




Full length article



Learning and planning for optimal synergistic human-robot coordination in manufacturing contexts

Samuele Sandrini ^{a,c} ,* , Marco Faroni ^b , Nicola Pedrocchi ^a

^a Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, Milan, Italy

^b Department of Electronics, Informatics and Bio-Engineering (DEIB), Politecnico di Milano, Milan, Italy

^c Department of Control and Computer Engineering (DAUIN), Politecnico di Torino, Turin, Italy

ARTICLE INFO

Dataset link: https://jrl-cari-cnr-unibs.github.io/synergistic_http/

Keywords:

Planning

Scheduling and coordination

Task planning

Collaborative robotics

Safety in HRC

ABSTRACT

Collaborative robotics cells leverage heterogeneous agents to provide agile production solutions. Effective coordination is essential to prevent inefficiencies and risks for human operators working alongside robots. This paper proposes a human-aware task allocation and scheduling model based on Mixed Integer Nonlinear Programming to optimize efficiency and safety starting from the task planning stages. The approach exploits synergies that encode the coupling effects between pairs of tasks executed in parallel by the agents, arising from the safety constraints imposed on robot agents. These terms are learned from previous executions using a Bayesian estimation; the inference of the posterior probability distribution of the synergy coefficients is performed using the Markov Chain Monte Carlo method. The synergy enhances task planning by adapting the nominal duration of the plan according to the effect of the operator's presence. Simulations and experimental results demonstrate that the proposed method produces improved human-aware task plans, reducing useless interference between agents, increasing human-robot distance, and achieving up to an 18% reduction in process execution time.

1. Introduction

Production stations often involve the coexistence of human operators and robots to ensure flexibility, reconfigurability, and suitability for high-mix, low-volume production demands. Safety has a direct impact in terms of efficiency in collaborative environments [1], e.g., the *Speed and Separation Monitoring* (SSM) proposed by ISO/TS 15066 [2] requires that the robot speed be modulated based on the agents' distance to keep a protective separation. Researchers have tackled the "safety-vs-efficiency" trade-off by mainly acting on motion planning and control levels (e.g., impedance control [3,4], safety-aware path planning [5,6], and fast motion replanning [7]). At the same time, only a few works focused on the task planning layer (e.g. [8]). When the task planner is unaware of safety aspects, it produces plans that may lead to interference between the agents and increase the frequency of safety stops that must be activated to guarantee the operators' safety. In such a context, our paper investigates the coordination of agents to achieve efficient production, considering safety-related effects and agent interference. The approach pursues a synergistic human-robot collaboration for Industry 5.0 [9], with particular relevance to assembly and disassembly applications, object sorting, and bin picking for collaborative robotic cells, as shown in Fig. 1.

1.1. Related works

Task Allocation and Scheduling in Human-Robot Collaboration (HRC) have been tackled using heterogeneous approaches. Several works focused only on the task allocation problem [10]. For example, [11] uses AND/OR graph plan representation [12] and solves the optimal allocation problem by graph search with A*. Building upon this work, [13] emphasized the agent's capability-based allocation problem by introducing indices about task complexity, agent dexterity, and effort. In [14,15], the problem of assigning tasks to agents online is addressed using the AND/OR graph representation and the AO* graph search, taking into account the physical load of the agent that is quantified by motion capture and ergonomic risk assessment. The ergonomics factor in the allocation problem plays a crucial role in [16,17]. In [18], the online task allocation problem is integrated with a custom *control node* into a *Behavior Tree* [19]; the allocation is solved through Mixed Integer Linear Programming (MILP) on a subset of tasks at runtime only when it is time to perform that subset of actions. [20] investigates both allocation and scheduling. The allocation is solved offline based on agent capabilities and is decoupled from the scheduling. A variant

* Corresponding author at: Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, Milan, Italy.

E-mail address: samuele.sandrini@stiima.cnr.it (S. Sandrini).

<https://doi.org/10.1016/j.rcim.2025.103006>

Received 27 July 2024; Received in revised form 21 January 2025; Accepted 20 February 2025

Available online 11 March 2025

0736-5845/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

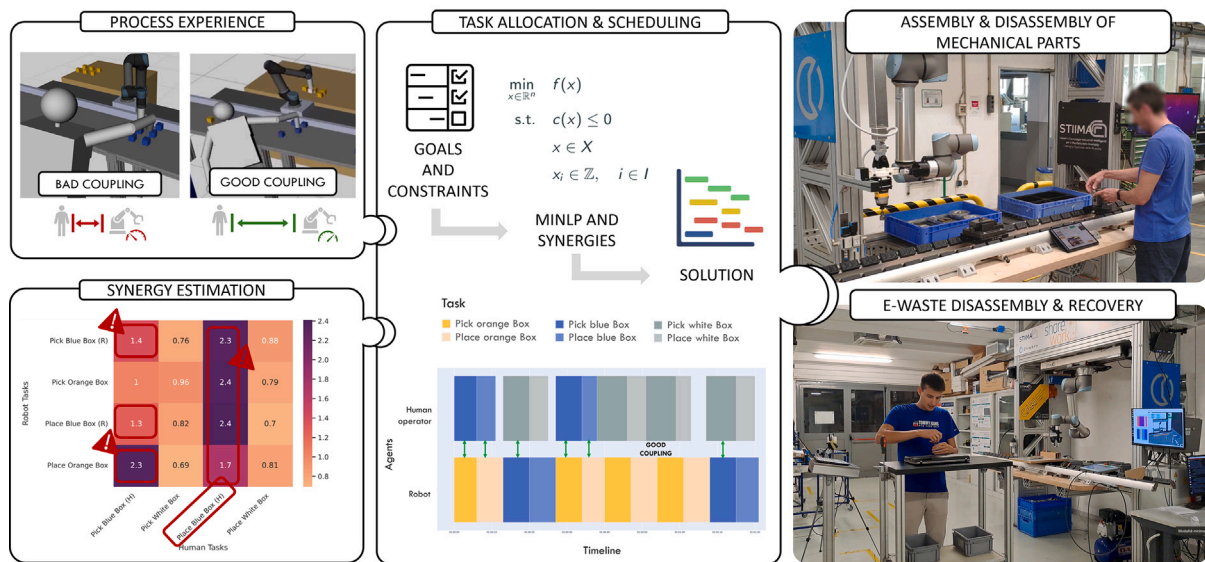


Fig. 1. Visual description of the proposed method. The boxes on the left show how task statistics and task synergies of the various agents are estimated using process experience, highlighting good or bad couplings between them. In the center, it is shown how these synergies are integrated to improve the MINLP algorithm for task allocation and scheduling. The boxes on the right show two industrial application scenarios that can benefit from the proposed method. Supplementary material can be found: https://jrl-cari-cnr-unibs.github.io/synergistic_http/.

of the Hungarian algorithm [21] solves the allocation, while a MILP approach solves the scheduling.

Some works incorporate the degradation of human performance during execution [22–24]. The work in [25] introduces the concept of awareness of the actual duration of the tasks performed by the operator using a two-level architecture: (i) a task allocation and scheduling layer based on Multi-Objective MILP and (ii) a dynamic scheduling layer that reschedules when triggered by the agents monitoring. The same architecture was used in [26] to adapt the schedule locally in case of failure.

In [8,27] the human–robot coordination is pursued through a *Timeline-based* task planner included in the *Task and Motion Planning (TAMP)* framework [28]. In [29], the authors combine TAMP with a human motion prediction module based on a goal-conditioned recurrent neural network to reduce human–robot interference. *Human-Aware Task Planning* was studied by Alami et al. in [30] using the *Hierarchical Task Network* paradigm [31]. This evolved into the *HATP (Hierarchical Agent-based Task Planner)* framework [32–34] and its variants [35], which consider the human agent as a rational agent with its decision model. *Theory of Mind (ToM)* was recently explicitly introduced at the task planning level in [36].

A branch of the literature focused on allocation and scheduling problems for industrial applications. For instance, [37] handles the coordination of heterogeneous agents using a MILP formalization and constrained scheduling for aircraft assembly; [38,39] apply timeline-based planning in the TAMP framework to a semi-automated assembly/disassembly industrial case study; [40] proposed a task allocation technique for HRC using a neural network for electric vehicle battery disassembly; [41] takes up the MILP formulation to handle allocation and scheduling for the assembly of printed circuit boards using meta-heuristic techniques; and [42] solves the allocation problem using genetic algorithms for HRC in mold assembly processes.

Some works consider safety and efficiency jointly at task planning level. [43] formalizes human-multi-robot allocation and scheduling problems jointly based on MILP; the cost function is optimized by considering the makespan, an index of task execution quality, and the agents' workload. They consider safety issues by introducing *spatial constraints*, avoiding the parallel execution of tasks in the same work area, and providing online monitoring and rescheduling. The agents'

occupancy maps are known beforehand. In [44], they introduce the cost of task switches during online rescheduling. [45] reformulates the dynamic scheduler in [23,25] to online adapt the schedule locally in case of safety halts. However, this implements an online countermeasure to risky or blocking situations without the proactivity needed to avoid such situations. [46] considers safety by parameterizing the optimization problem with the nominal SSM parameters. It assumes that each task pair is associated with a pair of 2D points and corresponds to a safety speed reduction, and the human movement is static at that point and deterministic given a task.

1.2. Contribution

Although most of the literature considers safety as an online countermeasure, this paper takes a synergistic approach at the task planning level by introducing synergy coefficients into the planning model. This provides a safe and efficient solution to agents' coordination in collaborative robotics. The key contributions of this paper can be summarized as follows:

Learning of human–robot synergies. We estimate the coupling effects on execution time between pairs of human–robot tasks, leveraging knowledge gained from previous process executions. Building on the definition of synergy that we presented in [47], we adopt a Bayesian estimation approach to learn coefficients. This approach iteratively updates the estimated synergy distribution according to new process observations.

