

Genetic Algorithm Based Approach for the Multi-Hoist Design and Scheduling Problem

Laajili Emna^{a,b}
emna.laajili@utbm.fr

Lamrous Sid^{a,b}
sid.lamrous@utbm.fr

Manier Marie-Ange^{a,b}
marie-ange.manier@utbm.fr

Nicod Jean-Marc^{a,c}
jean-marc.nicod@ens2m.fr

^aUniv. Bourgogne Franche-Comté, FEMTO-ST Institute/CNRS

^bUTBM, Belfort, 90010, France

^cENSMM, Besançon, 25000, France

Abstract—Electroplating facilities often face the Cyclic Hoist Scheduling Problem when a repetitive sequence of moves is searched for the hoists. This paper addresses this optimization problem extended to the design of the workshop, where we aim to minimize both the cycle time and the number of hoists used. For this goal, we propose a genetic meta-heuristic approach which introduces a novel solution encoding to enlarge the solutions' search space. Our encoding procedure is based on hoists' empty moves, and includes separator characters. With the latter, we obtain solutions that were not reachable by previous approaches. Each solution obtained thanks to the genetic operators is evaluated by using a Mixed Integer Linear Program. This one checks the constraints of the problem (such as capacity constraints and soaking time bounds) and computes the smallest cycle time for a given moving sequence and its associated number of hoists. Some results are presented using benchmark instances for which our approach allows to improve the best known solutions.

Index Terms—Cyclic Hoist Scheduling Problem, Design of Electroplating Facilities, Encoding approach, Genetic Algorithm.

I. INTRODUCTION

Automated material handling devices are extensively used in contemporary manufacturing systems. Like in electroplating facilities, programmable cranes (also called robots or hoists) are responsible for transporting parts from a work station to another. These robots represent one primordial component for the productivity of the line as they must respect restrictive bounded processing times of the parts, while performing transportation tasks. The production system is an automated electroplating line which consists of a number of chemical tanks arranged in a row. These tanks are usually ordered following the processing sequence. The first tank of the line is the loading station where batches of parts are prepared and mounted on carriers and the last tank is the unloading station where carriers are unloaded after being processed. In some cases, both operations of loading and unloading may occur in the same first station which is called associated loading-unloading station. After being loaded, the carriers soak successively in the chemicals following the processing sequence. They must dwell in the tanks within given time-windows, otherwise, the parts are considered defective. At the end of every soaking operation, the carriers are removed and transferred to the next tank via a material handling hoist. An electroplating line can have one or several operational hoists.

These programmable robots move on a single track fixed above the tanks line (Fig. 1).

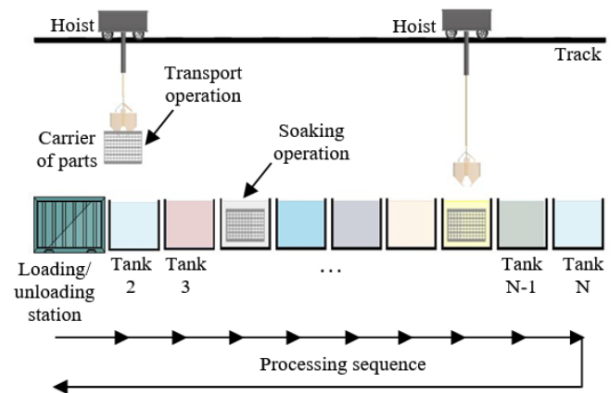


Fig. 1. An electroplating line with associated loading-unloading station

This paper considers the design and scheduling problem in electroplating facilities where the production is cyclic (Manier and Lamrous [1]). We call it the Cyclic Hoist Design and Scheduling Problem (CHDSP). Design means that we search for the convenient number of hoists that fulfills the production and cost objectives like enhancing productivity, lowering investment and energy costs, etc. For this reason, we solve the problem for different hoist fleet sizes. Scheduling is wanted to order the hoists' moves while respecting processing and time-window constraints. It aims at minimizing the cycle period. To solve this problem, we propose a genetic algorithm based approach. The assignment of a hoist number to the line with a respective moving sequence to every hoist is done due to an encoding approach based on empty moves. This one extends the initial encoding proposed by Manier and Lamrous in 2006 [1]. An MILP model is used to evaluate the feasibility of the moving sequences and to compute the period of the cycle.

