

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

Samuele Sandrini¹, Marco Faroni², Nicola Pedrocchi¹

Abstract—Modern hybrid robot cells leverage heterogeneous agents to provide agile production solutions. Effective agent coordination is crucial to avoid inefficiencies and potential hazards for human operators working among robots. This paper proposes a new human-aware task allocation and scheduling model based on Mixed Integer Non-Linear Programming (MINLP) to optimize efficiency and safety during task planning, scheduling, and allocation. The approach introduces a synergy index that encodes the coupling effects between pairs of tasks executed in parallel by the agents. These terms are learned from previous process executions by means of a Bayesian linear estimation. The task planning model is enhanced by the knowledge of synergy terms to adapt the nominal duration of the plan to consider the effect of the operator's presence. Simulations and experimental results demonstrate the effectiveness of the proposed method in obtaining a proactive human-aware solution starting from the task planning level. The proposed model reduces process execution time and achieves solutions with less agent interference, more considerable human-robot distance, and, thus, safer for agents.

Note to Practitioners—Poor coordination of humans and robots in hybrid cells may jeopardize the advantages of human-robot collaboration. This work is motivated by the problem of distributing tasks in a way that the coordination is optimized, reduces interference between robotic and human agents, and thus achieves a more efficient and safer solution. We propose a novel approach that combines learning the interference between the agents from previous task execution and leverages such knowledge to decide who to assign tasks to and when. We use a Bayesian regressor to learn the interference indices and a Mixed Integer Non-Linear Programming (MINLP) formulation to address the planning problem. The method was tested in a human-robot collaboration application scenario for E-waste disassembly, where agents have different tasks to perform in different areas, showing significant improvement in the cycle time and average human-robot distance.

Index Terms—Task Allocation; Task Scheduling; Collaborative Robotics; Task and Motion Planning; Autonomous agents; Agents Coordination; Hybrid Robotics-Cells; E-Waste Disassembly

I. 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 the ISO-TS 15066 requires that the robot speed is modulated based on the agents' distance to keep a protective separation [2].

Over the years, researchers have tackled the safety-vs-efficiency trade-off by acting on the motion planning and control levels (e.g., with impedance control [3], [4], safety-aware path planning [5], [6], and fast replanning [7]). However, a decision-making level that is unaware of safety aspects may lead to frequent interference between the agents and increase the risk of collisions and, therefore, the frequency of safety stops and re-plannings. Thus, it is crucial that the task planning, assignment, and scheduling of the tasks consider safety aspects, such as possible slowdowns between pairs of spatially close tasks executed in parallel by the agents to foster an efficient collaboration between agents.

This article proposes two novel Human-Aware Task Allocation and Scheduling models for Human-Robot Collaboration (HRC) solutions based on Mixed Integer Non-Linear Programming (MINLP). While most of the literature considers safety as an online countermeasure, this paper takes a proactive approach at the task planning level by providing a safe and efficient solution, unhinging the concept of safety versus efficiency. The key contributions of this paper can be summarized as follows:

- We propose a method for estimating synergy between pairs of human-robot tasks based on knowledge gained from past process executions. Synergy terms encode the slowdown to the average execution duration of a robot task owed to the parallelism of a task performed by the human operator [8]. We use Bayesian regression for coefficient estimation to exploit a priori knowledge of the effect of the agents' tasks on each other.
- We propose two MINLP models that use such synergies to account for the effects caused by parallel execution with operator tasks. The first model, called *Synergistic TP* (STP), introduces constraints to calculate the overlapping time between task pairs, with the end task time adjusted based on the actual human-robot coupling. This favors optimal coupling in terms of efficiency and safety during the execution. The second model, called *Relaxed Synergistic TP* (R-STP), aims to reduce the computational complexity of the first model by considering a simplified synergy term in the cost function. We extend the two methods to the human-multi-robot case. Finally, Appendix A describes a re-formulation of all these approaches with Mixed-Integer Linear Programming.