Human-aware task planning. Mixed Integer Nonlinear Programming (MINLP) is used to define human-aware task allocation and scheduling models. It integrates the learned synergies into the planning model to consider the coupling effect (detrimental or beneficial) caused by the parallel execution of tasks between robots and operators. We proposed two models: the first is called Synergistic Task Planning (STP), in which the completion time of robots' tasks is interdependent through synergy values with tasks executed in parallel by the human agent, and the second, Relaxed Synergistic Task Planning (R-STP), which reduces the computational complexity of STP by considering a simplified synergy term in the cost function.

Table 1

List of main variables involved in model formulation for the human-aware task allocation and scheduling.

Symbol	Description
$\mathcal{T}, m = \mathcal{T} , \tau_i \in \mathcal{T}$	The task set, the total number of tasks, the i th task
$\mathcal{A}^R, H, \mathcal{A} = \mathcal{A}^R \cup H$	The robot agent set, the human agent, the set of agents
$t_i^s \in \mathbb{R}^+$ and $t_i^e \in \mathbb{R}^+$	Start-Time and End-Time of task τ_i
$a_i^j \in \{0, 1\}$	Binary variable for the assignment of task τ_i to agent j
$C_{i,j} \in \{0, 1\}$	Binary input variable that specifies whether agent j can perform task τ_i
$P_{i,k} \in \{0, 1\}$	Binary input variable that specifies whether task τ_i must precede task τ_k
$d_{i,j}^j \in \mathbb{R}^+$	Expected duration of task τ_i executed by agent j
$OV_{i,k} \in \mathbb{R}_0^+$	Overlapping time between tasks τ_i and τ_k
$s_{i,k}^j \in \mathbb{R}^+$	Synergy term between task τ_i (executed by agent j) and task τ_k (executed by the human agent)
D	Dataset containing task observations

Unlike state-of-the-art approaches based on Mixed Integer Linear Programming, in the proposed models, the execution times of tasks performed by robotic agents are not fixed to their nominal values independently of human operator actions executed in parallel. Instead, they are influenced by these actions and adjusted according to synergy values. Moreover, the proposed approach is not parameterized over low-level implementations of safety specifications, geometric task layouts, or manually defined human–robot interactions. This makes our approach flexible for reconfiguring the process and robot programming. Such reconfiguration requires learning the synergy values and tasks' duration from new simulated or actual process data. Compared to [43,46], our models do not rely on the nominal parameters of the SSM, nor *a-priori* definition of static points associated with tasks, nor on manually tuned constraints of non-parallelism between tasks heuristically considered close.

The proposed methods were evaluated and compared with multiple baselines in a simulated HRC scenario and a case study of electronic waste disassembly in the real world. Simulations and real-world experiments demonstrate that our approach reduces the cycle time (up to 18%) and leads to safer executions with a larger average human–robot distance. A preliminary human factor analysis also revealed a preference for the proposed methods in terms of robot proximity, interruptions, and overall satisfaction. A video of the experiments is available online: <https://youtu.be/GGVBM0dyQYo>. We show how to linearize the proposed MINLP models in Appendix.

2. Problem formulation

The allocation and scheduling of all the tasks $\tau_i \in \mathcal{T}$ to the available agents $j \in \mathcal{A}$ in order to obtain the optimal plan π^* that minimizes the makespan of the process can be formalized as an MINLP problem as follows:

$$\begin{aligned} \pi^* = \arg \min \quad & \mathcal{M} \\ & t_i^s, t_i^e, a_i^j \\ \text{subject to} \quad & C_{\text{goal}}, \\ & C_{\text{precedence}}, \\ & C_{\text{capability}}, \\ & C_{\text{overlapping}}, \\ & C_{\text{performance}} \end{aligned} \quad (1)$$

where \mathcal{M} is the makespan:

$$\mathcal{M} = \max_{i=1, \dots, m} (t_i^e), \quad (2)$$

with t_i^s and t_i^e the start and end times of the tasks, respectively, and a_i^j represents the allocation variable of tasks to agents. The optimization constraints are discussed in the following paragraphs.

Please note that all variables involved are listed in Table 1.

2.1. Process constraints

C_{goal} imposes that all the tasks involved in the process shall be executed once:

$$C_{\text{goal}} := \sum_{j \in \mathcal{A}} a_i^j = 1 \quad \forall i \in \{1, \dots, m\}. \quad (3)$$

2.2. Precedence constraints

The constraint $C_{\text{precedence}}$ enforces precedence between tasks by imposing that the start time of τ_i is greater than the end time of τ_k . Specifically, denote $P_{i,k}$ as a boolean variable that models the precedence between task τ_i and τ_k :

$$P_{i,k} = \begin{cases} 1 & \text{if } \tau_i \text{ must be executed before } \tau_k, \\ 0 & \text{otherwise (there is no order between } \tau_i \text{ and } \tau_k). \end{cases} \quad (4)$$

Then, the $C_{\text{precedence}}$ can be expressed as:

$$C_{\text{precedence}} := \begin{cases} \dots \\ t_i^s \geq t_k^e P_{i,k} & \forall i, k \quad \vee \quad k \neq i. \\ \dots \end{cases} \quad (5)$$

2.3. Capability constraints

$C_{\text{capability}}$ prevents a task is assigned to an agent that cannot perform it. Specifically, denote $C_{i,j}$ as a boolean variable that gathers the possible assignment of task τ_i to the j th agent:

$$C_{i,j} = \begin{cases} 1 & \text{if agent } j \text{ can execute } \tau_i, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Thus, the capability constraint can be defined as:

$$C_{\text{capability}} := \begin{cases} \dots \\ a_i^j \leq C_{i,j} & \forall i, j. \\ \dots \end{cases} \quad (7)$$

2.4. Non-overlapping constraints

$C_{\text{overlapping}}$ avoids overlapping of tasks assigned to the same agent. Specifically, we impose:

$$C_{\text{overlapping}} := \begin{cases} \dots \\ t_i^s > t_k^s - M(1 - \delta_{i,k}) - M(2 - a_i^j - a_k^j) \\ t_i^s \leq t_k^s + M\delta_{i,k} + M(2 - a_i^j - a_k^j) \\ t_i^e > t_k^e - M(1 - \delta_{i,k}) - M(2 - a_i^j - a_k^j) \\ t_i^e \leq t_k^e + M\delta_{i,k} + M(2 - a_i^j - a_k^j) \\ \dots \end{cases} \quad (8)$$

$$\forall i, k \quad \vee \quad k \neq i \quad \vee \quad \forall j.$$

This formulation uses the so-called ‘‘big-M method’’ [48], where a large constant M and a binary variable $\delta_{i,k}$ are introduced, to express the non-overlapping constraints in a general way with respect to the task

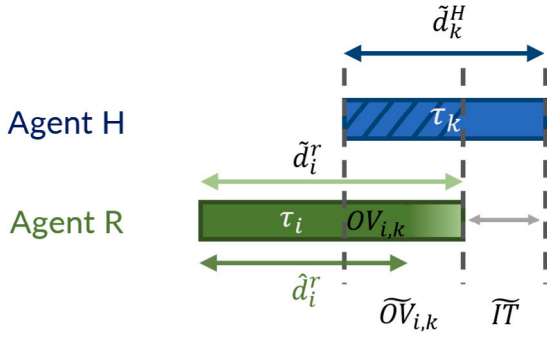


Fig. 2. Visualization of the involved variables during the overlapping between human and robot tasks. The measured durations (\tilde{d}_k^H for the human task and \tilde{d}_i^R for a task performed by a generic robot) together with the nominal duration \hat{d}_i^r of the task τ_i and the nominal $OV_{i,k}$ and actual $\widetilde{OV}_{i,k}$ overlapping are reported.

ordering and allocation. If the same agent does not execute both tasks, then $2 - a_i^j - a_k^j \neq 0$ and (8) always hold. In contrast, if $a_i^j = a_k^j = 1$, the constraints are managed by $\delta_{i,k}$. If $\delta_{i,k} = 1$, the first and third constraints are active (τ_i after τ_k), while the second and fourth always hold and vice versa if $\delta_{i,k} = 0$ (τ_i before τ_k).

2.5. Performance constraints

$C_{\text{performance}}$ determines the expected task end times based on the agent's performance and the impact of tasks executed in parallel. The formalization of such a constraint is crucial for the paper's contribution and needs some preliminary definitions, which are reported in the next paragraph. Then, two paragraphs will report two different models for this constraint.

2.5.1. Synergy among tasks

The approach presented here for human-robot synergies is based on the linear model we introduced in [47], and refer to Fig. 2 for a visual understanding of the following formulas.

We can then define the synergy term $s_{i,k}^j \in \mathbb{R}^+$ for task τ_i performed by the agent j while another agent executes task τ_k . To properly formulate it, we denote d_i^j and $d_i^j | k$ as the expected duration of i th task the j th agent performs without any other concurrent running task, and the duration under the constraint that another agent performs task τ_k . Note that if τ_i is performed by a robot agent and τ_k by the human agent, the difference between $d_i^j | k$ and d_i^j may be significant. The robot may slow down and even halt to preserve safety. Conversely, $d_i^j | k \equiv d_i^j$ if τ_i and τ_k are both performed by robot agents since we assume that the trajectories are synchronized.

Under such considerations, we denote $s_{i,k}^j$ as the synergy term:

$$s_{i,k}^j = \begin{cases} \frac{d_i^j | k}{d_i^j} & \text{if } \tau_i \text{ can be concurrent to } \tau_k, \text{ and } \tau_k \\ & \text{is executed by a human agent} \\ 1 & \text{otherwise.} \end{cases} \quad (9)$$

The synergy models the coupling effect between pairs of tasks. If τ_k causes a slowdown in the robot task (e.g., safety module intervention), then $s_{i,k}^j > 1$; while $s_{i,k}^j < 1$ if the parallelism has beneficial effects. The synergy value will be equal to 1 when parallelism cannot occur, or the coupling effect is neutral.