The paper is organized as follows: Section 2 describes the problem and introduces the notation used. Section 3 gives a state of the art to the handled problem. In Section 4, we present our genetic solving approach, whose first step is an assignment phase based on the empty moves and the separator characters.

Experiments and results are reported and discussed in Section 5 and, finally, Section 6 draws a conclusion to this work.

II. PROBLEM DESCRIPTION

In electroplating systems, a great number of parts often go through the same processing sequence. As such, a fixed processing sequence is repeated constantly and then, the production is considered cyclic. The duration of the cycle is the cycle period or the cycle time. The hoists, responsible for the transport operations, will also repeat the same moving sequences. A hoist cyclic schedule is the cyclic move sequence composed of transfer operations for which we determine the starting and the ending times. Hence, the CHDSP we address in this study intends to find the cyclic schedule that minimizes the cycle period for each number of hoists that can be assigned to the line. As such, we will get several best possible couples of fleet size and cycle period, together with their associated schedules. These outcomes represent a decision support system that will enable decision-makers to choose a suitable couple of parameters to their production systems with regards to investment costs and productivity objectives.

The production line that we consider is composed of N tanks (Fig. 1). The first (fictive) tank is an associated loading-unloading station. The tanks from 2 to N are the soaking tanks. Then $N + 1$ operations are performed in the processing sequence: $N - 1$ in chemical tanks and 2 in tank 1 (the first (loading) and the last (unloading) operations). The line consists of H hoists which ensure the transfer of carriers between processing tanks ($h \in \llbracket 1, H \rrbracket$). This fleet size H is a variable of the problem. The repetitive move sequence of hoist h is denoted by S_h . It is an alternation of loaded and unloaded moves. In the first type, the hoist transfers a carrier from one tank i to its consecutive $i + 1$. In the second type, the hoist travels empty from tank $i + 1$ to a tank j , from which the next loaded move starts. We refer to loaded moves by $(i, i + 1)$, to unloaded moves by (i, j) , to unloaded move times by $d_{i,j}$ and to loaded move times by r_i ($i, j \in \llbracket 1, N \rrbracket$). $r_i = d_{i,i+1} + c$, 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. We denote by T the cycle time (period). Both the tanks and the hoists are disjunctive resources. They can only process one carrier at a time. Besides, the hoists need enough time to travel empty between two successive transportation tasks. They evenly must not pause during transport operations unless it is a dripping pause of the carrier. In other respects, the carriers must follow the same processing sequence and visit every tank just once. Once started, the soaking of a carrier cannot be interrupted. Furthermore, the soaking time in each tank i must range between a minimum and a maximum time bounds, denoted respectively by m_i and M_i . After being processed, a carrier must leave the soaking tank without any delay to not exceed the maximum time limit. Once it has left, it must immediately

be transferred to the next tank because intermediate buffer is not allowed.

III. STATE OF THE ART

The scheduling dimension of the CHDSP is known as the Cyclic Hoist Scheduling Problem (CHSP). It consists in searching the cyclic schedule of hoists' moves that minimizes the cycle period. CHSP is one variant of the general Hoist Scheduling Problem (HSP) for which Manier and Bloch [2] have identified four big classes: Cyclic "CHSP", Predictive "PHSP" [3], [4], Dynamic "DHSP" [5], [6] and Reactive "RHSP" [7], [8]. The authors have also provided a general notation to the HSP Based on it, the suitable notation of the handled problem is the following: $CHSP|H, N-1, 1//ass|N+1|(T_{min}, H_{min})$. It is the cyclic hoist scheduling problem in a single basic line with H hoists, $N - 1$ single capacitated tanks, plus the associated loading and unloading stations, a processing sequence of $N + 1$ operations (including the loading and the unloading operations), in a bi-objective optimization problem (minimize both the period T and the fleet size H).