The proposed methods were tested in a simulated HRC scenario for mosaic composition and validated on a real case study of electronic waste disassembly in a collaborative robotic work cell, as shown in the accompanying video¹.

¹ STIIMA-CNR - Institute of Intelligent Industrial Technologies and Systems, National Research Council of Italy {name.surname}@stiima.cnr.it

² Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy. marco.faroni@polimi.it

¹A full version of the video is available at <https://youtu.be/xEcQgYZN59c>

II. RELATED WORKS

Task Allocation and Scheduling in HRC have been tackled using heterogeneous approaches. Several works focused only on the task allocation problem [9]. For example, [10] use AND/OR graphs plans representation [11] and solve the optimal allocation problem by graph search with A*. Building upon this work, [12] emphasized the agent's capability-based allocation problem by introducing indices about task complexity, agent dexterity, and effort. In [13], [14], the problem of online agents' task allocation is addressed using AND/OR graph representation and 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 [15], [16]. In [17], the online task allocation problem is integrated with a customized *control node* into a *Behavior Tree* [18]; the allocation is solved through MILP on a task subset at runtime only when it is time to perform that subset of actions.

Both allocation and scheduling are considered in [19]. The authors approached the allocation problem based on agent capabilities, considering a single human and single robot duo in an offline fashion. The allocation and scheduling problems are solved in series, using a variant of the Hungarian algorithm for the former and a MILP approach for the latter.

Few works incorporate the degradation of humans' performance [20] during execution [21], [22]. The work in [23] 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 re-schedules when triggered by the agents monitoring. The same architecture was used in [24] to adapt the schedule locally in case of failures.

The authors of [25], [26] solved the human-robot coordination with a *Timeline-based* task planner included in the *Task and Motion Planning (TAMP)* framework [27]. In [28], the authors combine TAMP with a human motion prediction module based on a goal-conditioned recurrent neural network to reduce human-robot interferences. *Human-Aware Task Planning* was studied by Alami et al. in [29] using the *Hierarchical Task Network* paradigm [30]. This evolved into the *HATP (Hierarchical Agent-based Task Planner)* framework [31], [32], [33] and its variants [34], which consider the human agent as a rational agent with its decision model. *Theory of Mind (ToM)* was recently explicitly introduced at Task Planning level in [35], [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] and [39] apply timeline-based planning in 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 by solving it using metaheuristic techniques. Finally,

[42] solves the allocation problem using genetic algorithms for HRC in mold assembly processes.

Only a few research works consider safety and efficiency aspects jointly at the task planning level, which is the present work's core. The authors in [43] formalized 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 indirectly consider safety effects by introducing *spatial constraints*, avoiding the parallel execution of tasks pre-defined to be in the same working area. In addition, they provide online monitoring with re-scheduling. In [44], they enrich the formulation by introducing the cost of task switches during online re-scheduling. The authors in [45] reformulate the dynamic scheduler presented in [22], [23] to online adapt the schedule in case of variations, e.g. due to safety halts. However, it is an online countermeasure, and the human-awareness needs to be managed in advance to achieve proactive coordination.

The work in [46] explicitly relates safety to efficiency by parameterizing the optimization problem with the static SSM parameters. This work is based on the guess that a static spatial point is associated with a task to evaluate the slowdown between pairs of tasks. This is only partially true in HRC because the way humans perform a given symbolic task may vary a lot from one execution to another. For this reason, we do not assume any domain-dependent knowledge of the agent coupling, yet we learn the synergy from executions. Our model is not based on the SSM nominal parameters, nor on the a-priori definition of static spatial points associated with tasks as in [46] and [43]. Instead, it is based on synergy factors learned from the process execution history and independent of the setup. Furthermore, our model is continuous as only the overlapping portion between task pairs is scaled according to their synergy.

III. METHODOLOGY

This section proposes a Task Planning problem formulation for the joint resolution of Human-Aware Allocation and Scheduling based on MINLP formalization and the Bayesian regression approach for synergy estimation. See also Appendix A for the linearization of the proposed MINLP models.

