP does not consider the spatial constraints. In this paper, we propose an extended version of the model which includes them so as to guarantee the total feasibility of any resulting solution. These constraints are first inspired from the model of Leung et al. [17] and have been adapted to fit with our choice of variables. Besides, we have also modified and completed them to cope with some special possible situations not previously considered.

For exhaustiveness, we present the complete model (first model without constraints related to collisions’ test [2] together with the new formulated constraints dealing with the collision avoidance between hoists). In the following, we first define the problem parameters.

### B. Notations

- $n$ : number of processing tanks of the line; Tank 1 is the loading-unloading station and tanks from 2 to  $n$  are the soaking tanks, Let  $N = \{1, 2, \dots, n\}$  and  $N - 1 = \{1, 2, \dots, n - 1\}$ ,

For  $i, j \in N$ :

- $(i, j)$ : empty move from tank  $i$  to tank  $j$ ,
- $d_{i,j}$ : duration of the empty move  $(i, j)$ ,

- $r_i$ : duration of the loaded move  $i$ . Move  $i$  is the move where the hoist transports a part from tank  $i$  to its consecutive  $i+1$ ,  $r_i = d_{i,i+1} + c$ , where  $c$  is a constant time needed by the hoist to lift a carrier from tank  $i$ , to pause if necessary above tank  $i$  and let the carrier drip-off, then to stabilize when arrived to tank  $i+1$  and to lower the carrier into tank  $i+1$ . The times  $d_{i,j}$  and  $r_i$  are given constants.
- $m_i$ : minimal soaking time in tank  $i$ ,
- $M_i$ : maximal soaking time in tank  $i$ ,
- $M$ : a very big number to represent the value  $+\infty$ ,
- $H$ : number of hoists associated to the line, deduced after the decoding procedure;
- $h$ : hoist number where  $h \in \{1, 2, \dots, H\}$ ; We assume that hoist 1 is always the responsible of the loaded move 1, so the empty move  $(2, j)$ ,
- $z_h$ : number of empty moves performed by hoist  $h$ , with

$$\sum_{h=1}^H z_h = n,$$

- $v_{h,u}$ : the  $u^{th}$  identified empty move for the hoist  $h$ ;  $u \in \{1, 2, \dots, z_h\}$ ,
- $Moves_h$ : ordered list of transport moves performed by hoist  $h$ . This list associated to each hoist is given by the assignment procedure; if move  $i$  is insured by hoist  $h$ , then  $i \in \{Moves_h\}$ ,
- $p$ : transport move preceding move  $i$ ; if  $i \in \{Moves_h\}$  then  $p \in \{Moves_h\}$ ,
- $Moves_k$ : ordered list of transport moves performed by hoist  $k$ . If hoist  $k$  performs move  $j$ , then  $j \in \{Moves_k\}$ ,
- $q$ : transport move preceding move  $j$ ; if  $j \in \{Moves_k\}$  then  $q \in \{Moves_k\}$ ,
- $z_i^h$ : Boolean data such as:

$$z_i^h = \begin{cases} 1 & \text{if the empty move starting from tank } i \\ & \text{is performed by hoist } h, \\ 0 & \text{otherwise.} \end{cases}$$

Note that

$$\sum_{i=1}^n z_i^h = z_h,$$

The decision variables are the following. For  $i, j \in N$ ,  $h \in \{1, 2, \dots, H\}$  and  $u \in \{1, 2, \dots, z_h\}$ :

- $T$ : time of the cycle period,
- $t_{i,j}$ : starting time of the empty move  $(i, j)$ ; For ease of use, we will write it as  $t_i$ , to refer to the empty move beginning from tank number  $i$ ,
- The Boolean variables  $b_i, a_{h,u}$  and  $y_{i,j}$ , such as:

$$b_i = \begin{cases} 0 & \text{if one carrier is introduced into tank } i \text{ and} \\ & \text{removed from it during the same cycle,} \\ 1 & \text{otherwise.} \end{cases}$$

$$a_{h,u} = \begin{cases} 0 & \text{if the consecutive empty moves } v_{h,u} \\ & \text{and } v_{h,u+1} \text{ are performed by hoist } h \\ & \text{in the same cycle,} \\ 1 & \text{otherwise.} \end{cases}$$

