MILP and CP Approaches for Human-Robot Collaborative Flow-Shop Scheduling

Dahibou Fall SOW[1], Meriem TOUAT[2], and Haoxun CHEN[1]

***Abstract*— Traditional scheduling problems fail to capture some important features of modern workshops. They typically consider only a single type of resource, either humans or robots, which makes them not suitable for workshops in Industry 5.0. In this paper, we study a NP-hard human–robot collaborative flowshop scheduling problem. This problem is characterized by strong interactions between human operators and robots and the consideration of multiple working shifts within a week. It considers both the assignment of operators to each shift and the scheduling of jobs in each shift while respecting some managerial requirements. The objective of this problem is to minimize the total tardiness of all jobs. A mixed-integer linear programming (MILP) model and a constraint programming (CP) model are formulated for the problem. The two models are solved by the MILP solver and the CP optimizer of CPLEX respectively. Computational experiments conducted on 54 randomly generated instances demonstrate the outperformance of the CP approach over the MILP approach in terms of scalability and solution quality. Specifically, the CP approach successfully found the best-known solution for 21 out of the 54 instances, whereas the MILP approach failed to achieve optimality for any instance within the computation time of 3600 seconds on a personal computer.**

I. INTRODUCTION

Manufacturing companies are using cutting-edge technologies such as artificial intelligence and collaborative robotics to modernize their production lines in the era of Industry 5.0. The goal of this digital transformation is to improve efficiency, adaptability, and product customization.

However, resource scheduling and planning now face new challenges as a result of this transformation. The rising complexity of contemporary production systems is difficult to address by traditional scheduling problems, which often impose assumptions like the utilization of a single type of resource, although many systems use a variety of resources, such as robots and human operators.

Moreover, in real industrial environment, jobs are usually scheduled for multiple shifts with consideration of the workloads of operators, work shift planning is thus crucial. Ignoring this planning may make a job schedule impractical or difficult to implement in such environments.

Human-robot collaborative scheduling (HRCS) has received increasing attention in recent years. Existing studies have explicitly considered both human and robotic resources to improve production performance. However, most studies are limited to simplified industrial settings or focus on only one aspect of the problem, such as robot allocation, task assignment, or makespan. Particularly, HRCS in flow-shop (FS) environments remains underexplored, especially when strong HRC, weekly shift planning, and tardiness of jobs are considered simultaneously.

Overcoming this limitation is important for practical applications. For instance, in automotive assembly lines, collaborative robots may assist human operators in performing repetitive or physically demanding tasks. In such environment, effective shift planning and task coordination between humans and robots can reduce production delays and improve customer satisfaction by meeting tight delivery due dates.

Existing studies on HRCS have addressed related problems from different perspectives. Some works focus on simplified FS environments. For instance, Johnson's rule is used to minimize the makespan in a two-stage FS with one collaborative robot and two workers ([1]). Although this study introduces human-robot cooperation in a FS environments, it does not consider work shift planning or job tardiness minimization.

Other studies consider integrated planning and scheduling with human and robotic resources. A simulation optimization approach for FS production is proposed, in which human operators and mobile robots are modeled as independent resources ([2]). However, the interaction between human operators and robots is not explicitly coordinated through shift assignment. Similarly, Constraint Programming (CP) and genetic algorithms are combined to schedule simultaneous human-robot work cells ([3]), but without explicitly considering weekly work shifts and operator assignment constraints.

Recent studies have also investigated human-robot collaboration in assembly lines or cooperative FS systems. Several collaboration modes between humans and robots are considered in an assembly line ([4]), while a HRCS-FS with constrained human and robot resources in studied ([5]). Nevertheless, these works mainly focus on makespan or cycle time minimization and do not jointly address strong HRCS, weekly shift planning, and total job tardiness minimization.

Job tardiness in HRCS remains less studied in the literature, although considerable attention has been given to criteria such as makespan, energy consumption, and resource utilization. However, meeting due dates is crucial in make-to-order production environments, where products are manufactured on demand and delivery delays may lead to customer dissatisfaction, penalties, or loss of future orders.