A. Problem Definition

Consider a set $\mathcal{T} = \{\tau_1, \dots, \tau_m\}$ of m tasks and a set of two agents $\mathcal{A} = \{H, R\}$, i.e., the human operator and the robot. We aim to find the task scheduling (i.e., when each task is to be executed, under precedence constraints) and the assignment of each task to an agent so as to minimize a cost function, e.g., the process completion time.

We define a task as a tuple $\tau_i = (\hat{d}_i^j, s_{i,k}^j, C_{i,j}, P_{i,k})$ that encodes the task's characteristics and the agents' ability to execute it. In particular:

- \hat{d}_i^j is the expected duration of task τ_i when performed by agent $j \in \mathcal{A}$;

TABLE I: List of Notations.

| Symbol | Description |
|--------------------------------|--|
| $t_i^s \in \mathbb{R}^+$ | Start-Time of task τ_i |
| $t_i^e \in \mathbb{R}^+$ | End-Time of task τ_i |
| $a_i^j \in \{0, 1\}$ | Binary variable for the assignment of task τ_i to agent j . |
| $C_{ij} \in \{0, 1\}$ | Binary input variable that specifies whether agent j can perform task τ_i . |
| $P_{ik} \in \{0, 1\}$ | Binary input variable that specifies whether task τ_i must precede task τ_k . |
| $\hat{d}_i^j \in \mathbb{R}^+$ | Expected duration of task τ_i executed by agent j . |
| $OV_{i,k} \in \mathbb{R}^+$ | Overlapping time between task τ_i and τ_k . |
| $s_{i,k}^j \in \mathbb{R}^+$ | Synergy Term between task τ_i (executed by the agent j) and task τ_k (executed by the human agent). |
| M | Classic “big-M” used in Linear Programming |

- $s_{i,k}^j$ is a synergy term that represents the execution slowdown of pairs of simultaneous tasks, *i.e.*:

$$s_{i,k}^j = \frac{\hat{d}_{i,k}^j}{\hat{d}_i^j} \quad (1)$$

where $\hat{d}_{i,k}^j$ is the expected duration of task τ_i performed by the agent j when the other agent ($\mathcal{A} \setminus j$) executes task τ_k and \hat{d}_i^j is the expected value of the duration of τ_i for all possible concurrent task. Thus, the synergy term takes into account the coupling effect between pairs of tasks: if a task executed by the human causes a slowdown of the robot task, then $s_{i,k}^j > 1$; if the parallelism has beneficial effects, then $s_{i,k}^j < 1$. This synergy term makes it possible to consider the slowdown caused by the safety module.

- $C_{i,j}$ encodes the capability of agent j to execute task τ_i . $C_{i,j} = 1$ if agent j can execute τ_i , 0 otherwise;
- $P_{i,k}$ specifies whether a precedence constraint between task τ_i and task τ_k exists. In particular, $P_{i,k} = 1$ if τ_i is to be executed before τ_k , 0 if no order is imposed between τ_i and τ_k .

The solution to the task planning problem for allocation and scheduling is a plan $\pi = \{\pi_i\}_{i=1}^{|\mathcal{T}|}$ such that:

$$\pi_i = \left(t_i^s, t_i^e, \{a_i^j\}_{j \in \mathcal{A}} \right) \quad \forall i = 1, \dots, |\mathcal{T}| \quad (2)$$

which is a sequence of tuples that, for each task $\tau_i \in \mathcal{T}$, specifies the start and end times (t_i^s and t_i^e), and assignment binary variables a_i^j defined as:

$$a_i^j = \begin{cases} 1 & \text{if Task } \tau_i \text{ is assigned to Agent } j \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

respecting the precedence constraints P_{ik} and the capability constraints C_{ij} .

Table I summarizes the main variables involved in the optimization problem.