Note that  $v_{h,z_h+1} = v_{h,1}$ .

$$y_{i,j} = \begin{cases} 1 & \text{if the loaded move } i-1 \text{ starts before the} \\ & \text{loaded move } j-1, \\ 0 & \text{otherwise.} \end{cases}$$

### C. Mixed Integer Linear Programming Model

The model that evaluates the feasibility of the hoists' move sequences and then schedules them to find the minimal cycle time in the case of multiple hoists is formulated as a Mixed Integer Linear Programming Model, as follows:

$$\text{Minimize } T \quad (1)$$

Subject to

$$0 \leq t_i \leq T, \quad \forall i \in N, \quad (2)$$

$$t_2 = r_1, \quad (3)$$

$$t_1 + m_{n+1} \leq T, \quad (4)$$

$$m_i \leq T, \quad \forall i \in N, \quad (5)$$

$$(b_i - 1).M \leq t_i - (t_{i+1} - r_i) \leq b_i.M, \quad \forall i \in N - 1, \quad (6)$$

$$(b_n - 1).M \leq t_n - (t_1 - r_n) \leq b_n.M, \quad (7)$$

$$m_i - b_i.M \leq t_{i+1} - r_i - t_i \leq M_i + b_i.M, \quad \forall i \in N - 1, \quad (8)$$

$$m_i + (b_i - 1).M \leq T + t_{i+1} - r_i - t_{i+1} \leq M_i + (1 - b_i).M,$$

$$\forall i \in N - 1, \quad (9)$$

$$m_n - b_n.M \leq t_1 - r_n - t_n \leq M_n + b_n.M, \quad (10)$$

$$m_n + (b_n - 1).M \leq T + t_1 - r_n - t_n \leq M_n + (1 - b_n).M, \quad (11)$$

$$\sum_{u=1}^{z_h} a_{h,u} = 1, \quad \forall h \in \{1, 2, \dots, H\} \quad (12)$$

$$\text{If } h = 1, \quad \forall u \in \{1, 2, \dots, z_1 - 1\},$$

$$a_{1,u} = 0 \text{ and } a_{1,z_1} = 1, \quad (13)$$

$$\forall h \in \{1, 2, \dots, H\}, \quad \forall u \in \{1, 2, \dots, z_h\} :$$

$$t_j + d_{j,i} \leq t_{i+1} - r_i + a_{h,u}.M, \quad (14)$$

$$t_j + d_{j,i} \leq T + t_{i+1} - r_i + (1 - a_{h,u}).M, \quad (15)$$

with  $v_{h,u} = (j, i)$ ,  $v_{h,u+1} = (i+1, l)$  and  $v_{h,z_h+1} = v_{h,1}$ .

$$t_{j+1} - r_j - (t_{i+1} - r_i) \leq M.y_{i+1,j+1}, \quad \forall i, j \in N, i \neq j \quad (16)$$

$$y_{i,j} + y_{j,i} = 1, \quad \forall i, j \in N, i \neq j \quad (17)$$

$$\forall i \in \{Moves_h\}, \forall j \in \{Moves_k\}, \forall h \in \{1, 2, \dots, H\} :$$

$$t_{i+1} + d_{i+1,j} - t_{q+1} - d_{q+1,j} < M.(3 - y_{i+1,j+1} - z_{i+1}^h - \sum_{k=h}^H z_{j+1}^k), \quad k > h, j < i, \quad (18)$$