The CHSP was broadly studied in the literature with different production specifications and physical system parameters and solved with various approaches. Phillips and Unger [9] were the first to investigate it within a single hoist line. They considered the objective of throughput maximization and proposed a Mixed Integer Programming Model (MILP) to solve the problem. Subsequently, many other researchers have dealt with the CHSP in the single hoist case. Most of them proposed exact solving approaches like linear programming models [10], constraint logic programming models [11], binary search procedures [12], branch and bound solution procedures [13], etc. In the range of approximate approaches, Lim [14] suggested a genetic algorithm based procedure to solve the CHSP. As for the multiple hoist case, the CHSP has also stirred up the attention of researchers. Many of them assumed a zoning approach that divides the line into a number of disjunctive sets of tanks and then, each hoist is assigned to a set ([15]–[18]). Likewise, they fulfilled the collision free constraint because the partitioned zones are not overlapped. Other researchers have treated the multi-hoist case differently. Hanen and Munier [19] assigned the transport moves arbitrarily to the hoists and advanced a MILP model and a branch and bound procedure to solve the problem. Manier et al. [20] investigated the CHSP in systems with duplicated and multi-function tanks that are near to real industrial environments. The authors suggested a static partitioning method that assigns the transfer operations to the hoists, so that the hoists' zones may overlap and the tanks inside may not be adjacent. Che and Chu [21] formulated the collision free constraints as disjunctive inequalities and examined two additional properties that inspect collisions for all the feasible solutions. To solve the problem, they introduced a branch and bound procedure. Zhou and Liu [22] studied the CHSP in bi-hoist systems with overlapping hoist zones and introduced a heuristic algorithm to generate and assign moving sequences to the hoists. They developed a linear programming model with the collision avoidance constraints to find the

optimal schedule among the feasible assignments. Chtourou et al. [23] tackled also the CHSP in a bi-hoist system and proposed a collision test procedure to check collisions on the generated moving sequences.

Other recent studies have focused on the multi-degree or k -degree aspect. In a k -degree CHSP, k identical parts enter and leave the production line in one cycle. Li et al. [33] and Mao et al. [34] developed a MILP model to solve the problem also in the multi-hoist case. In [33], systems with parallel machines were considered. The CHSP was also studied with other optimization objectives, like in the study of Xu et al. [24] where they investigated the waste minimization for an environment friendly electroplating line. Liu et al. [25] also advanced a triple objective model where they focused on productivity maximization, energy saving and freshwater minimization. Thereby, within the intensive studies of the CHSP, the design dimension was rarely examined and each time with a different vision. For Zhao et al. [26], it concerned the arrangement of the production line to optimize the spatial allocation of the tanks. Qu et al. [27] treated the simultaneous design and operation problem of CHSP in 2-Dimension line structures where they supposed that the production line is a compact 2-D structure rather than 1-D line. When the design dimension has been deployed as the objective of minimizing the number of hoists to be used in the line [16] [17], the studies only focused on this objective and did not consider the minimization of the cycle time.

Hence, to our best knowledge, the CHDSP with both objectives of minimizing the fleet size and minimizing the cycle time has only and first be tackled by Manier and Lamrous [1] and [28]. In these studies, the fleet size was a variable of the problem. The authors introduced a novel encoding approach of the solutions based on the hoists' empty moves. For each generated solution, a decoding procedure gives the fleet size together with the cyclic move sequence of each hoist. The scheduling dimension was formulated with a Mixed Integer Linear Programming model and the whole problem was solved with an evolutionary algorithm. Nonetheless, the encoding approach, as it was initially proposed, does not represent the whole search space and then, is not able to reach all the possible solutions. As a result, in this paper, we address the CHDSP with the aim to upgrade the empty move based encoding approach. We will show its limit and propose an adjustment to overtake it.

IV. CHDSP SOLVING APPROACH

A. The Empty move based assignment

The empty move based encoding approach [1] is a novel and interesting approach to encode the solutions for the CHDSP. Previous studies have always used the hoist loaded moves to encode the solutions. Due to the empty move based encoding, the solutions can represent many hoist moving sequences S_h whose number is equal to the fleet size H . Then, each solution corresponds to a number of hoists that can be integrated to the production line. Hence, the approach provides at the same time a couple of information: the fleet size H and the