B. Human-Aware Task Allocation and Scheduling (STP)

Based on the problem definition given in the previous section, a human-aware optimal task allocation and scheduling in (2) can be obtained as a solution of the following Mixed Integer Non-Linear Problem:

$$\begin{aligned} \pi^* = \arg \min_{t_i^s, t_i^e, a_i^j} \quad & \mathcal{J} \\ \text{subject to} \quad & \mathcal{C}_{\text{goal}}, \\ & \mathcal{C}_{\text{precedence}}, \\ & \mathcal{C}_{\text{capability}}, \\ & \mathcal{C}_{\text{not-overlapping}}, \\ & \mathcal{C}_{\text{performance}} \end{aligned} \quad (4)$$

where \mathcal{J} is the makespan:

$$\mathcal{J} = \max_{i=1, \dots, |\mathcal{T}|} (t_i^e) \quad (5)$$

$\mathcal{C}_{\text{goal}}$ stands for the process constraint that all tasks shall be executed once, *i.e.*:

$$\mathcal{C}_{\text{goal}} := \sum_{j \in \mathcal{A}} a_i^j = 1 \quad \forall i \in \{1, \dots, |\mathcal{T}|\}. \quad (6)$$

$\mathcal{C}_{\text{precedence}}$ enforces precedence constraints between tasks by imposing that the start time of τ_i is greater than the end time of τ_j if the corresponding value of $P_{i,k}$ is set to 1:

$$\mathcal{C}_{\text{precedence}} := t_i^s \geq t_k^e P_{i,k} \quad \forall i, k \in \{1, \dots, |\mathcal{T}|\}, k \neq i. \quad (7)$$

Our allocation model is *capability-based*, *i.e.*, it considers the agents' abilities. The allocation of tasks to agents that are unable to perform that specific task is avoided through the following constraint:

$$\mathcal{C}_{\text{capability}} := a_i^j \leq C_{i,j} \quad \forall (i, j) \in \{1, \dots, |\mathcal{T}|\} \times \mathcal{A} \quad (8)$$

$\mathcal{C}_{\text{not-overlapping}}$ avoids overlapping of tasks assigned to the same agent:

$$\mathcal{C}_{\text{not-overlapping}} :=$$

$$t_i^s > t_k^s - M(1 - \delta_{i,k}) - M(2 - a_i^j - a_k^j) \quad (9)$$

$$t_i^s \leq t_k^s + M\delta_{i,k} + M(2 - a_i^j - a_k^j) \quad (10)$$

$$t_i^s \geq t_k^e - M(1 - \delta_{i,k}) - M(2 - a_i^j - a_k^j) \quad (11)$$

$$t_k^s \geq t_i^e - M\delta_{i,k} - M(2 - a_i^j - a_k^j) \quad (12)$$

$$\forall i, k \in \{1, \dots, |\mathcal{T}|\}, k \neq i \quad \forall j \in \mathcal{A}$$

We use the big-M notation to ensure that these constraints are effective only if the same agent performs both tasks τ_i and τ_k ; indeed, if the same agent does not execute both tasks, then $2 - a_i^j - a_k^j \neq 0$ and (9)-(12) always hold. Conversely, assuming that $a_i^j = a_k^j = 1$, the behavior of the constraints is managed by the auxiliary variable $\delta_{i,k} \in \{0, 1\}$. If $\delta_{i,k} = 1$, constraints (9) and (11) are active, while (10) and (12) always hold and vice-versa if $\delta_{i,k} = 0$.

Finally, $\mathcal{C}_{\text{performance}}$ determines the expected task end times:

$$\begin{aligned} \mathcal{C}_{\text{performance}} := & t_i^e = t_i^s + \sum_{j \in \mathcal{A}} \hat{d}_i^j a_i^j \\ & + \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{T}|} \text{OV}_{i,k} (s_{i,k}^R - 1) a_i^R \quad (13) \\ & \forall i \in \{1, \dots, |\mathcal{T}|\} \end{aligned}$$