$$t_{j+1} + d_{j+1,i} - t_{p+1} - d_{p+1,i} < M.(3 - y_{j+1,i+1} - z_{i+1}^h - \sum_{k=h}^H z_{j+1}^k), \quad k > h, j < i, \quad (19)$$

$$t_{i+1} + d_{i+1,j} - t_{q+1} - d_{q+1,j} < M.(3 - y_{i+1,j+1} - z_{i+1}^h - \sum_{k=1}^h z_{j+1}^k), \quad k < h, j > i, \quad (20)$$

$$t_{j+1} + d_{j+1,i} - t_{p+1} - d_{p+1,i} < M.(3 - y_{j+1,i+1} - z_{i+1}^h - \sum_{k=1}^h z_{j+1}^k), \quad k < h, j > i, \quad (21)$$

$$t_{i+1} + d_{i+1,j} - t_{q+1} - d_{q+1,j} - T < M.(2 - z_{i+1}^h - \sum_{k=h}^H z_{j+1}^k), \quad k > h, j < i, \quad (22)$$

$$t_{j+1} + d_{j+1,i} - t_{p+1} - d_{p+1,i} - T < M.(2 - z_{i+1}^h - \sum_{k=h}^H z_{j+1}^k), \quad k > h, j < i, \quad (23)$$

$$t_{i+1} + d_{i+1,j} - t_{q+1} - d_{q+1,j} - T < M.(2 - z_{i+1}^h - \sum_{k=1}^h z_{j+1}^k), \quad k < h, j > i, \quad (24)$$

$$t_{j+1} + d_{j+1,i} - t_{p+1} - d_{p+1,i} - T < M.(2 - z_{i+1}^h - \sum_{k=1}^h z_{j+1}^k), \quad k < h, j > i, \quad (25)$$

$$\forall j \in \{Moves_h\}, \forall i \in \{Moves_k\}, \forall h \in \{1, 2, \dots, H\} :$$

$$t_{j+1} - t_{p+1} - d_{p+1,i} < M.(3 - y_{j+1,i+1} - z_{j+1}^h - \sum_{k=h}^H z_{i+1}^k), \quad k > h, i = j + 1, \quad (26)$$

$$t_{i+1} - t_{q+1} - d_{q+1,j} < M.(3 - y_{i+1,j+1} - z_{j+1}^h - \sum_{k=1}^h z_{i+1}^k), \quad k < h, i + 1 = j, \quad (27)$$

$$t_{j+1} - t_{p+1} - d_{p+1,i} - T < M.(2 - z_{j+1}^h - \sum_{k=h}^H z_{i+1}^k), \quad k > h, i = j + 1, \quad (28)$$

$$t_{i+1} - t_{q+1} - d_{q+1,j} - T < M.(2 - z_{j+1}^h - \sum_{k=1}^h z_{i+1}^k), \quad k < h, i + 1 = j, \quad (29)$$

$$b_i \in \{0, 1\}, \quad \forall i \in N, \quad (30)$$

$$a_{h,u} \in \{0, 1\}, \quad \forall h \in \{1, 2, \dots, H\}, \forall u \in \{1, 2, \dots, z_h\}, \quad (31)$$

$$y_{i,j} \in \{0, 1\}, \quad \forall i, j \in N, \quad (32)$$

Equation (1) represents the objective function that aims at minimizing the cycle time  $T$ . Equations (2) to (15) correspond to the first model proposed in [2]: (2) to (5) define and bound the decision variables  $T$  and  $t_i$ , (6) and (7) define the Boolean variables  $b_i$ , (8) to (11) check the soaking times' bounds constraints, (12) to (15) are associated to the transportation resources for constraints (a) and (d) of part II.