moving sequence S_h of each hoist. The approach principle is the following: A solution of the problem is any possible combination of tank numbers i ($i \in \llbracket 1, N \rrbracket$). It is encoded as a list of these numbers, that we denote by L . The size of the list is S ($2 \leq S \leq N$). Every two successive numbers i and j ($i, j \in \llbracket 1, N \rrbracket$) of the list represents an empty move (i, j) between the tanks i and j . The number of decoded empty moves from a list is equal to its size S . Hence, the decoding should be cyclic and the last decoded empty move should begin with the last tank number of the list and ends with its first tank number. In general, the number of empty moves corresponding to a solution should be equal to N , to coordinate with all the N loaded moves and to ensure that all soaking operations are filled. However, sometimes the hoists, after performing a loaded move $(i, i + 1)$, should wait above the same tank $i + 1$, at least during the soaking time, before performing the next loaded move $(i + 1, i + 2)$. Here, the empty move $(i + 1, i + 1)$ has a duration but has not a spatial component, so it is called "fictive empty move". Then, if fictive empty moves exist in a solution, they are recognized from the absent associated tank numbers in its list (as $S \leq N$). They are decoded as the couples of the same absent tank number. In each couple, an absent tank number figures twice. That is, if $\{ijk\}$ is a list of a CHDSP solution in a production line composed of 4 tanks ($N = 4$) and l is the absent tank number, the decoding of this list provides four empty moves: (i, j) , (j, k) , (k, i) and the fictive one (l, l) . Once the decoding of empty moves from a list is done, the associated moving sequences can easily be inferred because, the moving sequence alternates between loaded and unloaded moves and loaded moves occur between two successive tanks. That is, always with the list example $\{ijk\}$, the empty move (i, j) is preceded by the loaded move $(i - 1, i)$ and followed by $(j, j + 1)$, the empty move (j, k) is preceded by the loaded move $(j - 1, j)$ and followed by $(k, k + 1)$, and so forth. Fig. 2 shows the outcome of the decoding procedure of the list $L = \{1, 4, 2, 6\}$ in a six-tank problem ($N = 6$). The associated empty moves are $(1, 4)$, $(4, 2)$, $(2, 6)$, $(6, 1)$ with two fictive empty moves $(3, 3)$ and $(5, 5)$. The alternation of loaded and unloaded moves results in two cyclic moving sequences which means that this solution example represents a line with two hoists ($H = 2$).

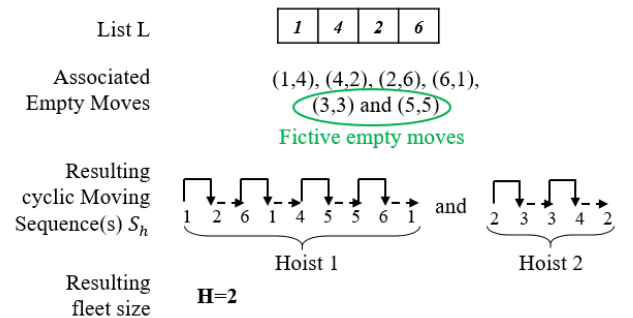


Fig. 2. Decoding procedure of a list example

Nevertheless, the latter described approach, as defined above, is not able to draw all possible solutions of the search space. We prove this fact using a benchmark instance found in [9]. We illustrate it on Fig. 3 whose first part gives the optimal cyclic moving sequence for one hoist. We then can identify the empty moves associated to this solution from which we can infer the empty move based encoded list. However, the deduced empty moves cannot be represented in a single list but in fact, in two separated lists representing two different cycles (for the example in Fig. 3 we obtain two sub-lists $\{2, 11, 9, 10, 4, 8, 3, 13\}$ and $\{7, 12, 5, 1\}$). As a result, the initial empty move based encoding in [29] fails to represent this optimal solution. More generally, it involves that some solutions of the search space were not achievable using this first approach.