As shown in Fig. 2, the synergy does not apply linearly to the whole task but can modify the temporal overlapping of the task. Specifically, we denote $OV_{i,k}$ as the overlapping time defined as:

$$OV_{i,k} = \begin{cases} 0, & \text{if } \Delta T_{i,k} < 0 \\ \Delta T_{i,k}, & \text{otherwise} \end{cases} \quad \forall i, k \quad \forall k > i, \quad (10)$$

where:

$$\Delta T_{i,k} = \min(t_i^e, t_k^e) - \max(t_i^s, t_k^s). \quad (11)$$

We can, finally, see the measure of the overlapping term as the nominal one stretched by the synergy term:

$$\widetilde{OV}_{i,k} = s_{i,k}^j OV_{i,k} \quad (12)$$

2.5.2. Human-aware task allocation and scheduling (STP)

Given the definition of the scaled overlapping through synergy (12), we can define the performance constraints $C_{\text{performance}}$ which models the expected task end times based on the agent's performance and the impact of tasks executed in parallel. Specifically, it turns in:

$$C_{\text{performance}} := \begin{cases} \dots \\ t_i^e = t_i^s + \sum_{j \in \mathcal{A}} \tilde{d}_i^j a_i^j + \\ \quad \sum_{r \in \mathcal{A}^R} \sum_{k=1}^m \sum_{k \neq i} \widetilde{OV}_{i,k} (s_{i,k}^r - 1) a_i^r a_k^H \\ \dots \end{cases} \quad \forall i. \quad (13)$$

The task end time t_i^e is determined by the start time t_i^s , plus a first term that sets the nominal duration based on the selected agent and a second term that adjusts this duration based on parallelism with other tasks, considering the estimated synergy factors between parallel agents. Only the portion of the task that overlaps with the parallel task is affected by the coupling effect and is scaled according to the synergy value.

2.5.3. Relaxed human-aware task allocation and scheduling formulation (R-STP)

The introduction of synergy terms $s_{i,k}^r$ into the model involves a set of closely coupled constraints (8)–(13). Intuitively, this can create a loop effect where changes in overlapping affect the robot's task duration, which in turn alters the overlap again, making the problem computationally demanding. To address this issue, we propose an approximated model to reduce the computational complexity. We simplify the problem by decoupling the synergy from the task duration (13) while adding a penalty term to the cost function. Thus, we replace (13) with:

$$C_{\text{performance}} := \begin{cases} t_i^e = t_i^s + \sum_{j \in \mathcal{A}} \hat{d}_i^j a_i^j \\ \forall i \in \{1, \dots, m\} \end{cases} \quad (14)$$

and (2) with a proxy for the makespan:

$$\mathcal{M} = \max_{i=1, \dots, m} (t_i^e) + \Delta S \quad (15)$$

where:

$$\Delta S = \sum_{r \in \mathcal{A}^R} \sum_{i=1}^m \sum_{k=1}^m \sum_{k \neq i} \widetilde{OV}_{i,k} (s_{i,k}^r - 1) a_i^r a_k^H. \quad (16)$$

The rationale behind (15) is that the makespan is equal to the sum of the nominal duration of the tasks and the lengthening of the various tasks given by the coupling effect between the human and robots' tasks (ΔS). This makes it possible to switch from the nominal makespan (considering there is no effect between one agent and another) to the actual makespan that considers the reciprocal effect between humans and robots.

3. Learning of human-robot synergies

Constraints (13) and (14) need an estimate of the synergy term. The learning approach for human-robot synergies is based on the linear model we introduced in [47] that relates the measured duration of human tasks to the level of parallelism of each robot task through the synergy terms defined in (9).

Precisely, for each observation collected in the dataset \mathcal{D} , we extract the measures of: (i) the actual duration of each human task τ_k , denoted

as \tilde{d}_k^H ; (ii) the actual overlapping time between each pair of tasks τ_k and τ_i , denoted as $\tilde{OV}_{i,k}$; (iii) the actual idle time of the task τ_k , denoted as $\tilde{\Gamma}_k^H$. Given these measures, the following model holds for a generic pair of robot (r) and human:

$$\tilde{d}_k^H = \sum_{\substack{i=1 \\ i \neq k}}^m \left(\tilde{\Gamma}_k^H + \tilde{OV}_{i,k} \right) a_i^r, \quad \forall r \in \mathcal{A}^R. \quad (17)$$

This formulation is made only for the pair of human–robot agents as we are interested in the duration variation caused by the human operator on the robots' tasks.

Synergy coefficient values are estimated in terms of distribution probability following the Bayesian approach. To do so, we assume that the measurements \tilde{d}_k^H follow a normal distribution around the model in (17), and using (12) that relates the actual overlapping with the nominal overlapping given the generated plan through the synergies coefficients, we get:

$$d_k^H \sim \mathcal{N} \left(\sum_{\substack{i=1 \\ i \neq k}}^m \left(\tilde{\Gamma}_k^H + s_{i,k}^r OV_{i,k} \right) a_i^r, \sigma_m^2 \right). \quad (18)$$

To capture the uncertainty of the proposed model, we assume that the standard deviation σ_m follows a uniform distribution with lower bound l_b and upper bound u_b :

$$\sigma_m \sim \mathcal{U}(l_b, u_b). \quad (19)$$

The prior knowledge of the synergy terms, defined as ratios relative to the average duration of a task, inherently holds non-negative values. Hence, it is reasonable to assume a log-normal distribution, with μ_s and σ_s respectively, the mean and the standard deviation of the log of the distribution:

$$s_{i,k}^r \sim \text{LogN}(\mu_s, \sigma_s^2). \quad (20)$$

Given the above prior knowledge of the model, the posterior distribution of the estimated synergy parameters $S_k^r = (s_{1,k}^r, \dots, s_{m,k}^r)$ can be obtained through the application of Bayes' Theorem:

$$p(S_k^r | D) = \frac{p(D | S_k^r) p(S_k^r)}{p(D)}. \quad (21)$$

The posterior distribution is computed using a Markov-Chain Monte Carlo (MCMC) algorithm, such as No-U Turn Sampler (NUTS) [49] that gives an unbiased estimate on the limit. It is possible to update the prior knowledge distributions using measurements of task durations for both the person and the robot, along with the percentage of overlap. This can be done either after a batch realization of the tasks or after each execution. The method allows for progressive estimates of the probability distributions of human–robot synergies, making the model robust to stochastic human behavior and applicable in real-world cases.

4. Simulations

4.1. Framework architecture

The software architecture is derived from [47] and is improved here in the planning, dispatching, and statistics modules (see Fig. 3). The architecture is designed to manage the pipeline from planning to execution. At the highest level, *task planning module* solves the allocation and scheduling problem. The *task dispatching module* (Algorithm 1) handles the task execution request to the lower levels based on the obtained task plan. At the lowest level, there is a *task execution module* for each agent. The task executor grounds symbolic tasks into geometric targets, solves the motion planning problems, and monitors the execution. In simulation, the human agent is managed like a robotic agent, while in a real-world case study, the user sends the execution request to an HMI and waits for feedback.

The *task planner interface* is between the higher and lower levels. This module receives task execution requests from the dispatcher,

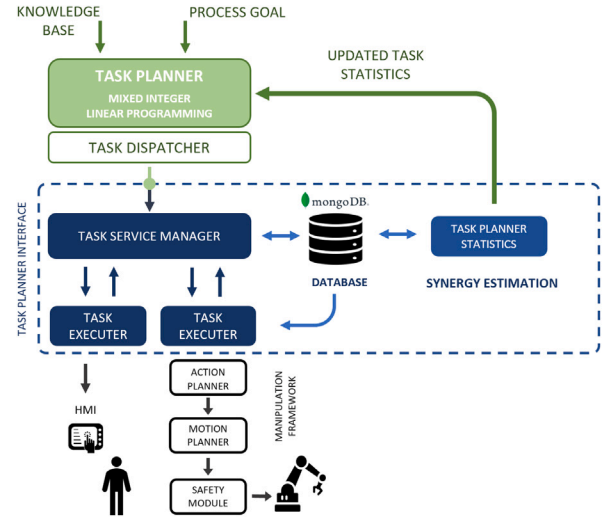


Fig. 3. Framework architecture: a three-tiered structure that manages the pipeline from planning to execution.

Algorithm 1 Task Dispatcher Algorithm

Require: $\pi = \{(t_i^s, t_i^e, a_i^j)\} \quad \forall i \in \{1, \dots, m\}, j \in \mathcal{A}$

Sort π based on t_i^s

while $\neg \text{isEmpty}(\pi)$ **do**

for $j \in \mathcal{A}$ **do**

π^j is the tasks in π for which $a_k^j = 1 \forall k$ -tasks

if $\text{isEmpty}(\pi^j)$ **then**

 continue

end if

$\tau_i \leftarrow \text{getFirstAction}(\pi^j)$

$t \leftarrow \text{getTime}()$ ▷ Get current simulation time

if $t \geq t_i^s$ & $\text{isFree}(\text{Agent}^j)$ **then**

 Send the execution request of task τ_i to agent j

$\pi.\text{pop}(\tau_i)$

$\text{setBusy}(\text{Agent}^j)$

end if

end for

end while

interacts with a MongoDB database [50], and stores information such as task start and completion time. The *task planner statistics module* is responsible for keeping up-to-date task execution statistics such as the expected duration and performing the synergy estimation between task pairs.

The MINLP models presented in Section 2 are integrated at the task planning level and implemented using GurobiPy [51]. In addition, the task dispatcher is implemented using the Algorithm 1 and exploits ROS-topic communication for task requests/responses. The Bayesian estimation of synergy terms proposed in Section 3 is implemented using Pyro [52] and integrated as a ROS-Service in the statistics module.

4.2. Simulation setup

The simulated HRC application involves pick-and-place operations for the composition of a mosaic. The application employs a UR5 collaborative robot mounted on a linear axis, working alongside a human operator. Each agent has its workspace where their respective boxes are placed: white boxes for the robot and orange boxes for the operator. The shared workspace contains blue cubes that both agents can manipulate.