Equations (16) to (29) deal with the collision-free situations: (16) and (17) define the Boolean variables  $y_{i,j}$  and (18) to (29) avoid twelve possible scenarios where collision between hoists may occur. The first eight equations (18) to (25) are inspired from Leung et al. [17]. To exhibit the collision scenarios, we consider the following state of the line. In a multi-hoist case, hoist number 1 will be the nearest hoist to tank 1 as it is responsible for unloading tank 1 and then for the empty move  $(2, j)$ . Each added hoist to the line will stand after its preceding one, until the last hoist, which will be the furthest from tank 1 but the nearest to tank  $n$ . Obviously, collisions may occur when highly numbered hoists must perform transport moves from lowest numbered tanks, or vice versa. Therefore, if we consider two hoists  $h$  and  $k$  and two moves  $i$  and  $j$  (i.e., two tanks  $i$  and  $j$ ),  $h$  performs  $i$  and  $k$  performs  $j$ , two collision scenarios may appear here: either  $(k > h$  and  $j < i)$  or  $(k < h$  and  $j > i)$ . Furthermore, in each of these two scenarios, move  $i$  can be performed before move  $j$  ( $y_{i+1,j+1} = 1$ ) or inversely ( $y_{j+1,i+1} = 1$ ). Thus, the resulting number of scenarios is 4, which are checked in constraints (18), (19), (20) and (21). These same scenarios may also happen when the moves  $i$  and  $j$  are performed in two consecutive cycles. In this case, if move  $i$  is accomplished in the next cycle, then its starting time is increased by the value  $T$  (constraints (23) and (25)). If it is the move  $j$  that starts in the next cycle, its starting time is also increased by  $T$  and these cases are represented in constraints (22) and (24). As an example, Fig. 2 shows the situation considered in the equation (18).

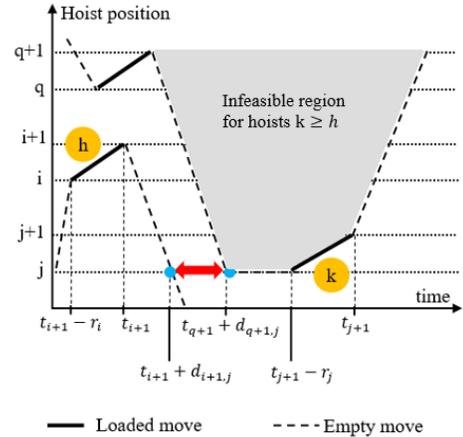


Fig. 2. Collision avoidance scenario corresponding to Eq. (18)

Equations (26) to (29) draw other possible scenarios of collision avoidance that were not addressed by Leung et al. [17]. In fact, the first situation (26) is when  $k > h$ ,  $h$  performs  $j$  and  $k$  performs  $i$  and  $i = j + 1$  (i.e.,  $i > j$  and here the destination tank  $j + 1$  of move  $j$  is the departure tank  $i$  of move  $i$ ). In this case, a collision should be avoided above tank  $i$ . Constraint (27) is wanted to avoid a collision in the situation where  $k < h$ ,  $h$  performs  $j$  and  $k$  performs  $i$  and  $j = i + 1$ . Constraints (28) and (29) investigate two previous scenarios but in the case of two consecutive cycles. Finally equations (30) to (32) give the definition intervals of the binary variables  $b_i$ ,  $a_{h,u}$  and  $y_{i,j}$ .

The model that we propose complements our previous MILP [2] by integrating the identification and management of collisions between hoists. It also improves the model of Leung et al. [17], in the sense that it extends the number of the considered collision situations, in particular when one of the hoists is waiting empty above a tank (Fig. 2). For the sequences corresponding to such cases, our extended model returns the mention not feasible (i.e., a cycle time of value 0). Our constraints (26) to (29) make it possible in particular to avoid boundary collisions, as shown in Fig. 3.