To overtake this problem, and with the aim to upgrade the same current approach, we have noticed that we can introduce separators to gather the separated lists in only one list. To translate this idea in the previous encoding approach, we propose to encode a separator as the number zero. It links in between the separated lists corresponding to the same solution. As such, due to the improved approach, we can represent the optimal solution deployed in Fig. 3 in one single list ($\{2, 11, 9, 10, 4, 8, 3, 13, 0, 7, 12, 5, 1\}$). Thereby, with the separator idea, we should redefine some elements of the approach. The list L of a solution to the problem is any possible combination of tank numbers i ($i \in \llbracket 1, N \rrbracket$) that could contain some zeros to refer to separators. The size of the list is always S , where S is the sum of the number of tank numbers i (denoted by TN) and the number of zeros contained in this list (denoted by SN). Thus, $S = TN + SN$. To have an upper bound of the size S , we limit the number of separators that could be integrated in a solution to a maximum allowed number of separators that we denote by NS_{max} . The latter depends on the problem case solved ($2 \leq S \leq N + NS_{max}$ and $NS_{max} = \lfloor \frac{N-2}{2} \rfloor$). The decoding procedure remains the same unless we meet a separator, the decoding should be cyclic. That is, in a sub-list of a list, delimited by two separators, the last decoded empty move begins with the last tank number of the sub-list (that precedes the second separator) and ends with its first tank number (that follows the first separator). Hence, the extended approach allows to reach more solutions in the search space as it enables to represent all the possible combinations of empty moves due to the integrated separators. In fact, if we consider a list as a graph of empty moves, a solution encoded with the first approach is associated to a graph with a unique circuit. However, with the improved approach, a solution can be associated to a graph with one or more circuits. We point out this fact in Fig. 4. We draw the associated graphs of the list corresponding to the optimal solution studied in Fig. 3 and the same list without the separator.

B. The scheduling dimension

Once the generated lists are decoded, the corresponding fleet size H and moving sequences S_h are deduced. Then,

we evaluate each associated solution. The evaluation is done due to the same Mixed Integer Linear Programming Model (MILP) proposed by Manier and Lamrous [28]. If the schedule is feasible (i.e., if all constraints are respected), this evaluation provides the resulting period T for each moving sequence S_h decoded from the list L . Hence, for each generated list L , we get the triplet (H, S_h, T) .

C. The Genetic algorithm based resolution

Genetic Algorithms (GA) [30] [31] are meta-heuristic methods inspired from the Darwinist principle of the evolution theory. It is a population-based stochastic algorithm that performs a random search by crossovers and mutations between individuals within the population. Non-dominated sorting genetic algorithm proposed by Deb and al. in [32] is a multi-objective genetic algorithm originally designed to determine an optimum Pareto. The algorithm is based on a rather classical mechanism of genetic algorithm. It is an elitist algorithm that makes a selection by tournament so as to choose individuals to cross and mutate to generate new populations.

In our case study, the problem is a bi-objective one. We aim to minimize both the cycle time and the number of hoists. The number of feasible solutions of one hoist ($H = 1$) being commonly low, we have tried to start our GA with at least one feasible solution of this configuration among all the solutions of the initial population. The improvement of the Pareto front must operate globally on all configurations, mono and multi hoists. Indeed, if all solutions of one hoist are tested as not feasible, the cycle time for this configuration is defined identically to a large value, to simulate the infinity. The improvement process, hence, in this case, becomes purely random.

GA follows here the same operations as the algorithm published in [32]. The novelty in this paper brings up a new type of non-digital gene that is the separator, as explained in the first part of this paper. The manipulation of the separators operates in the same way as for the integer type genes. The mutation operator allowing either to switch two genes, to add an absent gene, or simply to delete a gene randomly, can bring out solutions without separators.

The crossover probability is 0.8. This represents the fraction of the population at the next generation, not including elite children, that the crossover function creates. The remaining of the population is completed by mutation to provide the same population size at each generation. To ensure the process of maintaining distances between individuals, we use crowding distance. This distance aims at preserving diversity within the population. For this, it assigns to each individual a rank according to the proximity of its score to individuals with the closest scores; the more distant an individual is, the better is its rank. This distance is calculated according to the scores of the individuals for the objective functions.

After defining the initial population of 100 individuals integrating at least one feasible solution of a single hoist, GA operates as follows: (1) Select the parents for the next generation using the current population selection function. The