[1]Dahibou Fall SOW and Haoxun CHEN are with Laboratoire Informatique et Société Numérique (LIST3N), Université de Technologie de Troyes (UTT), 12 rue Marie Curie, CS 42060, Troyes 10300, France. dahibou.fallsow@utt.fr, haoxun.chen@utt.fr

[2]Meriem TOUAT is with Arts et Metiers Institute of Technology, LISPEN, France. meriem.touat@ensam.eu

From a methodological perspective, existing methods primarily rely on mixed-integer linear programming (MILP) or metaheuristic algorithms ([3]). However, these methods may face scalability issues when solving large-scale scheduling problems. In recent years, CP has emerged as an effective paradigm for scheduling applications. The effectiveness of CP for open-shop scheduling problems has been demonstrated ([6]), while MILP and CP formulations are proposed for the no-wait FS problem ([7]). The strength of CP Optimizer for job-shop scheduling and resource-constrained project scheduling problems is further shown ([8]). More recently, MILP and CP formulations are compared for several shop scheduling variants, including parallel machines, FS, job-shops, and open-shops ([9]).

To address these requirements, we introduce a collaborative human-robot flow-shop scheduling problem (CHRF-SSP), which is an extension of the NP-hard classical flow-shop scheduling problem (FSSP) ([10]). Because this problem incorporates strong human-robot interaction, weekly job-shift assignment, and the minimization of the overall job tardiness, it reflects the reality of contemporary FS.

To the best of our knowledge, this problem has not been addressed in the literature.

The contributions of this paper include:

- Introducing a new FSSP that takes work shifts planning and HRCS into account, with the total tardiness of jobs as its objective function.
- Propose CP and MILP model for the problem.
- Evaluate the performances of the two models solved by using a commerical solver.

The remainder of the paper is organized as follows. The problem and its MILP models are formulated in Section II. The CP model of this problem is presented in Section III. Computational results and performance analysis are preseneted in Section IV. Section V concludes this paper with suggestions for future research.

II. Problem description and MILP modeling

We examine an FSSP in a make-to-order manufacturing system, where no inventory is kept on hand and each product is produced only upon customer request.

The objective of this problem is to minimize the overall tardiness of all jobs, which is the total of all delays that occur with respect to their respective due dates. In make-to-order environments, when delayed deliveries may result in unhappy customers, contractual penalties, or lost future orders, this performance criterion is very crucial.

A series of M workstations arranged in a predetermined order and a set of N jobs make up the system.

Each job J_j, $j \in \{1, 2, \ldots, N\}$, must be processed successively on all workstations M_i, $i \in \{1, 2, \ldots, M\}$, following the same predefined order from M_1 to M_M. The operation of workstation M_i on job J_j is denoted by O_{ij} and requires a deterministic processing time p_{ij}. Each job j is associated with a due date d_j.

Each workstation can process at most one job at a time, and each job can be processed by only one workstation at any time. Operations are non-preemptive; once started, an operation must be completed without interruption.

The production system relies on heterogeneous resources composed of human operators and robots. Let W_w, $w \in \{1, 2, \ldots, W\}$ denote the set of human operators, and R_r, $r \in \{1, 2, \ldots, R\}$ the set of robots. Workstations are classified according to their level of automation into three categories:

- S_{11}: fully automated workstations operated exclusively by robots;
- S_{12}: semi-automated workstations requiring collaboration between a robot and a human operator;
- S_2: manual workstations operated solely by human operators.

In this paper, human-robot collaboration refers to the coordinated assignment of human and robotic resources to some workstations in the scheduling process. More specifically, collaboration occurs at semi-automated workstations S_{12}, where each operation requires both a robot and a human operator.

The set of workstations requiring human intervention is defined as $S' = S_{12} \cup S_2$. The assignment of robots to these workstations is pre-defined and not optimized within the problem. For each robot $r \in R$, we define the assignment variable z_{ri}, where $z_{ri} = 1$ if robot r is assigned to workstation M_i. They are given parameters.

Production is organized over a planning horizon of one week, composed of five working days with two shifts per day (day and night), resulting in a total of ten shifts. Each shift Sh_t, $t \in \{1, \ldots, 10\}$, has a fixed duration of 8 hours (480 minutes). Each human operator can be assigned to at most one shift per day, in accordance with standard labor regulations.