Here, t_i^s is the start time of τ_i , the second term determines the task's duration according to the selected agent, and the third term alters the nominal duration of the task based on parallelism with the various tasks. Because safety in human-robot collaboration affects the robot motion, we consider only the synergy effect when the task τ_i is executed by the robot ($a_i^R = 1$). The extension to situations where the coupling may affect also the human tasks' duration is straightforward. Note that the synergy term is scaled to the real overlapping time between the task pair, which is defined as:

$$\text{OV}_{i,k} = \begin{cases} 0, & \text{if } \Delta T_{i,k} < 0 \\ \Delta T_{i,k}, & \text{otherwise} \end{cases} \quad (14)$$

$$\forall i, k \in \{1, \dots, |\mathcal{T}|\}, k > i$$

where:

$$\Delta T_{i,k} = \min(t_i^e, t_k^e) - \max(t_i^s, t_k^s) \quad (15)$$

In (13), it is clear that the duration of a robot's task is conditioned on the concurrent human's task. Otherwise, the average duration would be misleading during planning and scheduling.

C. Relaxed Human-Aware Task Allocation and Scheduling Formulation (R-STP)

The synergy term $s_{i,k}^R$ introduced affects a set of closely coupled constraints (9)-(12)-(13), and this can make the problem computationally complex. 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:

$$\mathcal{C}_{\text{performance}} := t_i^e = t_i^s + \sum_{j \in \mathcal{A}} \hat{d}_i^j a_i^j \quad \forall i \in \{1, \dots, |\mathcal{T}|\} \quad (16)$$

and (5) with:

$$\mathcal{J} = \max_{i=1, \dots, |\mathcal{T}|} (t_i^e) + \Delta S \quad (17)$$

where:

$$\Delta S = \sum_{i=1}^{|\mathcal{T}|} \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{T}|} \text{OV}_{i,k} (s_{i,k}^R - 1) a_i^R \quad (18)$$

The rationale behind (17) 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 human and robot tasks (ΔS).

D. Extension to human multi-robot systems

In the previous sections, we referred to the case of two agents: one human operator and one robot. This section extends the formalization to the case of a hybrid cell with multiple robots and a human operator. In the human-multi-robot case, the agent set can be defined as $\mathcal{A} = A^R \cup A^H$, with $A^R = \{r_1, \dots, r_n\}$. Following the formalization of Section III-B, we replace (13) with:

$$\begin{aligned} \mathcal{C}_{\text{performance}} := & t_i^e = t_i^s + \sum_{j \in \mathcal{A}} \hat{d}_i^j a_i^j \\ & + \sum_{r \in A^R} \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{T}|} \text{OV}_{i,k} (s_{i,k}^r - 1) a_i^r a_k^H \quad (19) \\ & \forall i \in \{1, \dots, |\mathcal{T}|\} \end{aligned}$$

where $s_{i,k}^r$ is the synergy term between the human agent and the robot r of the task pair (τ_i, τ_k) .

Similarly, to extend the relaxed problem (Section III-C) to the human-multi-robot case, we replace (17) with:

$$\Delta S = \sum_{r \in A^R} \sum_{i=1}^{|\mathcal{T}|} \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{T}|} \text{OV}_{i,k} (s_{i,k}^r - 1) a_i^r a_k^H \quad (20)$$

E. Synergy Estimation

The concept of *synergy* for each pair of human-robot tasks plays a crucial role in this article and must be estimated from experience. [8] proposed a linear model that relates the measured duration of human tasks to the level of parallelism of each task the robot performs through the synergy terms defined in (1). Specifically, for a batch of n realizations of task τ_k executed by the human (H), we can write:

$$d_k^H \Big|_n - T_{\text{idle}} \Big|_n = \sum_{i=1}^m s_{i,k}^R \hat{d}_i^R \rho_{i,k}^R \Big|_n \quad (21)$$

where \hat{d}_i^R is the average duration of task τ_i performed by the robot agent, $s_{i,k}^R$ is the synergy term described in (1), $\rho_{i,k}^R$ is the percentage of task τ_i in parallel with τ_k , which is defined as the ratio between the overlapping time of τ_i and τ_k , and the actual duration of τ_i , i.e.:

$$\rho_{i,k}^R = \frac{\text{OV}_{i,k}}{d_i^R} \Big|_n \quad (22)$$

Instead of learning the synergy via linear regression as in [8], we estimate the synergy coefficients using a Bayesian approach. To do so, we assume that the measurements follow a normal distribution, i.e.:

$$d_k^H \Big|_n - T_{\text{idle}} \Big|_n \sim \mathcal{N} \left(\sum_{i=1}^m s_{i,k}^R \hat{d}_i^R \rho_{i,k}^R, \sigma_m^2 \right) \quad (23)$$