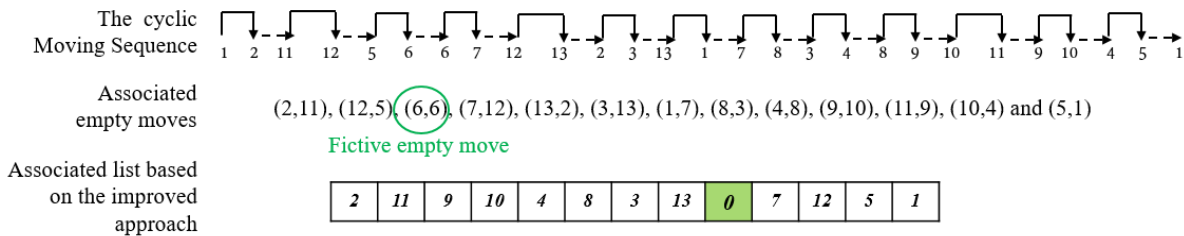


Fig. 3. Representation of the single-hoist optimal solution for the Phillips and Unger's instance [9]

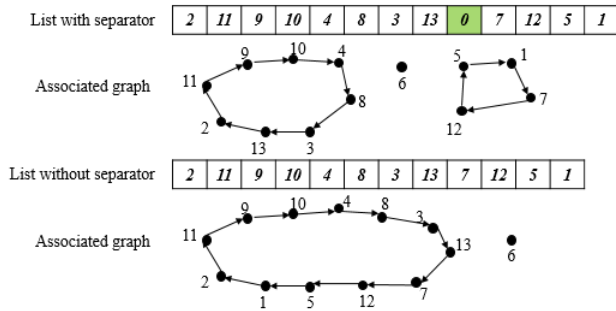


Fig. 4. Multi-circuit graphs with or without separator improvement approach

only integrated selection function is the tournament. (2) Create children from selected parents by mutation and crossover. (3) Score the children by calculating their objective function values and their feasibility. (4) Combine the current population and children in a matrix, the extended population. (5) Compute the rank and crowding distance for all individuals in the extended population. (6) Trim the extended population to have the appropriate number of individuals of each rank.

The algorithm stops when the condition of no evolution results over 100 successive generations is true.

V. EXPERIMENTS AND RESULTS

Experimentation is performed on an Intel® Core™ i7-3770 CPU @ 3.40 GHz, 8GB RAM, 64-bit using Matlab under Windows Operating System. For each instance we performed 20 executions. The average execution time for each instance is 145 min. Compared to [28] with the same material conditions, population sizes and stopping conditions, the execution time is almost halved. This reduction would be explained by the direct resolution of the problem with a bi-objective algorithm. A Pareto front containing solutions is reached faster than in [28]. The computation time of the initialization phase represents 70% of this time. Indeed, it is not easy to find a feasible solution of one hoist.

Our proposed genetic algorithm including separators in the encoding procedure is named SGA in the following. This algorithm has been tested on three industrial benchmarks of the literature, found in [9] (instance named *Phil* here) and in [29] (instances named *Ligne1* and *Ligne2*). These benchmarks include 12 or 13 tanks and various specific characteristics which makes them rather representative of the existing facilities. A

more complete description of these instances, as well as the associated data, can be found in [28]. In table I, we provide the results that we obtained for these instances, in terms of best and mean cycle times, found for each configuration of hoists from 1 to 7. We compare them with the results of the genetic algorithm in [28] (named GA in the table), and also with the best known cycle times obtained with exact methods solving either the single hoist scheduling problem, or sometimes, the two-hoist scheduling problem.

The first main result is that, with SGA, our encoding enables us to represent all the solutions, which can be seen for instance *Phil* and for which we found the optimal single hoist cycle time 521 (not reachable by using GA without separators). We also improved solutions for 2, 3 and 4 hoists, and we obtained the same ones for more hoists. For instances *Ligne1* and *Ligne2*, we found the same best results for 2 hoists and more. The obtained cycle times (361 and 661) are involved by the dominating tank which has the greatest minimal soaking time and which has become the critical resource of the line. For one hoist, the cycle time obtained by SGA is worse than GA for *Ligne1*, but better than GA for *Ligne2* as we found the optimal solution. So globally, we have shown that SGA improves the previous approach GA, as it most often obtains better values of the best cycle time. Moreover, SGA is able to reach optimal solutions in some cases. However, one drawback of SGA is the robustness of the method as we obtain bigger mean cycle time (more variations among the replications). This point should be improved.

VI. CONCLUSION AND PERSPECTIVES

This paper has proposed a valuable extension of the method previously proposed in [28]. It consists of a performing hybrid genetic algorithm with an original encoding procedure for a workshop scheduling problem addressing both the sizing of the transportation resources and the scheduling of the tasks. As a further work, as well as improving the stability and performance of our proposed algorithm, collision free constraints could be included within a subsequent procedure that checks collisions on the best obtained feasible solutions.