Now, we present a MILP model for the studied CHRFSSP. The parameters and decision variables used in this model are presented in TABLE I

$$\min \quad \sum_{j \in \mathcal{J}} T_j \tag{1}$$

$$\text{s.t.} \quad T_j \geq C_{Mj} - d_j, \quad T_j \geq 0 \qquad \forall j \tag{2}$$

$$C_{i+1,j} \geq C_{i,j} + p_{ij} \qquad \forall i = 1, \ldots, M-1, \ \forall j \tag{3}$$

$$C_{ij} \geq C_{ij'} + p_{ij'} - B(1 - x_{jj'i}) \qquad \forall i \in \mathcal{M}, \ \forall j \neq j' \tag{4}$$

$$x_{jj'i} + x_{j'ji} = 1 \qquad \forall i \in \mathcal{M}, \ \forall j \neq j' \tag{5}$$

$$\sum_{t=1}^{10} y_{jt} = 1 \qquad \forall j \in \mathcal{J} \tag{6}$$

$$y_{jt} \leq \sum_{w \in \mathcal{W}} b_{wti} \qquad \forall i \in S', \ \forall j, \ \forall t \in \mathcal{SH} \tag{7}$$

$$\sum_{w \in \mathcal{W}} b_{wti} = 1 \qquad \forall i \in S', \ \forall t \in \mathcal{SH} \tag{8}$$

$$\sum_{i \in S'} b_{wti} \leq 1 \qquad \forall w \in \mathcal{W}, \ \forall t \in \mathcal{SH} \tag{9}$$

$$\sum_{i \in S'} (b_{w,2d-1,i} + b_{w,2d,i}) \leq 1 \qquad \forall w, \ \forall d \in \{1, \ldots, 5\} \tag{10}$$

$$C_{ij} - p_{ij} \geq \text{Start}_t - B(1 - y_{jt}) \qquad \forall i \in \mathcal{M}, \ \forall j, \ \forall t \in \mathcal{SH} \tag{11}$$

$$C_{ij} \leq \text{End}_t + B(1 - y_{jt}) \qquad \forall i \in \mathcal{M}, \ \forall j, \ \forall t \in \mathcal{SH} \tag{12}$$

$$x_{jj'i} \in \{0, 1\} \qquad \forall i \in \mathcal{M}, \ \forall j \neq j' \tag{13}$$

$$y_{jt}, \ b_{wti} \in \{0, 1\} \qquad \forall j, \ \forall i \in S', \ \forall w, \ \forall t \tag{14}$$

$$C_{ij} \geq 0 \qquad \forall i \in \mathcal{M}, \ \forall j \in \mathcal{J} \tag{15}$$

Equation (1) defines the objective function as the total tardiness of all jobs. Equation (2) describes the tardiness of each job. Constraint (3) guarantees that a workstation is allowed to process a job after the job has been processed on the previous workstation. Constraint (4) ensures that a job can only start on a workstation after the previous job has been completed. Constraint (5) requires a feasible ordering of all jobs on all workstations. Constraint (6) assigns each job to exactly one shift. Constraints (7) and (8) ensure that,

Symbol	Description
N	Total number of jobs
M	Total number of workstations
W	Total number of human operators
S_1	Set of robotic workstations, $S_1 = S_{11} \cup S_{12}$
S_2	Set of non-robotic workstations
S_{11}	Set of fully robotic workstations
S_{12}	Set of partially robotic workstations
S'	Set of workstations requiring human intervention, $S' = S_2 \cup S_{12}$
$\mathcal{SH}$	Set of shift
p_{ij}	Processing time of operation O_{ij}
d_j	Due date of job j
B	Large positive constant (Big-M)
Start_t	Start time of shift t
End_t	End time of shift t
j	Index of jobs, $j \in \mathcal{J} = \{1, \ldots, N\}$
i	Index of workstations, $i \in \mathcal{M} = \{1, \ldots, M\}$
w	Index of human operators, $w \in \mathcal{W}$
t	Index of shifts, $t \in \mathcal{SH}$
s	Index of workstations requiring human intervention, $s \in S'$
$x_{jj'i}$	1 if j' precedes j on workstation i, 0 otherwise
y_{jt}	1 if job j is processed during shift t, 0 otherwise
C_{ij}	Completion time of operation O_{ij}
T_j	Tardiness of job j
b_{wts}	1 if operator w is assigned to workstation s during shift t, 0 otherwise

TABLE I: Notation used in the model.

if a job is scheduled on a workstation requiring human intervention, at least one operator is assigned to it, and that each of such workstations has exactly one operator per shift. Constraint (9) limits each operator to operate (work for) at most one workstation per shift, and constraint (10) prevents an operator from working in both shifts on the same day. Constraints (11) and (12) ensure that all human-related operations of each job start after its assigned shift begins and finish before the shift ends. Constraints (13) indicate the binary nature of assignment and ordering variables, while constraints (14) indicate the binary nature of allocation variables. Finally, constraints (15) impose non-negativity on continuous variables such as completion times and operator workloads.