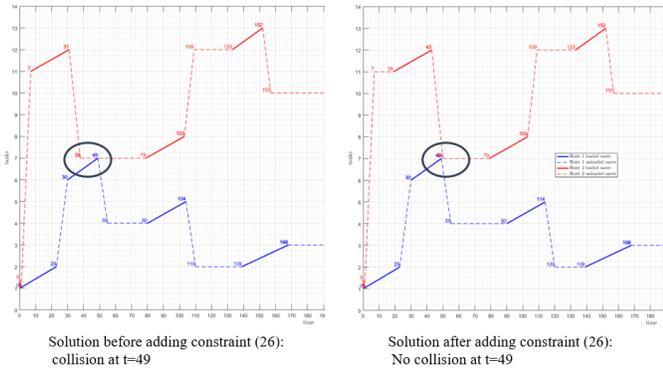


Fig. 3. Collision avoidance when one hoist does not move

## V. COMPUTATIONAL EXPERIMENTS

### A. Experiments

Experimentation was performed on an Intel Core i5-6500 CPU @ 3.20 GHz 3.19 GHz, 16 Go RAM, 64-bit, using the Matlab software under Windows Operating System. Our model was tested on three well known benchmark instances: the first was proposed in [5] that we label Phil, and the two others can be found in [2] that we call Line 1 and Line 2. These instances are composed of 12 or 13 tanks, not including the loading-unloading station. The tests are carried on the feasible solutions that are the outputs of our first proposed approach [3], in which a genetic algorithm combined with the first version of MILP [2] (not dealing with collision avoidance) enabled us to obtain for each possible number of hoists a list of best solutions. Nevertheless, these provided solutions are partially feasible as they satisfy all the constraints except the collision-free ones. Hence, in order to validate the new proposed model integrating the collision-free constraints, we have applied it to these solutions in a

post-verification procedure. Let us note that this one does not depend on the method initially used to generate the partially feasible solutions. But we can obtain good feasible solutions only if we test good initial solutions!. Table I shows an extract of these results (7 solutions) for the three tested instances, for number of hoists from 2 to 5. T1 corresponds to the cycle times got from the first proposed approach without collision treatment [3], T2 are the new cycle times obtained after checking the collision constraints of [17], and T3 are those obtained after applying our new model. We only provide multi-hoist solutions, since applying collision-free constraints to single hoist solutions will not change the resulting schedules. In the same way, we do not report the solutions obtained for more than five hoists, which we found not relevant here.

TABLE I

SOME OF THE BEST SOLUTIONS FOUND WITH AND WITHOUT COLLISION AVOIDANCE, FOR THREE BENCHMARK INSTANCES

instance	T1	T2	T3	instance	T1	T2	T3
Phil 2 hoists	251	251	251	Phil 3 hoists	179	0	0
	252	0	0		182	0	0
	254	0	0		188	0	0
	255	0	0		188	636	0
	257	0	0		188	349	0
	258	0	0		190	0	0
	264	0	0		190.67	492	0
Phil 4 hoists	151	382	0	Phil 5 hoists	151	238	0
	151	398	0		151	264	0
	151	411	0		151	280	0
	151	423	0		151	285	0
	151	579	0		151	303	0
	151	816	0		151	315	0
	151	0	0		151	320	0
Line1 2 hoists	361	361	361	Line1 3 hoists	361	361	399
	361	361	361		361	361	510
	361	367	0		361	361	0
	361	376	0		361	365	0
	361	379	0		361	366	0
	361	382	0		361	369	0
	361	0	0		361	0	0
Line1 4 hoists	361	361	0	Line1 5 hoists	361	361	402
	361	361	0		361	361	0
	361	365	0		361	364	0
	361	367	0		361	366	0
	361	391	0		361	374.5	0
	361	409	0		361	376	0
	361	0	0		361	0	0
Line2 2 hoists	661	661	0	Line2 3 hoists	661	661	0
	661	661	0		661	661	0
	661	667	0		661	684	0
	661	712	0		661	698	0
	661	712	0		661	712	0
	661	796	0		661	744	0
	661	0	0		661	0	0
Line2 4 hoists	661	661	0	Line2 5 hoists	661	661	0
	661	...	...		...	...	...
	661	712	0		661	661	0
	661	0	0				
	661	0	0				