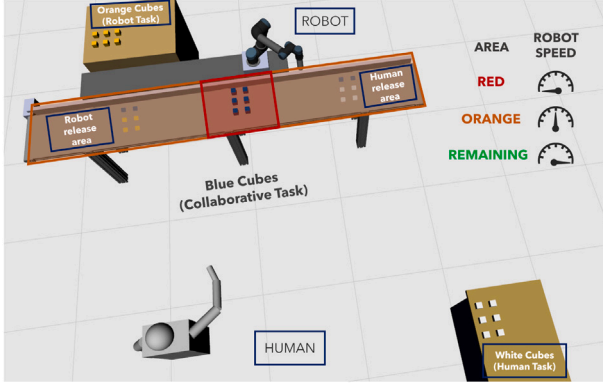


Fig. 4. Simulated setup. The robot workspace contains orange boxes, the human's white boxes, and the agent's release areas. Blue boxes are shared between agents. Highlighted are the safety areas used in scenario 1.

Table 2

SSM parameters description. ISO/TS 15066 specifies human's velocity v_h when not tracked. Robot and vision system reaction times are cumulated.

Symbol	Value	Description
v_h	1.6 m s^{-1}	Human velocity toward the robot
a_s	1 m s^{-2}	Max. robot deceleration
T_r	0.3 s	Max. system reaction time
C	0.2 m	Position uncertainty
S_d	Measured	Separation distance

The application was tested in two scenarios based on safety specifications aligned with ISO/TS 15066 [2].

The safety areas of scenario 1 (S1) are defined to comply with the *Speed And Separation Monitoring (SSM)* described in [2]. In Fig. 4, the safety areas of the proposed application are highlighted. When the human operator enters the orange area, the robot operates at 50% of its nominal speed. If the person enters the red area, the robot stops. In the remaining area, the robot moves at its nominal speed [53]. This safety implementation is typical of industrial applications as it can be achieved with safety-rated sensors (e.g., radar or laser barriers). The task-level goal for each agent is to compose a mosaic in their own releasing area using four proprietary boxes (white for the robot and orange for the human) and two shared boxes (blue ones) for a total of 24 tasks (12 picking and 12 placing).

Scenario 2 (S2) is based on the robot's velocity modulation to prevent it from colliding with a human. By continuously monitoring the minimum agent's distance, it is possible to calculate the maximum velocity of the robot toward the human ($v_{hr,max}$) following the definition of the minimum separation distance provided by the SSM (refer to [2]):

$$v_{hr,max}(t) \geq \sqrt{v_h^2 + (a_s T_r)^2 - 2a_s(C - S_d(t)) - a_s T_r - v_h} \quad (22)$$

in which the involved parameters are detailed in Table 2.

The task-level goal is for each agent to compose a mosaic in their own releasing area using four proprietary boxes for the robot (orange boxes), three proprietary boxes for the human operator (white boxes), and two shared cubes (blue ones).

In the first phase of simulations, 50 random plans were run to estimate the average duration of individual tasks and synergies. Once estimates were given, task planning was run. Regarding the parameters in Section 3, we ensured the median value of 1 of the synergy terms and $\sigma_s = 0.5$, while using $l_b = 0 \text{ s}$ and $u_b = 2 \text{ s}$ in (19).

4.3. Baseline and metrics

We compare our methods with two baselines: (i) *Baseline TP*, which is a simplified version of (1), and also similar to [47] (the coupling

effect between tasks pairs is not taken into account); (ii) the MILP model proposed in [43], where spatially close tasks are constrained not to be executed in parallel by the agents (referred to as *Not Neighboring TP*).

The evaluation of the proposed method against different task planners focused on performance and safety. For performance evaluation, we measure the plan execution duration as follows:

$$\text{Makespan} = \max_{i \in \{1, \dots, m\}} (\tilde{t}_i^e) \quad (23)$$

where \tilde{t}_i^e is the measured end time of a task τ_i in a task set \mathcal{T} during the execution of a plan π . As for safety, we monitor with the vision system the minimum distance (D_{min}) between the human operator and the robot during the execution of the plans. Then, we compute the cumulative probability distribution of such distance over all the executions of each method as:

$$\frac{1}{N_{\text{plans}}} \sum_{k=1}^{N_{\text{plans}}} \mathbb{P}(D_{min} \leq \tilde{d}) \quad \forall \tilde{d} \in [0, d_{max}]. \quad (24)$$

Thus, this index represents the probability that D_{min} falls below a given distance \tilde{d} (between 0 and the maximum distance d_{max} of the vision system) for the method at hand.

4.4. Results

First, we evaluate the estimated synergies for both scenarios.

The heat maps in Figs. 5(a) and 6(a) show the synergy for each robot task (matrix rows) and for each concurrent human task (matrix columns). A synergy greater than one means that the robot task takes more time than its average duration (thus, the coupling penalizes the plan execution). An index smaller than one implies that the task takes less time than its average (leading to a more efficient plan). For Scenario 1 in Fig. 5(a), the most significant slowdown occurs when the human operator places the shared boxes (blue boxes) in their release area. The start point causes the robot to stop at the execution start since the static point selection impacts the whole agent's trajectory.

The coupling effects between tasks also arise in Scenario 2, where the scaling factor acts more smoothly, changing continuously based on (22). The results of the synergy estimation of this scenario are reported in Fig. 6(a). The first row of the heat map shows that the tasks of picking/positioning shared boxes (blue boxes) by humans cause similar slowdowns of the robot, unlike the case of parallelism with picking/positioning of humans' boxes (white boxes), which causes faster execution than the average duration. The robot's task of picking the orange object is unaffected by any human tasks; vice versa, the robot's task of placing the blue box is significantly slowed down, except when the human performs the white box placement.

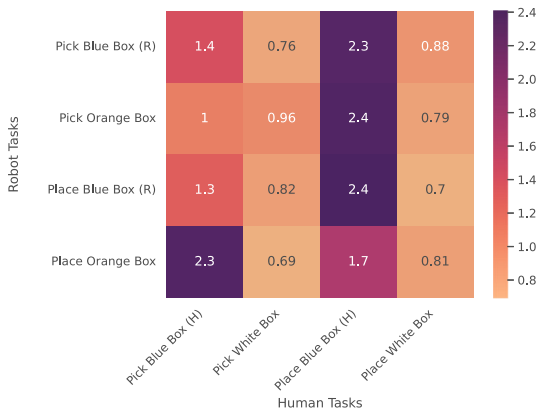
Table 3 compares the makespan (23) and the synergy of executed plans. Figs. 5 and 6 show the results of the online simulations of Scenarios 1 and 2. Figs. 5(b) and 6(b) compare each planner's makespan in 50 executions. In both scenarios, STP outperforms the other methods by reducing the makespan of about 18% w.r.t. Baseline TP and 13% w.r.t. Not Neighboring TP. The R-STP shows an average plan execution time comparable with the best case of the Baseline and the Not Neighboring TP; the average makespan obtained with the Relaxed-STP method outperforms the Baseline (reduction of about 13% for both scenarios) and the solutions obtained from Not Neighboring TP (reduction about 7% for both scenarios).

Figs. 5(c) and 6(c) show the average cumulative probability distribution of the minimum human-robot distance (24), measured during the execution of the plan. In Scenario 1, the STP and Relaxed-STP show a cumulative probability distribution comparable with the Not Neighboring TP method for distances about 0.8 m, with a minimum human-robot distance of about +0.4 m greater than that of the baseline case (0.04 m). Relaxed-STP has the lowest probability of having samples below 0.8 m. In Scenario 2, these differences are more evident

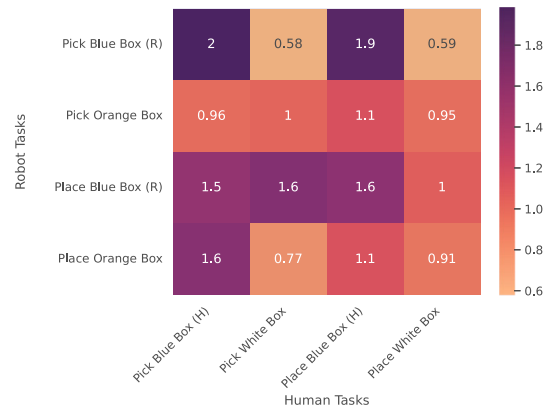
Table 3

Comparison of task planners: performance summary of optimized plans. The term ΔS is computed according to the definition in (9). The average, minimum, and maximum values are calculated over all N_{plans} . Scenario 1 with the safety areas is referred as S1, while Scenario 2 with dynamic speed scaling as S2.

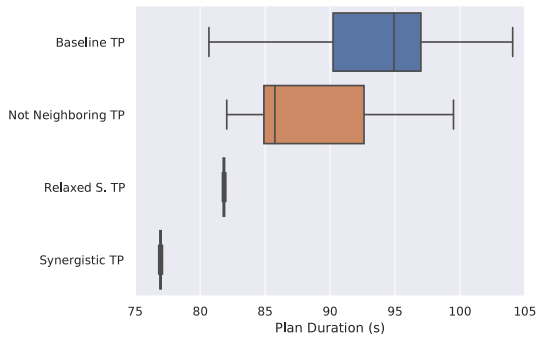
Method	Makespan (s)		Average ΔS (s)		Min ΔS (s)		Max ΔS (s)	
	S1	S2	S1	S2	S1	S2	S1	S2
Baseline TP	81.80	68.34	8.90 ± 7.80 (95%)	5.05 ± 13.67 (95%)	0.16	-6.63	14.96	16.37
Not Neighboring TP	81.80	68.34	1.93 ± 6.00 (95%)	-4.01 ± 3.75 (95%)	-4.00	-7.20	9.66	-1.56
Relaxed-STP	81.80	68.34	-4.65431	-8.76	-4.65	-8.76	-4.65	-8.76
Synergistic TP	79.61	62.11	-3.94048	-6.29	-3.94	6.29	-3.94	-6.29