Robots are involved in the workstation sets S_{11} and S_{12} specified by the parameters z_{ri}. S_{11} stations are robot-only stations, whereas S_{12} stations require both a robot and a human operator.

III. CP MODELING

The problem under study can also be formulated as a CP model, and it is solved using the commercial IBM CP Optimizer, which offers specialized scheduling tools including interval variables and optional intervals ([8]). We first introduce the following decision variables:

- $\mathbf{O}_{ij}$: interval variable between the start and the end of operation O_{ij}.
- $\mathbf{X}_{jt}$: optional interval variable indicating that job j is executed in shift t.
- $\mathbf{Y}_{wti}$: optional interval variable indicating that worker w operates workstation $i \in S'$ during shift t.

Using these decision variables, the objective function and constraints of the CP model can be formulated as:

$$\min \quad f = \sum_{j \in \mathcal{J}} \max\{0, \texttt{endOf}(\mathbf{O}_{Mj}) - d_j\} \tag{16}$$

$$\text{s.t.} \quad \texttt{endBeforeStart}(\mathbf{O}_{ij}, \mathbf{O}_{i+1,j}), \qquad \forall j,\ \forall i \tag{17}$$

$$\texttt{noOverlap}(\{\mathbf{O}_{ij} : j \in \mathcal{J}\}), \qquad \forall i \in \mathcal{M} \tag{18}$$

$$\sum_{t \in \mathcal{SH}} \texttt{presenceOf}(\mathbf{X}_{jt}) = 1, \qquad \forall j \in \mathcal{J} \tag{19}$$

$$\texttt{presenceOf}(\mathbf{X}_{jt}) = 1 \Rightarrow \begin{cases} \texttt{startOf}(\mathbf{O}_{ij}) \geq \text{Start}_t, \\ \texttt{endOf}(\mathbf{O}_{ij}) \leq \text{End}_t, \end{cases} \qquad \forall i,\ \forall j,\ \forall t \tag{20}$$

$$\sum_{w \in \mathcal{W}} \texttt{presenceOf}(\mathbf{Y}_{wti}) = 1, \qquad \forall i \in S',\ \forall t \in \mathcal{SH} \tag{21}$$

$$\sum_{i \in S'} \texttt{presenceOf}(\mathbf{Y}_{wti}) \leq 1, \qquad \forall w,\ \forall t \in \mathcal{SH} \tag{22}$$

$$\sum_{\substack{t \in \mathcal{SH}: \\ \text{day}(t) = d}} \sum_{i \in S'} \texttt{presenceOf}(\mathbf{Y}_{wti}) \leq 1, \qquad \forall w,\ \forall d \in \{1, \ldots, 5\} \tag{23}$$

Equation (16) defines the objective function of the CP model as the total tardiness of all jobs. The classical structure of FS is enforced by constraints (17): each workstation can only process one job at a time, all processing times are given, and each job must be processed by all workstations in a specified order. Each job must be assigned to exactly one shift, according to constraint (19). Constraint (20) ensures that jobs are completed in their assigned shift by requiring all operations of a job to start after a shift begins and terminate before the shift ends. Constraints (21) and (22) ensure that no worker can be allocated to more than one workstation during the same shift and that each workstation requiring human intervention is managed by exactly one worker in each shift. Constraint (23) guarantees adherence to labor and rest regulations by prohibiting operators from being assigned to two distinct shifts on the same day.

IV. COMPUTATIONAL STUDY

The purpose of this section is to evaluate the performance of the proposed CP and MILP models for the considered scheduling problem when solved using commercial CP and MILP solvers, respectively. In Section IV-A, we present the data generation of the instances tested. Section IV-B shows the computational results of the performance evaluation of the two models. Both models were implemented in **Python** using the **DOcplex** library and executed on a PC running **Windows 10**, equipped with an **Intel® Core™i7-9850H** processor and **16 GB of RAM**. They were solved by the CP optimizer and the MILP solver of IBM ILOG CPLEX respectively.