### B. Discussion

In a first step, the conducted tests have enabled us to validate our model, in particular on instances for which we find again feasible solutions. Secondly, the comparison of

the results of table I has helped us to analyze the interest of our a posteriori procedure that checks collision avoidance. Indeed, it shows that many of the best solutions of relaxed problems (without collision management) are not actually feasible (as collisions occur). This is particularly obvious for line 2, where none of the solutions initially selected as the best for 2 to 5 hoists are finally retained. Fortunately, we have obtained the best solution for the single hoist problem. Let us also note that many solutions obtained with Leung et al's model become not feasible after applying our last constraints, as they correspond to the particular identified cases.

In table I, we also observe that for a given instance and a given number of hoists, many solutions obtained have the same initial cycle time  $T_1$ , resulting in a false impression of repetition, because they are actually associated with different solutions in terms of assignment and/or move sequences. Thus, the cycle times 151, 361 and 661 for the three respective studied instances come from the temporal characteristics of the leading machine which is a tank here (because increasing the number of hoists means that transport resources are no longer the critical resources of the system).

Overall, we distinguish three cases: in the first one, the solution obtained by the hybrid metaheuristics remains the same after applying the collision constraints ( $T_3=T_1$ ). The second case is the one already mentioned where a solution becomes a non-feasible one ( $T_3=0$ ). In the third case, the application of collision constraints results in a solution with the same assignment and sequence of movements, but with different execution dates and bigger cycle time ( $T_3>T_1$ ) (see line 1 with 3 or 5 hoists in Table I).

These results show that achieving collision-free solutions is not so trivial. They highlight that in the post-verification process of the generated solutions got from the relaxed problem, there is a significant risk that there is no longer feasible solution among those retained for each number of hoists. This implies either to keep more or all of the best generated solutions after executing the initial algorithm, or to include collision-free constraints in the initial assessment MILP. However, the MILP solving time was already identified as the penalizing part of our previous hybrid metaheuristics [2]. A trade-off between this approach and a posteriori collision check would be to solve the initial MILP for all generated solutions, and just after that apply the collision constraints, but only to a relevant selected set of solutions.

## VI. CONCLUSION

This paper has proposed a new MILP integrated in a hybrid approach and dedicated to address the problem of both sizing and scheduling the transportation resources in a surface treatment workshop. The collision avoidance constraints were integrated in a post verification procedure which allows to guarantee the whole feasibility of the final provided solutions. Several research perspectives are opened up as a result of this work, including the integration of these constraints at specific stages of the resolution algorithm. Constraints could also be improved by allowing one hoist to push itself in order to make way for another hoist. This

implies new additional constraints to verify the temporal feasibility of these atypical movements.