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 :

$$\sigma_m \sim \mathcal{U}(l_b, u_b) \quad (24)$$

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:

$$s_{i,k}^R \sim \text{LogN}(\mu_s, \sigma_s) \quad (25)$$

Given the above prior knowledge of the model, the posterior distribution of the estimated $S_k^R = (s_{1,k}^R, \dots, s_{|\mathcal{T}|,k}^R)$ parameters can be obtained through the application of Bayes' Theorem:

$$p(S_k^R | \mathcal{D}) = \frac{p(\mathcal{D} | S_k^R) p(S_k^R)}{p(\mathcal{D})} \quad (26)$$

where \mathcal{D} is the dataset composed of the batch of realizations of the human task τ_k with all the tasks executed in parallel by the robot. The posterior distribution is not obtained analytically but can be computed using a Markov-Chain Monte Carlo algorithm, such as No-U Turn Sampler (NUTS) [47]. This approach incorporates uncertainty in estimating synergy coefficients, making the model more robust and applicable to real-world scenarios. By employing Bayesian methods, we can leverage prior knowledge and update our beliefs based on observed data, thus obtaining more reliable estimates for the synergy effects between human and robot tasks.

IV. SIMULATIONS

This section describes the software architecture (Section IV-A), the simulated scenarios (Section IV-B), the compared methods and metrics (Section IV-C), and the results (Section IV-D).

A. Framework Architecture

The software architecture resembles that proposed in [8] and is shown in Figure 1. At the highest level, a *task planning module* solves the allocation and scheduling problem. At the lowest level, there is an independent *task execution module* for each agent. For the robot 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, it would send the execution request to an HMI and wait for feedback.

An additional interface module called *task planner interface* is added between the higher and the lower levels. This module receives task execution requests from the dispatcher, interacts with a MongoDB database [48], 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 durations and performing the synergy estimation between task pairs.

The MINLP models presented in Section III are integrated at the task planning level and implemented using the GurobiPy library [49]. In addition, the task dispatcher was implemented using the Algorithm 1. The Bayesian estimation of synergy terms proposed in Section III-E is implemented using Pyro [50] and integrated into the architecture as a ROS-Service in the statistics module.

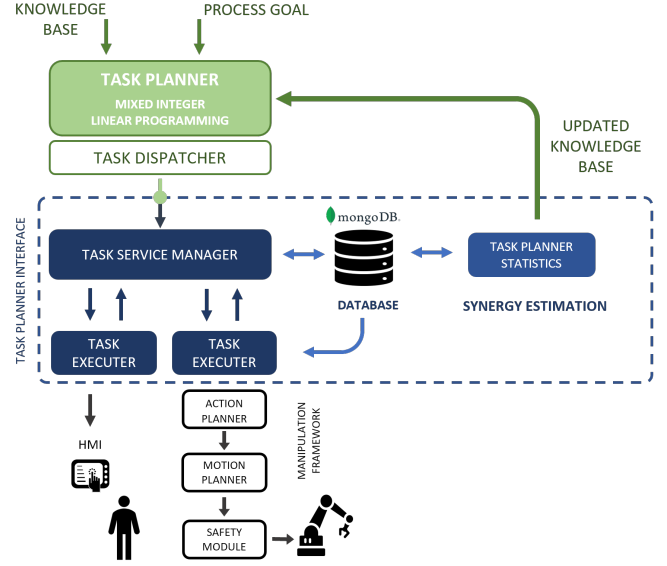


Fig. 1: Framework Architecture (adapted from [8]): a three-tiered structure in which at the highest level is the task planning and task dispatching module, at the lowest level is a manipulation framework for motion planning and trajectory execution with safety SSM module, and at the middle level is a middleware for interacting with the database for both lower and higher levels.