A. Instance Generation

Since no standard benchmark exists for the proposed HRCFSS problem, the instances tested were generated randomly to reflect different production sizes, workstation automation levels, and due date tightness. Based on experimental frameworks commonly adopted in the literature, and more specifically in [11], the data of the instances were generated in a manner consistent with our shift planning problem.

Each instance is associated with a workstation-job configuration (M, N), where $M \in \{3, 5, 7\}$ denotes the number of workstations and $N \in \{20, 50\}$ denotes the number of jobs.

- Processing times: Randomly generated according to a discrete uniform distribution with a 5 minute time granularity. The processing time ranges depend on the instance size, as follows:
  - $[10, 60]$ minutes for instances with $N = 20$ jobs;
  - $[10, 40]$ minutes for instances with $N = 50$ jobs;
- Due dates: Generated following the method in [12] sampled from the interval $\left[P\left(1-\tau-\frac{\rho}{2}\right), P\left(1-\tau+\frac{\rho}{2}\right)\right]$, where $P = \max_{i\in\{1,\ldots,m\}} \sum_{j=1}^{n} p_{ij}$, $\tau \in \{0.2, 0.4, 0.6\}$ is the tardiness factor and $\rho \in \{0.2, 0.6, 1\}$ is the due date range.
- Workstation types and operators: Three scenarios are considered regarding workstation configurations:
  (i) 20% fully robotic, 40% semi-robotic, and 40% manual;
  (ii) 60% fully robotic, 20% semi-robotic, and 20% manual;
  (iii) 40% fully robotic, 20% semi-robotic, and 40% manual.

  Since semi-robotic and manual stations require human intervention, the number of available operators is defined as $W = 2 \times$ the number of workstations requiring human intervention, that is, twice the number of workstations involving human labor. The assignment of robots to workstations (z_{ri}) is generated randomly.
- Shift: Each shift lasts 480 minutes, with start and end times regularly spaced across the planning horizon. So We have $\mathcal{SH} = \{480, 968, ..., 4800\}$.
- Big-M Constant B: Defined as the product of the shift duration and 10, i.e., B = shift duration× 10.
- Three distinct instances were created for each paramer setting by randomly generating both processing times and due dates.

In total, we generated $3 \times 2 \times 3 \times 3 = 54$ instances. Each instance was solved by ILOG IBM CPLEX solver with a CPU time limit of 3600 seconds.

We use two performance measures, namely the optimality gap and the relative percentage deviation (RPD), to evaluate the performance of our proposed CP based method with respect to MILP based method.

The RPD measures the quality of the best feasible solution obtained by a given method with respect to the best upper bound found by both methods. It is defined as:

$$\text{RPD}_{\text{method}} = \frac{UB_{\text{method}} - UB^*}{UB^*},$$

where UB_{method} is the best upper bound obtained by the considered method, and UB^* is the best upper bound found by both methods.

The optimality gap evaluates the relative deviation between the best feasible solution and the best lower bound obtained by a given method. It is computed as:

$$\text{gap}_{\text{method}} = \frac{UB_{\text{method}} - LB_{\text{method}}}{LB_{\text{method}}},$$

where LB_{method} denotes the best lower bound obtained by a given method.

B. Performance evaluation of the CP

Tables II report the results, for all combinations of tested instances and for both MILP and CP. The reported parameters include: $\mathcal{M}$, the number of machines; $\mathcal{J}$, the number of jobs; Sc, the scenario configuration; Inst, the instance number; f_{Cplex}, the objective function value obtained by the MILP solver of CPLEX; f_{CP}, the objective function value obtained by the CP optimizer of CPLEX; CPU_MILP and CPU_CP, the computation times for the MILP and CP methods respectively; gap_MILP and gap_CP, the relative gaps of the solution compared to the optimal lower bound for MILP and CP respectively; and RPD_MILP, the relative deviation of the MILP solution compared to the best solution found by the two methods.

For the **54** tested instances, the CP approach is able to find the best solution for **21** instances within the given time limit, whereas the MILP model fails to prove optimality for any instance within **3600** seconds.

The CP method consistently achieves zero gap, whereas the MILP method has a non-negligible average optimality gap of around **42%** for the instances with the smallest configuration. For instance, for the configuration $(\mathbf{M = 3, N = 20})$, The average optimality gap is around **42%**, which increases significantly as the problem size grows, reaching **200%** for $(\mathbf{M = 3, N = 50})$ and over **1000%** for $(\mathbf{M = 7, N = 50})$.