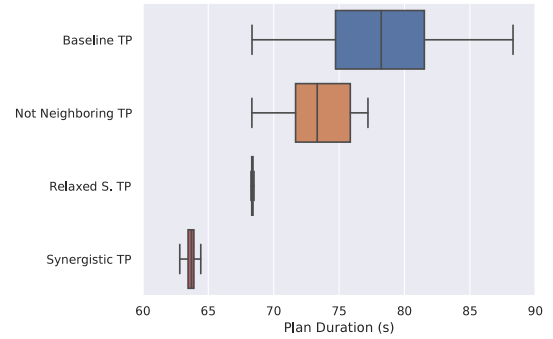
(a) Estimated synergy terms (robot side) in matrix format



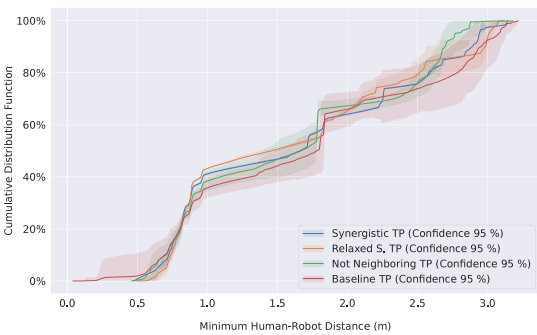
(a) Estimated synergy terms (robot side) in matrix format



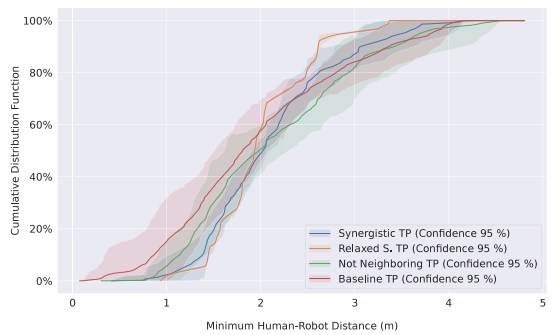
(b) Plan execution duration



(b) Plan execution duration



(c) Cumulative probability of the minimum human-robot distance



(c) Cumulative probability of the minimum human-robot distance

Fig. 5. Results of Scenario 1: static safety zones.

Fig. 6. Results of Scenario 2: safety with dynamic speed scaling.

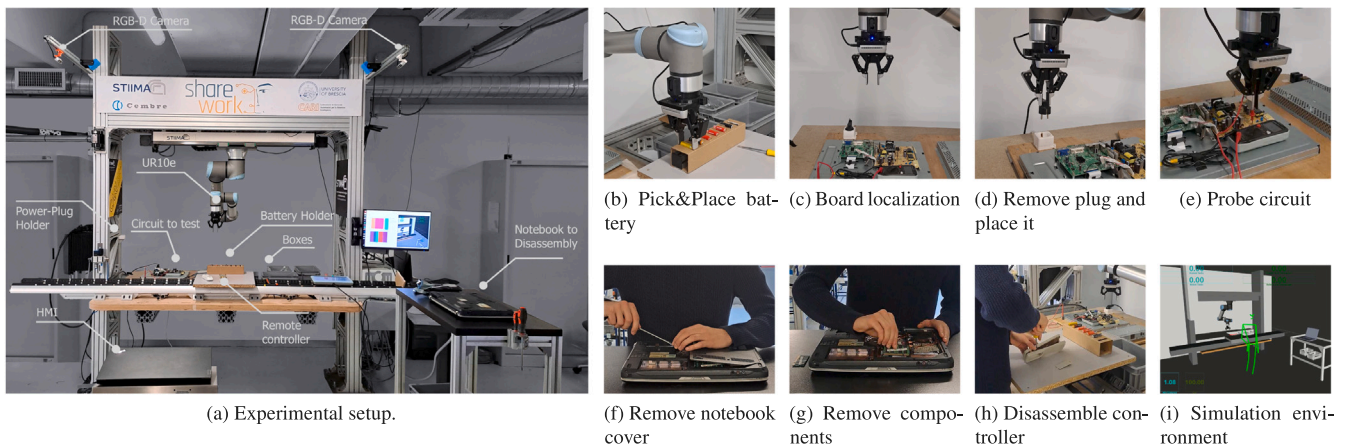


Fig. 7. Experimental tests for the disassembly and recycling of e-waste in HRC set. In (a), the experimental setup is shown, (b)–(e) show the robot tasks, (f)–(h) show the human operator tasks, and (i) shows the simulation environment.

for larger distances. Relaxed-STP is the least likely to have distances below 1.8m, followed by STP. Overall, Relaxed-STP can be regarded as the safest method, with a minimum distance +0.86 m greater than Baseline TP (0.07 m), followed by STP (+0.34 m) and Not Neighboring TP with minimum distances greater than about +0.23 m than Baseline TP, respectively.

4.5. Discussion on computational complexity

Although this paper is not focused on providing real-time solutions, we discuss the proposed model’s computational complexity. Indeed, the time frame for obtaining a solution must align with the industrial scenario in which the planner computes a solution, and the resulting plan is subsequently put into execution. The MINLP is generally \mathcal{NP} -hard, and the curse of dimensionality of integer variables (e.g., increasing the number of tasks and agents) and handling nonlinearities can lead to suboptimal solutions with limited computation. For real-world problems such as those presented in the article, the solution consists of tuning the two parameters of the solver. The first is the MIP optimality gap, which serves as an indicator of solution quality. It is the relative ratio between the best-known bound on the objective function and the incumbent solution’s objective value. This value is equal to 0 when is guaranteed that the solution is optimal. The second parameter is the maximum available time, depending on the available computation.

Table 4 shows the parameters used for the experiments and the solution time achieved within the two simulated scenarios using Gurobi [51] running on a notebook equipped with an Intel i7-1165G7 2.8 GHz 4-core CPU and 16 GB of RAM. The number of tasks involved in the two scenarios is about 20 and is compatible with use cases of industrial interest and other research work in the field of planning for HRC. Moreover, the achieved computation time is compatible with the “offline” planning scenario. If the number of tasks and agents increases significantly, is possible to use hierarchical planning [54] (i.e., group atomic actions into more complex tasks) or use heuristic methods to speed up the search for approximate solutions [55].

5. Case study

We consider an E-waste disassembly and testing case study inspired to the ShareWork EU project industrial use case [39]. We use the collaborative cell in Fig. 7(a), which consists of a Universal Robots UR10e mounted upside down. The cell was designed within the EU

Table 4

Optimization Parameters of proposed methods. The timeout time is set to 60 s in both Relaxed-STP, while it is set to 240 s in both Synergistic TP.

Method	MIP-GAP		Solution Time	
	S1	S2	S1	S2
Relaxed-STP	1.70 %	2 %	18 s	42 s
Synergistic TP	8 %	10.8 %	192 s	240 s

project ShareWork. Two Realsense D435 cameras operating at 30 Hz monitor the shared workspace between humans and robots. These cameras provide the human tracking system with RGB-Depth frames. Drawing upon established research findings in this field [56–59], the human pose estimation system uses Mediapipe BlazePose [60] for skeleton tracking on RGB frame, reconstructs the 3D pose with the depth map using the pin-hole model, and applies a Kalman filter (code available [61]) to reduce noise.

The experiments are also summarized in the attached video.

5.1. E-waste disassembly and testing scenario

The tasks assigned to the agents in the applied scenario are illustrated in Fig. 7. The robot is tasked with testing an electrical circuit, which involves locating the board for inspection, probing the circuit with a multimeter probe, disconnecting the power socket, and placing it in a suitable holder. In addition, the robot must place two batteries from the battery holder in an appropriate box (for recovery or disposal). The precedence constraint for the robot is the localization task, which must be completed before any tasks involving the board. Concurrently, a human operator is responsible for disassembling a laptop: first, removing the covers and, subsequently, extracting the components. He/she must also disassemble two remote controls and store the batteries in the battery holder, from which the robot will retrieve them two times. The remote controller and batteries are placed in a workspace shared with the robot. The total number of tasks is 9; each test takes about 4 min. The process was executed 30 times with random plans (generated by Baseline TP). Fig. 8(a) reports the results of such estimation.

5.2. Experiments protocol

The experiments involved 15 participants, including master’s students, Ph.D. students, researchers, and other workers, aged between 19 and 31. After initial training on the tasks to execute, the participants

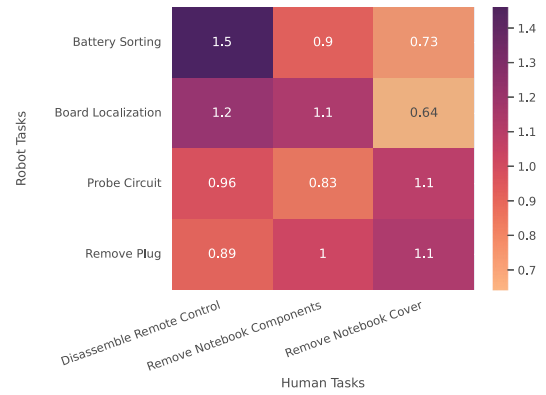
performed five plans for each method in randomized order. While executing the plans, we collected task duration and the human–robot distance. At the end of each batch of executions, the subjects filled out a questionnaire to assess their subjective evaluation of the cooperation with the robot. The questions are in Table 5 and relate to overall satisfaction (Q1), safety (Q2), human–robot distance (Q3), idle time (Q4), perceived fluency assessment (Q5) and robot interruptions (Q6). The questions are partially inspired by the questionnaire of [62,63], adapting and expanding them to suit the specific case under analysis. The questionnaire and the related statistical analysis are not meant to be an exhaustive evaluation but aim to provide a preliminary assessment of the workers’ perceived impact. This complements the quantitative evaluation with an initial understanding of the human experience with the proposed method. The questionnaire follows a five-option response format (from “strongly disagree” to “strongly agree”).