Algorithm 1 Task Dispatcher Algorithm

Require: $\pi = \{(t_i^s, t_i^e, a_i^j)\} \quad \forall i \in \{1, \dots, |\mathcal{T}|\}, j \in \mathcal{A}$
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 $a_k^j = 1 \quad \forall k$ -tasks
 if $\text{isEmpty}(\pi^j)$ **then**
 continue
 end if
 $\tau_i \leftarrow \pi^j.\text{getFirst}()$
 $t \leftarrow \text{getTime}()$ ▷ Get current simulation time
 if $t \geq t_i^s$ & $\text{Agent}^j.\text{isFree}()$ **then**
 Send the execution request of task τ_i
 $\pi.\text{pop}(\tau_i)$
 $\text{Agent}^j.\text{setBusy}()$
 end if
 end for
end while

B. 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. Additionally, the shared workspace contains blue cubes, which both agents can manipulate.

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

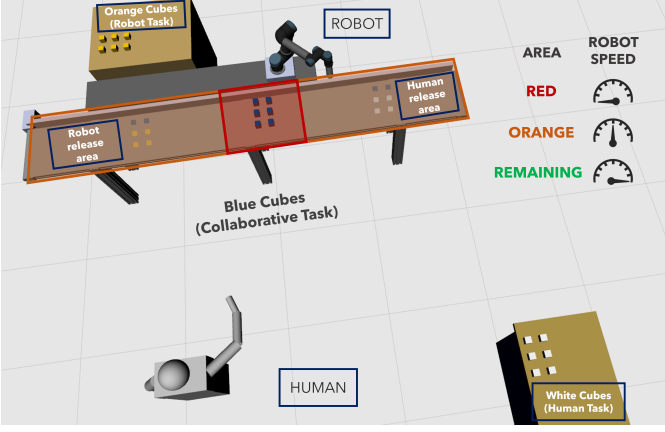


Fig. 2: Simulated setup showing the robot workspace containing the orange boxes, the human one with white boxes, and the corresponding agent's release areas in opacity. Blue boxes are shared between agents. Highlighted are the safety areas used in scenario 1.

Safety areas characterize scenario 1. This safety approach aligns with the *Speed And Separation Monitoring (SSM)* outlined in [2], defining static safety zones to maintain a minimum protective separation distance between humans and robots. In Figure 2, 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. It was chosen to test this scenario since it is widely used in HRC settings in industrial environments [51]. 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 cubes (blue ones).

Scenario 2 is based on the robot's velocity modulation to prevent it from colliding with a human, thus ensuring a minimum protective agent separation. 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]):

$$v_{hr_{max}}(t) \geq \sqrt{v_h^2 + (a_s T_r)^2 - 2a_s(C - S_d(t)) - a_s T_r - v_h} \quad (27)$$

in which the involved parameters are detailed in Table II.

TABLE II: SSM Parameters Description

| Parameter | Value | Description |
|-----------|------------------------|--------------------------------------|
| v_h | 1.6 m s^{-1} | Human Velocity toward the Robot* |
| a_s | 1 m s^{-2} | Maximum Robot Cartesian Deceleration |
| T_r | 0.3 s | Maximum System Reaction Time** |
| C | 0.2 m | Position Uncertainty |
| S_d | Measured | Separation Distance |

* Specified by ISO15066 in cases where human velocity is not measured.

** Robot and Vision System.

The task-level goal in this scenario 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).

We chose to simulate such safety rules because they represent an ideal implementation of the current safety specifications and must always be ensured, even in the most advanced cases of online replanning [7].

The simulations were conducted in two phases on a PC with a 2.8 GHz and 4-core CPU. The initial phase involved the offline execution of 50 random plans to estimate the average durations of individual tasks and the synergies. Then, task planning was performed based on the estimates obtained in the first phase. Regarding the parameters related to the priors of the distributions mentioned in Section III-E for synergy estimation, we set $l_b = 0 \text{ s}$, $u_b = 2 \text{ s}$, and $\mu_s = 0$ to ensure the median value of 1 of the synergy terms and $\sigma_s = 0.5$.