## REFERENCES

- [1] M. A. Manier and S. Lamrous, Design and scheduling of electroplating facilities, in Proceedings - ICSSSM06: 2006 International Conference on Service Systems and Service Management, 2007, vol. 2, pp. 1114-1119.
- [2] M. A. Manier and S. Lamrous, An evolutionary approach for the design and scheduling of electroplating facilities, *J. Math. Model. Algorithms*, vol. 7, no. 2, pp. 197-215, 2008.
- [3] E.Laajili, S. Lamrous, M.A. Manier, and J.M. Nicod, Genetic Algorithm Based Approach for the Multi-Hoist Design and Scheduling Problem, IESM: International Conference on Industrial Engineering and Systems Management, 2019.
- [4] M. A. Manier and C. Bloch, A classification for hoist scheduling problems, *Int. J. Flex. Manuf. Syst.*, vol. 15, no. 1, pp. 37-55, 2003.
- [5] L. W. Phillips and P. S. Unger, Mathematical programming solution of a hoist scheduling program, *AIIE Trans.*, vol. 8, no. 2, pp. 219-225, 1976.
- [6] G. W. Shapiro and H. L. W. Nuttle, Hoist scheduling for a pcb electroplating facility, *IIE Trans.*, vol. 20, no. 2, pp. 157-167, 1988.
- [7] P. Baptiste, B. Legeard, M. A. Manier, and C. Varnier, Optimization With Constraint Logic Programming: The Hoist Scheduling Problem Solved With Various Solvers, Applications of Artificial Intelligence in Engineering (AIENG 93), Vol. 2, pp. 599-614, 1993.
- [8] L. Lei, Determining the optimal starting times in a cyclic schedule with a given route, *Comput. Oper. Res.*, vol. 20, no. 8, pp. 807-816, 1993.
- [9] R. Armstrong, L. Lei, and S. Gu, A bounding scheme for deriving the minimal cycle time of a single-transporter N-stage process with time-window constraints, *Eur. J. Oper. Res.*, vol. 78, no. 1, pp. 130-140, 1994.
- [10] J.-M. Lim, A genetic algorithm for a single hoist scheduling in the printed-circuit-board electroplating line, *Comput. Ind. Eng.*, vol. 33, no. 34, pp. 789-792, 1997.
- [11] L. Lei and T.-J. Wang, The Minimum Common-Cycle Algorithm for Cyclic Scheduling of Two Material Handling Hoists with Time Window Constraints, *Manage. Sci.*, vol. 37, no. 12, pp. 1629-1639, 1991.
- [12] L. Lei, R. Armstrong, and S. Gu, Minimizing the fleet size with dependent time-window and single-track constraints, *Oper. Res. Lett.*, vol. 14, no. 2, pp. 91-98, 1993.
- [13] R. Armstrong, S. Gu, and L. Lei, A greedy algorithm to determine the number of transporters in a cyclic electroplating process, *IIE Trans.*, vol. 28, no. 5, pp. 347-355, 1996.
- [14] Z. Zhou and L. Li, A solution for cyclic scheduling of multi-hoists without overlapping, *Ann. Oper. Res.*, vol. 168, no. 1, pp. 5-21, 2009.
- [15] M. A. Manier, C. Varnier, and P. Baptiste, Constraint-based model for the cyclic multi-hoists scheduling problem, *Prod. Plan. Control*, vol. 11, no. 3, pp. 244-257, 2000.
- [16] A. Che and C. Chu, Single-track multi-hoist scheduling problem: A collision-free resolution based on a branch-and-bound approach, *Int. J. Prod. Res.*, vol. 42, no. 12, pp. 2435-2456, 2004.
- [17] J. M. Y. Leung, G. Zhang, X. Yang, R. Mak, and K. Lam, Optimal Cyclic Multi-Hoist Scheduling: A Mixed Integer Programming Approach, *Oper. Res.*, vol. 52, no. 6, pp. 965-976, 2004.
- [18] A. Che, W. Lei, J. Feng, and C. Chu, An improved mixed integer programming approach for multi-hoist cyclic scheduling problem, *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 302-309, 2014.
- [19] S. Chtourou, M. A. Manier, and T. Loukil, A hybrid algorithm for the cyclic hoist scheduling problem with two transportation resources, *Comput. Ind. Eng.*, vol. 65, no.3, pp. 426-437, 2013.
- [20] X. Li, F. T. S. Chan, and S. H. Chung, Optimal multi-degree cyclic scheduling of multiple robots without overlapping in robotic flowshops with parallel machines, *J. Manuf. Syst.*, vol. 36, pp. 62-75, 2015.
- [21] Y. nian Mao, Q. hua Tang, Z. xiang Li, and L. ping Zhang, Mixed-integer linear programming method for multi-degree and multi-hoist cyclic scheduling with time windows, *Eng. Optim.*, vol. 50, no. 11, pp. 1978-1995, 2018.