5.3. Results

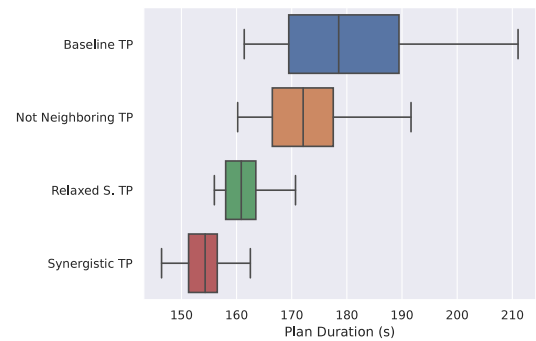
Fig. 8(b) compares execution times. Both the proposed methods lead to shorter execution times. STP (154.6s on average) achieved a reduction of 14.7% compared to the Baseline TP (181.2s on average), and 10.9% compared to the Not Neighboring TP (173.5s on average). Similarly, Relaxed-STP (161.5s) achieved a reduction of 10.9% compared to the Baseline TP and 7.0% compared to the Not Neighboring TP.

Fig. 8(c) compares the human–robot distance measured during the executions; for each human–robot distance, it shows the percentage of samples below that distance measured during the runs. The results show that the proposed methods are less likely to cause small human–robot distances. By taking 0.4 m as an example of risky distance, STP and Relaxed-STP have a percentage of 0.85% and 0.90% samples below such threshold. In contrast, Not Neighboring TP has 6.75% and Baseline TP has 15.95%. It should be noted that the percentage of samples obtained below this distance with the proposed methods is comparable to that of false positives detected by Mediapipe.

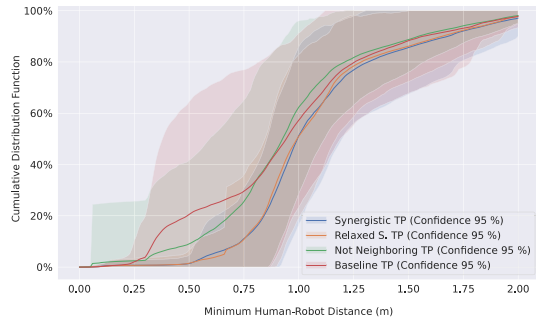
Fig. 9 displays the user responses to the questionnaire, presenting the response counts for each method in a stacked fashion. Moreover, a probability density estimation of the user responses is superimposed for each method. The users’ responses were statistically analyzed to test the null hypothesis among the methods: the Kruskal–Wallis Test [64] was performed for each item to assess whether statistically significant differences exist among the response populations corresponding to the four methods. The test results are summarized in the last two columns of Table 5. The four task planning methods are statistically equivalent for Q2 (perceived safety) and Q4 (idle time). The users’ responses concerning safety have a median value of “agree” for all four methods, and this suggests that the low-level safety module makes the operator feel safe, regardless of the task planning method. Regarding idle times, the responses have a median value of “neutral”, except for the Not Neighboring TP, with a median value of “agree”. Statistical differences ($p < 0.05$) between methods were observed for overall satisfaction (Q1), perceived human–robot proximity (Q3), fluency (Q5), and robot interruptions (Q6). Additionally, we examined the presence of statistical differences between the pairs of methods we proposed and the comparative methods. Test results are reported in the first four columns of the table (p -values), and in all cases, statistical differences ($p < 0.05$) were identified in favor of the proposed methods: Q1 has a median value of “agree” for the proposed methods and “neutral” for the baselines; Q3 has a median of “strongly disagree” with STP, “disagree” with Relaxed-STP, and “neutral” for both baselines; as for Q5 both proposed methods have a median of “strongly agree”, while Baseline TP scored “neutral” and Not Neighboring TP scored “agree”; finally, Q6 has a median of “strongly disagree” for STP, “disagree” for Relaxed-STP, “neutral” for Not Neighboring TP, and “agree” for Baseline TP.



(a) Estimated synergy matrix.



(b) Plan execution duration.



(c) Cumulative probability of the minimum human–robot distance

Fig. 8. Results of the real-world use case scenario.

6. Conclusions

The paper proposes a novel human-aware task allocation and scheduling model based on learned human–robot synergy. Synergy learning enables the evaluation of coupling effects in concurrent task execution by the agents. Integrating synergy values into the allocation and scheduling model allows for more efficient and safe solutions as early as the task planning stage. The simulations and the case study experiments confirmed the effectiveness of the proposed models. This emerged from both quantitative metrics and subjective user evaluations: the results demonstrate that the proposed methods lead to a reduction in average execution time and a decreased likelihood of small human–robot distances; statistical analysis of user responses revealed a preference for the proposed methods, noting improvements in proximity management, system fluidity, reduced robot interruptions, and overall satisfaction. Future works will focus on the integration of risk assessment results into the proposed model: while synergies

Table 5

Statistical results of Kruskal–Wallis Test on user's questionnaire responses. Method A is the Baseline TP; Method B is the Not Neighboring TP; Method C is the Relaxed-STP; Method D is the Synergistic TP. Cells in the table are highlighted in green if p -values < 0.05 (statistically significant difference detected), in red if statistically equivalent (p -values > 0.05).

Questions	Method C vs A	Method C vs B	Method D vs A	Method D vs B	Statistical difference	Kruskal–Wallis Test (p -value)
Q1: I am satisfied with the coexistence with the robot to execute the process.	2.3e-04	2.2e-03	1.1e-04	6.5e-04	True	1.2e-05
Q2: I felt safe working with the robot.	7.0e-01	9.1e-02	2.6e-01	3.8e-02	False	1.1e-01
Q3: I felt I was getting too close to the robot to perform my tasks.	3.9e-02	1.1e-02	3.9e-02	1.1e-02	True	1.1e-02
Q4: Idle times bother me.	2.3e-01	1.3e-01	4.0e-01	2.1e-01	False	3.5e-01
Q5: The coexistence with the robot during the execution of the process was fluent.	8.6e-05	2.1e-03	5.9e-05	7.1e-04	True	5.4e-06
Q6: The robot often stopped because you are in the area near the robot	1.2e-05	1.1e-04	4.8e-06	1.5e-05	True	1.2e-08

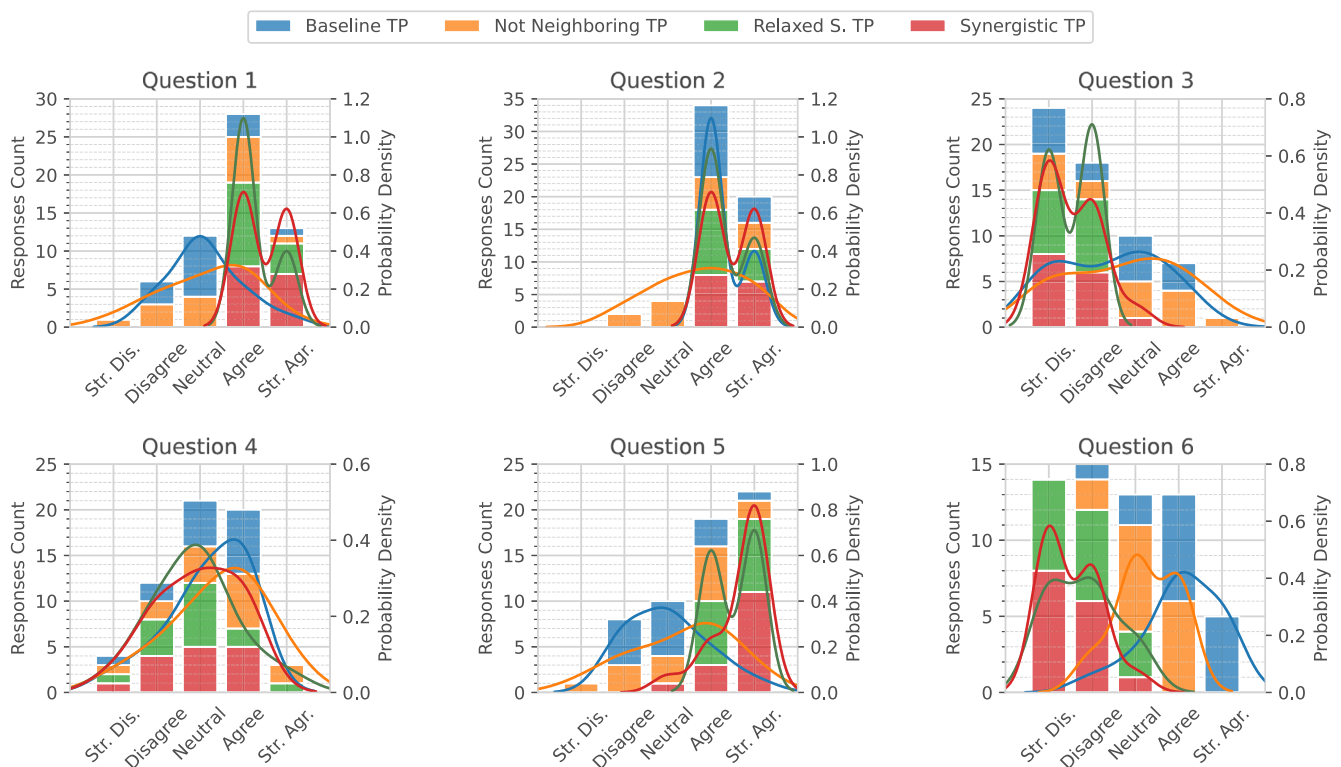


Fig. 9. Questionnaire responses for each question. The x -axis displays user responses ranging from “strongly disagree” to “strongly agree”. Responses counts of each method are stacked, with solid lines representing the estimated continuous probability densities. The full text of the questions is provided in Table 5.

capture the quantitative impact of human–robot collaboration, the introduction of a penalty matrix, derived from risk evaluations provided by safety experts, would enable the model to account for task-related risks accounting to safety specifications. Moreover, to transition from offline task planning to replanning, a fast heuristic method for task plan adaptation based on synergies will be investigated to tackle unexpected human behavior.

CRediT authorship contribution statement

Samuele Sandrini: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Marco Faroni:** Writing – review & editing, Writing – original draft, Validation, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Nicola Pedrocchi:**

Writing – review & editing, Writing – original draft, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was partially carried out within the MICS (Made in Italy Circular and Sustainable) Extended Partnership and received funding from Next-Generation EU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1551.11-10-2022, PE00000004) CUP MICSD43C22003120001.