By contrast, the CP approach maintains a low average gap, with a much better performance even for the instances with larger configurations. For the $(\mathbf{M = 3, N = 50})$ configuration, the gap remains around **8%**, while the optimality gap of the MILP method is **200%**.

Furthermore, the CP approach consistently achieves a zero RPD for all tested instances, indicating that CP always reaches the best solution for the entire set of instances.

On the other hand, even for small and medium-sized instances, the MILP method shows noticeably bigger relative deviations.

Based on these results, we conclude that the proposed CP approach outperforms the MILP method. For all the studied configurations, it achieves the largest number of optimal solutions, the lowest average optimality gap, as well as the

$\mathcal{M}$	$\mathcal{J}$	Sc	Inst	f_{Cplex}	f_{CP}	CPU_MILP	CPU_CP	gap_MILP	gap_CP	RPD_MILP
3	20	1	1	282.05	93.87	>3600	13	44.67	0	200.46
3	20	2	1	182.09	90.36	>3600	16	33.50	0	101.41
3	20	3	1	309.13	105.64	>3600	8	55.42	0	192.62
3	20	1	2	128.52	26.13	>3600	5	73.47	0	391.84
3	20	2	2	67.92	25.92	>3600	17	20.52	0	162.03
3	20	3	2	109.25	28.57	>3600	7	83.16	0	282.39
3	20	1	3	106.24	80.64	>3600	51	27.80	0	31.74
3	20	2	3	90.35	79.02	>3600	27	3.48	0	14.32
3	20	3	3	148.61	83.24	>3600	8	44.35	0	78.53
3	50	1	1	10716.05	1123.18	>3600	>3600	232.32	12.04	854.08
3	50	2	1	4844.97	1303.97	>3600	>3600	255.56	7.03	271.55
3	**50**	3	1	8230.27	1462.89	>3600	>3600	**260.81**	**8.14**	462.60
3	50	1	2	5712.82	1876.06	>3600	>3600	198.33	20.72	204.51
3	50	2	2	9100.01	1035.22	>3600	>3600	192.48	15.08	779.04
3	50	3	2	17318.17	765.63	>3600	>3600	333.46	8.24	2161.95
3	50	1	3	6467.62	1253.55	>3600	2546	177.30	0	415.94
3	50	2	3	6649.53	702.14	>3600	2801	140.16	0	846.87
3	50	3	3	6390.97	996.61	>3600	2366	194.88	0	541.26
5	20	1	1	1673.06	1058.133	>3600	3244	87.80	0	58.10
5	20	2	1	1614.72	1369.209	>3600	2916	92.48	0	17.93
5	20	3	1	1726.8	1095.69	>3600	3472	92.35	0	57.61
5	20	1	2	1960.7	1128.94	>3600	3432	99.42	0	73.62
5	20	2	2	1709.88	1256.02	>3600	2557	105.06	0	36.16
5	20	3	2	2072.04	1134.84	>3600	2986	99.60	0	82.55
5	20	1	3	1739.77	932.54	>3600	1657	87.80	0	86.53
5	20	2	3	1709.88	973.72	>3600	2557	92.48	0	75.57
5	20	3	3	2063.71	969.08	>3600	1694	100.03	0	112.94
5	50	1	1	15138.15	7961.805	>3600	>3600	138.46	96.28	90.11
5	50	2	1	11232.13	7424.54	>3600	>3600	122.34	96.01	51.28
5	50	3	1	24386.25	6389.286	>3600	>3600	101.03	95.36	281.59
5	50	1	2	17198.06	7046.99	>3600	>3600	275.35	96.89	144.04
5	50	2	2	21035.82	6245.26	>3600	>3600	352.88	96.66	236.82
5	50	3	2	19241.57	6390.21	>3600	>3600	257.12	96.57	281.59
5	50	1	3	15328.02	6008.17	>3600	>3600	428.16	95.81	155.12
5	50	2	3	11436.54	5971.71	>3600	>3600	317.32	95.78	91.51
5	50	3	3	11035.21	3915.31	>3600	>3600	297.25	93.57	181.84
7	20	1	1	3673.11	1495.56	>3600	>3600	166.32	68.56	145.61
7	20	2	1	2197.29	1866.98	>3600	>3600	103.54	66.01	17.70
7	20	3	1	2397.29	1869.8	>3600	>3600	101.01	69.03	28.21
7	20	1	2	2326.07	1448.83	>3600	>3600	107.40	76.28	60.55
7	20	2	2	3300.24	1412.78	>3600	>3600	99.35	77.28	133.60
7	20	3	2	3739.54	1982.54	>3600	>3600	98.77	76.69	88.61
7	20	1	3	2326.07	1123.64	>3600	>3600	100.72	57.32	106.99
7	20	2	3	3426.46	1109.02	>3600	>3600	92.43	55.62	208.95
7	20	3	3	3426.46	1221.48	>3600	>3600	121.72	55.36	180.52
7	50	1	1	46151.26	12112.20	>3600	>3600	657.23	98.22	281.05
7	50	2	1	48519.55	11870.34	>3600	>3600	707.39	82.48	308.68
7	50	3	1	52947.84	11901.23	>3600	>3600	898.73	87.36	344.87
7	50	1	2	46269.19	11337.79	>3600	>3600	801.35	99.82	308.13
7	50	2	2	58346.61	10054.45	>3600	>3600	623.17	99.03	480.31
7	50	3	2	37597.71	11056.91	>3600	>3600	588.72	99.75	240.04
7	**50**	1	3	30663.40	12346.61	>3600	>3600	**1367.31**	87.97	148.35
7	**50**	2	3	28183.43	10646.2	>3600	>3600	**1491.16**	85.06	164.73
7	**50**	3	3	23778.67	11481.32	>3600	>3600	**1513.25**	85.29	107.11