REFERENCES

- [1] M.A. Manier, and S. Lamrous, "Design and scheduling of electroplating facilities." International Conference on Service Systems and Service Management, vol. 2, pp. 1114-1119, IEEE, 2006.

TABLE I
RESULTS USING BENCHMARK INSTANCES

Benchmark	<i>Phil</i>					<i>Ligne1</i>					<i>Ligne2</i>				
	T_{opt}^*	GA		SGA		T_{opt}	GA		SGA		T_{opt}	GA		SGA	
		T_b^{**}	T_m^{***}	T_b	T_m		T_b	T_m	T_b	T_m		T_b	T_m	T_b	T_m
1	521	665	673.8	521	1024	425	452	677	492	1032	712	858	1294	712	1332
2	251	332	346	264	360	—	361	402	361	404	—	661	662	661	662
3	—	196	205.8	192.33	231.5	—	361	361	361	361	—	661	661	661	661
4	—	210	226.4	163	207	—	361	361	361	361	—	661	661	661	661
5	—	151	159.4	151.33	154.14	—	361	361	361	361	—	661	661	661	661
6	—	—	—	151	—	—	361	361	361	361	—	661	661	661	661
7	—	151	151	151	151	—	361	361	361	361	—	678	678	661	661

* T_{opt} : The optimal cycle time from the literature, ** T_b and *** T_m : The best cycle time and the mean cycle time over 10 runs.