Appendix. Linear representation of proposed methods

This section presents a linearization of the proposed MINLP models introduced in Section 2. The nonlinearity of the model in Section 2.5.2 is primarily related to the cost function, which can be linearized by introducing an auxiliary variable $t^{e-\max}$, i.e., the makespan, with the following linear constraints:

$$C_{\text{makespan}} := t^{e-\max} \geq t_i^e \quad \forall i \in \{1, \dots, m\} \quad (\text{A.1})$$

Thus, the nonlinear cost function in (2) is replaced by:

$$\mathcal{M} = t^{e-\max} \quad (\text{A.2})$$

The overlapping definition (10) needs to introduce a couple of additional continuous variables (maximum start-time $t_{i,k}^{s-\max}$ and the minimum stop-time $t_{i,k}^{e-\min}$ of the task pair) end three auxiliary binary variables ($\sigma_{i,k}^I$, $\sigma_{i,k}^{II}$ and $\sigma_{i,k}^{III}$) for each task couple (τ_i, τ_k) :

$$C_{\text{max-start}} := t_{i,k}^{s-\max} \geq t_i^s \quad (\text{A.3})$$

$$t_{i,k}^{s-\max} \geq t_k^s \quad (\text{A.4})$$

$$t_{i,k}^{s-\max} \leq t_i^s + M(1 - \sigma_{i,k}^I) \quad (\text{A.5})$$

$$t_{i,k}^{s-\max} \leq t_k^s + M(\sigma_{i,k}^I) \quad (\text{A.6})$$

$$C_{\text{min-end}} :=$$

$$t_{i,k}^{e-\min} \leq t_i^e \quad (\text{A.7})$$

$$t_{i,k}^{e-\min} \leq t_k^e \quad (\text{A.8})$$

$$t_{i,k}^{e-\min} \geq t_i^e - M(1 - \sigma_{i,k}^{II}) \quad (\text{A.9})$$

$$t_{i,k}^{e-\min} \geq t_k^e - M(\sigma_{i,k}^{II}) \quad (\text{A.10})$$

$$C_{\text{overlapping}} :=$$

$$OV_{i,k} \leq (t_{i,k}^{e-\min} - t_{i,k}^{s-\max}) + M\sigma_{i,k}^{III} \quad (\text{A.11})$$

$$OV_{i,k} \leq M(1 - \sigma_{i,k}^{III}) \quad (\text{A.12})$$

$$OV_{i,k} \geq -M(1 - \sigma_{i,k}^{III}) \quad (\text{A.13})$$

$$OV_{i,k} \geq 0 \quad (\text{A.14})$$

$$\forall i, k \in \{1, \dots, m\}, k \neq i$$

Thus, (10)–(11) are replaced by (A.11)–(A.14). The bilinear term in (13) can be linearized as follows:

$$t_i^e = t_i^s + \sum_{j \in \mathcal{A}} \hat{a}_i^j \alpha_i^j + \sum_{r \in \mathcal{A}^R} \sum_{k \neq i} \alpha_{i,k}^r (s_{i,k}^r - 1) \quad \forall i \in \{1, \dots, m\} \quad (\text{A.15})$$

where $\alpha_{i,k}^r$ is an additional binary variable that selects whether or not to apply the additional term to the task duration:

$$C_{\text{adaptation-term}} :=$$

$$\alpha_{i,k}^r \leq OV_{i,k} + M \left(2 - a_k^H - \sum_{j \in \mathcal{A}^R} \alpha_j^j \right) \quad (\text{A.16})$$

$$\alpha_{i,k}^r \geq OV_{i,k} - M \left(2 - a_k^H - \sum_{j \in \mathcal{A}^R} \alpha_j^j \right) \quad (\text{A.17})$$

$$\alpha_{i,k}^r \leq M \alpha_i^r \quad (\text{A.18})$$

$$\alpha_{i,k}^r \geq -M \alpha_i^r \quad (\text{A.19})$$

$$\forall i, k \in \{1, \dots, m\}, k \neq i, \forall r \in \mathcal{A}^R$$

In this case, (A.16)–(A.19) are in charge of applying the overlapping effect only if the (i, k) task couple is executed by a robot and a human, respectively.

The same approach can be followed to linearize the model 2.5.3. Thus, by adding (A.1) to the model, it is possible to rewrite the nonlinear cost function in (15) with:

$$\mathcal{M} = t^{e-\max} + \Delta S \quad (\text{A.20})$$

Moreover, by adding the constraints in (A.3)–(A.14) is possible to replace the nonlinear definition of ΔS in (16) with:

$$\Delta S = \sum_{r \in \mathcal{A}^R} \sum_{i=1}^m \sum_{k \neq i}^m \alpha_{i,k}^r (s_{i,k}^r - 1) \quad (\text{A.21})$$

Appendix B. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.rcim.2025.103006>.

Data availability

A project webpage with supplementary material, including datasets, code, and videos, is available at: https://jrl-cari-cnr-unibs.github.io/synergistic_hrtp/.

References

- [1] R. Awad, M. Fechter, J. van Heerden, Integrated risk assessment and safety consideration during design of HRC workplaces, in: IEEE Int Conf on Emerging Technologies and Factory Automation, ETFA, 2017, pp. 1–10.
- [2] Technical Committee ISO/TC 299, ISO/TS 15066-2016 Robots and robotic devices: Collaborative robots, Standard, vol. 2016, International Organization for Standardization, Geneva, CH, 2016.
- [3] P. Franceschi, N. Pedrocchi, M. Beschi, Adaptive impedance controller for human-robot arbitration based on cooperative differential game theory, in: IEEE Int Conf on Robotics and Automation, ICRA, 2022, pp. 7881–7887.
- [4] P. Song, Y. Yu, X. Zhang, A tutorial survey and comparison of impedance control on robotic manipulation, *Robotica* 37 (5) (2019) 801–836.
- [5] M. Faroni, M. Beschi, N. Pedrocchi, Safety-aware time-optimal motion planning with uncertain human state estimation, *IEEE Robot. Autom. Lett.* 7 (4) (2022) 12219–12226.
- [6] R. Laha, W. Wu, R. Sun, N. Mansfeld, L.F. Figueredo, S. Haddadin, S²: On safe and time efficient robot motion planning, in: IEEE Int Conf on Robotics and Automation, ICRA, 2023, pp. 12758–12764.
- [7] C. Tonola, M. Faroni, M. Beschi, N. Pedrocchi, Anytime informed multi-path replanning strategy for complex environments, *IEEE Access* (2023).
- [8] M. Faroni, A. Umbrico, M. Beschi, A. Orlandini, A. Cesta, N. Pedrocchi, Optimal task and motion planning and execution for multiagent systems in dynamic environments, *IEEE Trans. Cybern.* (2023) 1–12.
- [9] K.A. Demir, G. Döven, B. Sezen, Industry 5.0 and human-robot co-working, *Procedia Comput. Sci.* 158 (2019) 688–695.
- [10] M.H. Christoph Petzoldt, M. Freitag, Review of task allocation for human-robot collaboration in assembly, *Int. J. Comput. Integr. Manuf.* 36 (11) (2023) 1675–1715.
- [11] L. Johannsmeier, S. Haddadin, A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes, *IEEE Robot. Autom. Lett.* 2 (1) (2016) 41–48.
- [12] L. Homem de Mello, A. Sanderson, AND/OR graph representation of assembly plans, *IEEE Trans. Robot. Autom.* 6 (2) (1990) 188–199.
- [13] E. Lamon, A. De Franco, L. Peternel, A. Ajoudani, A capability-aware role allocation approach to industrial assembly tasks, *IEEE Robot. Autom. Lett.* 4 (4) (2019) 3378–3385.
- [14] E. Merlo, E. Lamon, F. Fusaro, M. Lorenzini, A. Carfi, F. Mastrogiovanni, A. Ajoudani, Dynamic human-robot role allocation based on human ergonomics risk prediction and robot actions adaptation, in: IEEE Int Conf on Robotics and Automation, ICRA, 2022, pp. 2825–2831.
- [15] E. Merlo, E. Lamon, F. Fusaro, M. Lorenzini, A. Carfi, F. Mastrogiovanni, A. Ajoudani, An ergonomic role allocation framework for dynamic human–robot collaborative tasks, *J. Manuf. Syst.* 67 (2023) 111–121.
- [16] B. Busch, M. Toussaint, M. Lopes, Planning ergonomic sequences of actions in human-robot interaction, in: IEEE Int Conf on Robotics and Automation, ICRA, 2018, pp. 1916–1923.
- [17] I.E. Makrini, K. Merckaert, J.D. Winter, D. Lefeber, B. Vanderborght, Task allocation for improved ergonomics in human-robot collaborative assembly, *Interact. Stud.* 20 (1) (2019) 102–133.