TABLE II: Computational results.

smallest relative deviation from the best solution found by the two methods.

By contrast, the MILP model suffers from a significant degradation in solution quality as the problem size increases.

Regarding computation time, CP is highly efficient for the instances with smaller configurations, with average computation time as low as **16.9** seconds for $(\mathbf{M = 3, N = 20})$. However, for larger configurations, computation time increases rapidly, reaching the **3600** second time limit for $(\mathbf{M = 5, N = 50})$ and beyond. By contrast, the MILP method suffers significant degradation in solution quality as the problem size increases, with large optimality gaps for the bigger configurations.

Overall, these results show that while the CP approach is highly efficient for small instances, its computation time increases rapidly as the problem complexity grows. In particular, configurations with a large number of machines and jobs constitute the most challenging instances, confirming

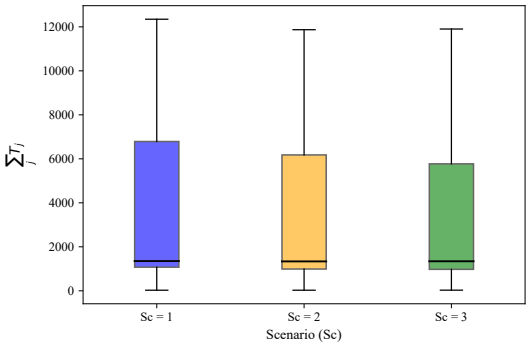

Fig. 1: Total tardiness by scenario for 54 instances.

that both parameters are the main factors driving the growth of the computation time.

C. Sensitivity analysis

This sensitivity analysis evaluates the impact of the three resource allocation scenarios on the performance of the proposed HRCS problem, with the total tardiness $\sum_j T_j$ as the performance indicator. The distribution of $\sum_j T_j$ under each scenario is illustrated by the boxplot in Figure 1.

It shows that the three scenarios lead to noticeably different distributions of total tardiness, highlighting the influence of resource allocation decisions on scheduling performance. Scenario 1 exhibits the lowest median total tardiness and a relatively narrow interquartile range, indicating more balanced schedules and a stable behavior across instances. This suggests that the corresponding allocation of human and robotic resources provides an effective coordination between tasks, limiting delays.

Scenario 2 presents a higher median tardiness and a wider dispersion, reflecting a greater sensitivity to instance characteristics.

Scenario 3 shows the largest variability in total tardiness, with a higher upper quartile and more pronounced outliers. Although some instances achieve competitive tardiness values, the overall dispersion suggests that this scenario is less efficient and may generate highly unbalanced schedules depending on the workstation-job configuration.