- [2] M.A. Manier, and C. Bloch, "A classification for hoist scheduling problems.", International Journal of Flexible Manufacturing Systems, vol. 15(1), pp. 37–55, Springer, 2003.
- [3] C. Varnier, and P. Baptiste, "A CLP approach for finding a transition schedule between two cyclic mono-product productions in electroplating facilities.", International Conference on Industrial Engineering and Production Management, pp. 372–381, 1995.
- [4] C. Caux, and H. Pierreval, "Solving a hoist scheduling problem as a sequencing problem.", IFAC Proceedings Volumes, vol. 30(19), pp. 315–319, Elsevier, 1997.
- [5] J. Lamothe, C. Thierry, and J. Delmas, "A multihoist model for the real time hoist scheduling problem.", CESA'96 IMACS Multiconference: computational engineering in systems applications, pp. 461–466, 1996.
- [6] A. El Amraoui, and M. Elhafsi, "An efficient new heuristic for the hoist scheduling problem.", Computers & Operations Research, vol. 67, pp. 184–192, Elsevier, 2016.
- [7] A. Thesen, and L. Lei, "An expert scheduling system for material handling hoists.", Journal of Manufacturing Systems, vol. 9(3), pp. 247–252, Elsevier, 1990.
- [8] Y. Yih, and C. Chiu, "Incremental Learning for Hoist Scheduling Problems in Circuit Board Electroplating Lines.", ASME-PUBLICATIONS-PED, vol. 65, pp. 119–119, ASME, 1993.
- [9] L.W. Phillips, and P.S. Unger, "Mathematical programming solution of a hoist scheduling program.", AIIE transactions, vol. 8(2), pp. 219–225, Taylor & Francis, 1976.
- [10] G.W. Shapiro, and H.L. Nuttle, "Hoist scheduling for a PCB electroplating facility.", IIE transactions, vol. 20(2), pp. 157–167, Taylor & Francis, 1988.
- [11] 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, vol. 2, pp. 599–614, 1993.
- [12] L. Lei, "Determining the optimal starting times in a cyclic schedule with a given route.", Computers & operations research, vol. 20(8), pp. 807–816, Elsevier, 1993.
- [13] 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.", European Journal of Operational Research, vol. 78(1), pp. 130–140, Elsevier, 1994.
- [14] J.-M. Lim, "A genetic algorithm for a single hoist scheduling in the printed-circuit-board electroplating line.", Computers & industrial engineering, vol. 33(3-4), pp. 789–792, Elsevier, 1997.
- [15] L. Lei, and T.-J. Wang, "The minimum common-cycle algorithm for cyclic scheduling of two material handling hoists with time window constraints.", Management Science, vol. 37(12), pp. 1629–1639, INFORMS, 1991.
- [16] L. Lei, R. Armstrong, and S. Gu, "Minimizing the fleet size with dependent time-window and single-track constraints.", Operations Research Letters, vol. 14(2), pp. 91–98, Elsevier, 1993.
- [17] R. Armstrong, S. Gu, and L. Lei, "A greedy algorithm to determine the number of transporters in a cyclic electroplating process.", IIE transactions, vol. 28(5), pp. 347–355, Taylor & Francis, 1996.
- [18] Z. Zhou, and L. Li, "A solution for cyclic scheduling of multi-hoists without overlapping.", Annals of Operations Research, vol. 168(1), pp. 5–21, Springer, 2009.
- [19] C. Hanen, and A. Munier, "Periodic scheduling of several hoists.", Proceedings of the Fourth International Workshop on Project Management and Scheduling, pp. 12–15, 1994.
- [20] M.-A. Manier, C. Varnier, and P. Baptiste, "Constraint-based model for the cyclic multi-hoists scheduling problem.", Production Planning & Control, vol. 11(3), pp. 244–257, Taylor & Francis, 2000.
- [21] A. Che and C. Chu, "Single-track multi-hoist scheduling problem: a collision-free resolution based on a branch-and-bound approach.", International Journal of Production Research, vol. 42(12), pp. 2435–2456, Taylor & Francis, 2004.
- [22] Z. Zhou, and J. Liu, "SA heuristic algorithm for the two-hoist cyclic scheduling problem with overlapping hoist coverage ranges.", IIE Transactions, vol. 40(8), pp. 782–794, Taylor & Francis, 2008.
- [23] S. Chtourou, M.-A. Manier, and T. Loukil, "A hybrid algorithm for the cyclic hoist scheduling problem with two transportation resources.", Computers & Industrial Engineering, vol. 65(3), pp. 426–437, Elsevier, 2013.
- [24] Q. Xu, and Y. Huang, "Graph-assisted cyclic hoist scheduling for environmentally benign electroplating.", Industrial & engineering chemistry research, vol. 43(26), pp. 8307–8316, ACS Publications, 2004.
- [25] C. Liu, C. Zhao, and Q. Xu, "Integration of electroplating process design and operation for simultaneous productivity maximization, energy saving, and freshwater minimization.", Chemical engineering science, vol. 68(1), pp. 202–214, Elsevier, 2012.
- [26] C. Zhao, J. Fu, and Q. Xu, "Production-ratio oriented optimization for multi-recipe material handling via simultaneous hoist scheduling and production line arrangement.", Computers & Chemical Engineering, vol. 50, pp. 28–38, Elsevier, 2013.
- [27] H. Qu, S. Wang, and Q. Xu, "Simultaneous 2D hoist scheduling and production line design for multi-recipe and multi-stage material handling processes.", Chemical Engineering Science, vol. 167, pp. 251–264, Elsevier, 2017.
- [28] M.-A. Manier, and S. Lamrous, "An evolutionary approach for the design and scheduling of electroplating facilities.", Journal of Mathematical Modelling and Algorithms, vol. 7(2), pp. 197–215, Springer, 2008.
- [29] M.-A. Manier, "Contribution à l'ordonnancement cyclique du système de manutention d'une ligne de galvanoplastie, phdthesis, Besançon, 1994.
- [30] J. H. Holland, "Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence.", The University of Michigan Press, 1975.
- [31] D. E. Goldberg, and J. H. Holland, "Genetic algorithms and machine learning.", Machine learning, vol. 3(2), pp. 95–99, Springer, 1988.
- [32] D. Kalyanmoy, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm : NSGA-II.", IEEE Transactions on Evolutionary Computation, vol. 6(2), pp. 182–197, Institute of Electrical and Electronics Engineers, 2002.
- [33] 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.", Journal of Manufacturing Systems, vol. 36, pp. 62–75, Elsevier, 2015.
- [34] Y-n. Mao, Q-h. Tang, Z-x. Li, and L-p. Zhang, "Mixed-integer linear programming method for multi-degree and multi-hoist cyclic scheduling with time windows.", Engineering Optimization, vol. 50(11), pp. 1978–1995, Taylor & Francis, 2018.