- [18] F. Fusaro, E. Lamon, E. De Momi, A. Ajoudani, An integrated dynamic method for allocating roles and planning tasks for mixed human-robot teams, in: IEEE 30th IEEE Int Conf on Robot & Human Interactive Communication, RO-MAN, 2021, pp. 534–539.
- [19] M. Iovino, E. Scukins, J. Styrud, P. Ögren, C. Smith, A survey of behavior trees in robotics and ai, *Robot. Auton. Syst.* 154 (2022) 104096.
- [20] A. Monguzzi, M. Badawi, A.M. Zanchettin, P. Rocco, A mixed capability-based and optimization methodology for human-robot task allocation and scheduling, in: IEEE Int Conf on Robot and Human Interactive Communication (RO-MAN), 2022, pp. 1271–1276.
- [21] H.W. Kuhn, The hungarian method for the assignment problem, *Naval Res. Logist.* 52 (1) (2005) 7–21.
- [22] C.T. Landi, V. Villani, F. Ferraguti, L. Sabattini, C. Secchi, C. Fantuzzi, Relieving operators' workload: Towards affective robotics in industrial scenarios, *Mechatronics* 54 (2018) 144–154.
- [23] A. Pupa, C.T. Landi, M. Bertolani, C. Secchi, A dynamic architecture for task assignment and scheduling for collaborative robotic cells, in: *Human-Friendly Robotics 2020: 13th International Workshop*, Springer, 2021, pp. 74–88.
- [24] S. Zhang, Y. Chen, J. Zhang, Y. Jia, Real-time adaptive assembly scheduling in human-multi-robot collaboration according to human capability, in: IEEE Int Conf on Robotics and Automation, ICRA, 2020, pp. 3860–3866.
- [25] A. Pupa, W. Van Dijk, C. Secchi, A human-centered dynamic scheduling architecture for collaborative application, *IEEE Robot. Autom. Lett.* 6 (3) (2021) 4736–4743.
- [26] A. Pupa, W. Van Dijk, C. Brekelmans, C. Secchi, A resilient and effective task scheduling approach for industrial human-robot collaboration, *Sensors* 22 (13) (2022) 4901.
- [27] M. Faroni, M. Beschi, S. Ghidini, N. Pedrocchi, A. Umbrico, A. Orlandini, A. Cesta, A layered control approach to human-aware task and motion planning for human-robot collaboration, in: IEEE Int Conf on Robot and Human Interactive Communication, RO-MAN, 2020, pp. 1204–1210.
- [28] C.R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L.P. Kaelbling, T. Lozano-Pérez, Integrated task and motion planning, *Annu. Rev. Control. Robot. Auton. Syst.* 4 (2021) 265–293.
- [29] A.T. Le, P. Kratzer, S. Hagenmayer, M. Toussaint, J. Mainprice, Hierarchical human-motion prediction and logic-geometric programming for minimal interference human-robot tasks, in: IEEE Int. Conf. on Robot & Human Interactive Communication (RO-MAN), 2021, pp. 7–14.
- [30] R. Alami, A. Clodic, V. Montreuil, E.A. Sisbot, R. Chatila, Toward human-aware robot task planning, in: AAAI Spring Symposium: To Boldly Go where No Human-Robot Team Has Gone before, 2006, pp. 39–46.
- [31] I. Georgievski, M. Aiello, HTN planning: Overview, comparison, and beyond, *Artificial Intelligence* 222 (2015) 124–156.
- [32] L. De Silva, R. Lallement, R. Alami, The HATP hierarchical planner: Formalisation and an initial study of its usability and practicality, in: IEEE/RSJ Int Conf on Intelligent Robots and Systems, IROS, 2015, pp. 6465–6472.
- [33] R. Lallement, L. De Silva, R. Alami, HATP: An HTN planner for robotics, 2014, arXiv:1405.5345.
- [34] R. Lallement, L. de Silva, R. Alami, HATP: hierarchical agent-based task planner, in: Int Conf on Autonomous Agents and Multiagent Systems, AAMAS, 2018.
- [35] G. Buisan, R. Alami, A human-aware task planner explicitly reasoning about human and robot decision, action and reaction, in: Companion of the 2021 ACM/IEEE Int Conf on Human-Robot Interaction, 2021, pp. 544–548.
- [36] A. Favier, S. Shekhar, R. Alami, Anticipating false beliefs and planning pertinent reactions in human-aware task planning with models of theory of mind, in: International Conference on Automated Planning and Scheduling, ICAPS 2023, 2023.
- [37] A. Ham, M.-J. Park, Human-robot task allocation and scheduling: Boeing 777 case study, *IEEE Robot. Autom. Lett.* 6 (2) (2021) 1256–1263.
- [38] A. Umbrico, M. Anasagasti, S.-O. Bezrucav, F. Canale, A. Cesta, B. Corves, N. Mandischer, M. Mondragon, C.N. Rappis, A. Orlandini, Enhanced cognition for adaptive human-robot collaboration, in: IEEE Int Conf on Emerging Technologies and Factory Automation, ETFA, 2022, pp. 1–7.
- [39] A. Umbrico, A. Orlandini, A. Cesta, M. Faroni, M. Beschi, N. Pedrocchi, A. Scala, P. Tavormina, S. Koukas, A. Zalonis, et al., Design of advanced human-robot collaborative cells for personalized human-robot collaborations, *Appl. Sci.* 12 (14) (2022) 6839.
- [40] Y. Liu, Z. Jiang, C. Ke, A task allocation method in human-robot collaboration (HRC) for the disassembly of automotive traction batteries, in: IEEE Int Conf on Cyber-Physical Social Intelligence, ICCSI, 2022, pp. 332–337.
- [41] K. Bogner, U. Pferschy, R. Unterberger, H. Zeiner, Optimised scheduling in human-robot collaboration—a use case in the assembly of printed circuit boards, *Int. J. Prod. Res.* 56 (16) (2018) 5522–5540.
- [42] Y.Y. Liao, K. Ryu, Task allocation in human-robot collaboration (HRC) based on task characteristics and agent capability for mold assembly, *Procedia Manuf.* 51 (2020) 179–186.
- [43] M. Lippi, A. Marino, A mixed-integer linear programming formulation for human multi-robot task allocation, in: IEEE Int Conf on Robot & Human Interactive Communication, RO-MAN, 2021, pp. 1017–1023.
- [44] M. Lippi, P. Di Lillo, A. Marino, A task allocation framework for human multi-robot collaborative settings, in: IEEE Int Conf on Robotics and Automation, ICRA, 2023, pp. 7614–7620.
- [45] A. Pupa, C. Secchi, A safety-aware architecture for task scheduling and execution for human-robot collaboration, in: IEEE/RSJ Int Conf on Intelligent Robots and Systems, IROS, 2021, pp. 1895–1902.
- [46] M. Faccio, I. Granata, R. Minto, Task allocation model for human-robot collaboration with variable cobot speed, *J. Intell. Manuf.* (2023) 1–14.
- [47] S. Sandrini, M. Faroni, N. Pedrocchi, Learning action duration and synergy in task planning for human-robot collaboration, in: IEEE Int Conf on Emerging Technologies and Factory Automation, ETFA, 2022, pp. 1–6.
- [48] I. Griva, S. Nash, A. Sofer, *Linear and Nonlinear Optimization*, Second Edition, Society for Industrial and Applied Mathematics (SIAM), 3600 Market Street, Floor 6, Philadelphia, PA 19104, 2009.
- [49] M.D. Hoffman, A. Gelman, et al., The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo, *J. Mach. Learn. Res.* 15 (1) (2014) 1593–1623.
- [50] C. Györfödi, R. Györfödi, G. Pecherle, A. Olah, A comparative study: MongoDB vs. MySQL, in: Int Conf on Engineering of Modern Electric Systems, EMES, 2015, pp. 1–6.
- [51] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2023, URL <https://www.gurobi.com>.
- [52] E. Bingham, J.P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletos, R. Singh, P.A. Szerlip, P. Horsfall, N.D. Goodman, Pyro: Deep universal probabilistic programming, *J. Mach. Learn. Res.* 20 (2019) 28:1–28:6.
- [53] P. Karagiannis, N. Kousi, G. Michalos, K. Dimoulas, K. Mparis, D. Dimosthenopoulos, Ö. Tokçalar, T. Guasch, G.P. Gerio, S. Makris, Adaptive speed and separation monitoring based on switching of safety zones for effective human robot collaboration, *Robot. Comput.-Integr. Manuf.* 77 (2022) 102361.
- [54] P. Bercher, R. Alford, D. Höller, A survey on hierarchical planning-one abstract idea, many concrete realizations, in: IJCAI, 2019, pp. 6267–6275.
- [55] J. Roslöf, I. Harjunkoski, T. Westerlund, J. Isaksson, Solving a large-scale industrial scheduling problem using MILP combined with a heuristic procedure, *European J. Oper. Res.* 138 (1) (2002) 29–42.
- [56] C. Scholz, H.-L. Cao, E. Imrith, N. Roshandel, H. Firouzipooyaei, A. Burkiewicz, M. Amighi, S. Menet, D.W. Sisavath, A. Paolillo, et al., Sensor-enabled safety systems for human-robot collaboration: A review, *IEEE Sens. J.* (2024).
- [57] P. Palmieri, M. Melchiorre, L.S. Scimmi, S. Pastorelli, S. Mauro, Human arm motion tracking by kinect sensor using kalman filter for collaborative robotics, in: Advances in Italian Mechanism Science: Proceedings of the 3rd International Conference of IFToMM Italy 3, Springer, 2021, pp. 326–334.
- [58] F. Ferraguti, C.T. Landi, S. Costi, M. Bonfè, S. Farsoni, C. Secchi, C. Fantuzzi, Safety barrier functions and multi-camera tracking for human-robot shared environment, *Robot. Auton. Syst.* 124 (2020) 103388.
- [59] M. Ferrari, S. Sandrini, C. Tonola, E. Villagrossi, M. Beschi, Predicting human motion using the unscented Kalman filter for safe and efficient human-robot collaboration, in: 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation, ETFA, 2024, pp. 1–8.
- [60] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, M. Grundmann, BlazePose: On-device real-time body pose tracking, 2020, arXiv:2006.10204.
- [61] S. Sandrini, M. Beschi, A ROS-Based human pose tracking system based on Mediapipe and Kalman filter, URL https://github.com/SamueleSandrini/skeleton_trajectory.
- [62] P.A. Lasota, J.A. Shah, Analyzing the effects of human-aware motion planning on close-proximity human-robot collaboration, *Hum. Factors* 57 (1) (2015) 21–33.
- [63] C. Petzoldt, D. Niermann, E. Maack, M. Sontopksi, B. Vur, M. Freitag, Implementation and evaluation of dynamic task allocation for human-robot collaboration in assembly, *Appl. Sci.* 12 (24) (2022) 12645.
- [64] W.H. Kruskal, W.A. Wallis, Use of ranks in one-criterion variance analysis, *J. Amer. Statist. Assoc.* 47 (260) (1952) 583–621.