Overall, this analysis demonstrates that while all three scenarios are feasible, their impact on performance differs substantially. Scenario 1 appears to be the most efficient and reliable in terms of minimizing total tardiness, whereas Scenarios 2 and 3 exhibit higher variability and are more sensitive to the underlying problem parameters. These results underline the importance of carefully selecting the resource allocation strategy when designing efficient HRC schedules.

V. Conclusion

This paper has addressed a HRCS problem in a FS environment characterized by strong interaction between human operators and robots, under realistic industrial constraints. The studied problem extends the classical flowshop scheduling problem by jointly considering weekly shift planning, operator assignment, and minimization of the total tardiness of jobs, which are key decisions for make-to-order production systems. To the best of our knowledge, such joint decisions has not been addressed in the literature. Two exact approaches were proposed and evaluated: a MILP-based approach and a CP-based approach. Computational experiments were conducted on a set of 54 instances. Within a time limit of 3600 seconds, the MILP approach was unable to prove optimality for any tested instance, highlighting the intrinsic complexity of the problem, especially as its size increases. By contrast, the CP approach found the best solutions for 21 out of the 54 tested instances, demonstrating a significantly better robustness and scalability. Even for large-scale instances involving a high number of jobs and workstations, the CP approach was able to deliver high-quality solutions within the imposed time limit, whereas the MILP method struggled to converge. For future research, several promising directions can be explored. In particular, the development of metaheuristic or hybrid approaches, such as genetic algorithms, tabu search, or variable neighborhood search, could provide high-quality solutions within reasonable computation times for very large instances where exact methods struggle.

References

[1] A. R. Sadik, A. Taramov, and B. Urban, "Optimization of tasks scheduling in cooperative robotics manufacturing via Johnson's algorithm: Case study of one collaborative robot in cooperation with two workers," in *Proc. 2017 IEEE Conf. on Systems, Process and Control (ICSPC)*, pp. 36–41, 2017.

[2] M. Vieira, S. Moniz, B. S. Gonçalves, T. Pinto-Varela, A. P. Barbosa-Póvoa, and P. Neto, "A two-level optimisation-simulation method for production planning and scheduling: The industrial case of a human–robot collaborative assembly line," *International Journal of Production Research*, vol. 60, no. 9, pp. 2942–2962, 2022.

[3] C. Ferreira, G. Figueira, and P. Amorim, "Scheduling human-robot teams in collaborative working cells," *International Journal of Production Economics*, vol. 235, p. 108094, 2021.

[4] A. Keshvarparast *et al.*, "Integrating collaboration scenarios and workforce individualization in collaborative assembly line balancing," *International Journal of Production Economics*, vol. 279, p. 109450, 2024.

[5] C. Zheng *et al.*, "Balancing and scheduling human-robot collaborative assembly lines with heterogeneous robots and limited resources: Constraint programming approach and fruit fly optimization algorithm," *Computers & Industrial Engineering*, vol. 203, p. 111046, 2025.

[6] A. Malapert *et al.*, "An optimal constraint programming approach to the open-shop problem," *INFORMS Journal on Computing*, vol. 24, no. 2, pp. 228–244, 2012.

[7] H. Samarghandi and M. Behroozi, "On the exact solution of the no-wait flow shop problem with due date constraints," *Computers & Operations Research*, vol. 81, pp. 141–159, 2017.

[8] P. Laborie, "An update on the comparison of MIP, CP and hybrid approaches for mixed resource allocation and scheduling," in *Proc. CPAIOR*, Springer, 2018, pp. 403–411.

[9] B. Naderi, R. Ruiz, and V. Roshanaei, "Mixed-integer programming vs. constraint programming for shop scheduling problems: New results and outlook," *INFORMS Journal on Computing*, vol. 35, no. 4, pp. 817–843, 2023.

[10] M. Danishvar *et al.*, "Energy-aware flowshop scheduling: A case for AI-driven sustainable manufacturing," *IEEE Access*, vol. 9, pp. 141678–141692, 2021.

[11] L.-M. Liao *et al.*, "Tabu search for non-permutation flowshop scheduling problem with minimizing total tardiness," *Applied Mathematics and Computation*, vol. 217, no. 2, pp. 741–750, 2010, doi: 10.1016/j.amc.2010.07.025.

[12] F. Xiong, M. Chu, Z. Li, Y. Du, and L. Wang, "Just-in-time scheduling for a distributed concrete precast flow shop system," *Computers & Operations Research*, vol. 129, p. 105204, 2021.