C. Baseline and Metrics

We compare our methods with two baselines: (i) *Baseline TP*, which is a simplified version of (4) without taking into account the coupling effect between tasks-pairs, similarly to [8]; (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:

$$\text{Makespan} = \max_{i \in \{1, \dots, |\mathcal{T}|\}} (\tilde{t}_i^e) \quad (28)$$

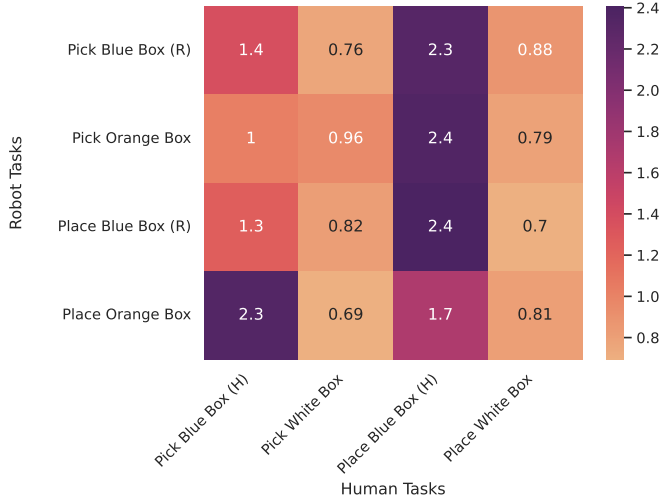
where \tilde{t}_i^e is the measured end time of a task τ_i in a task set \mathcal{T} to perform during a plan solution π . As for safety, we measure the minimum distance between the human operator and the robot during the execution of the plans. Then, we compute the cumulative probability distribution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

$$\frac{1}{N_{\text{plans}}} \sum_{k=1}^{N_{\text{plans}}} \mathbb{P}(D_{\min} \leq \tilde{d}) \quad \forall \tilde{d} \in [0, d_{\max}] \quad (29)$$

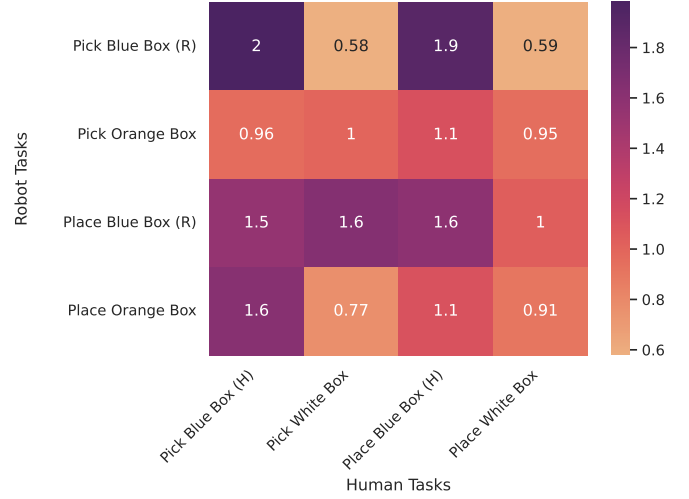
Thus, this index represents the probability that D_{\min} falls below a given distance \tilde{d} for the method at hand.

D. Results

First, we evaluate the synergy coefficients estimated for both scenarios. The heatmap in Figure 3 shows the synergy for each robot task (matrix rows) for each concurrent human task (matrix columns). A synergy index 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 (Figure 3a), the most significant slowdown occurs when the human operator places the shared boxes (blue boxes) in their designated release area. The start point of the place blue box task causes the robot stop at the execution start: this highlights the challenge of selecting a static point associated with a task,



(a) Scenario 1: Static safety zones



(b) Scenario 2: Continuous safety velocity scaling

Fig. 3: Robot Estimated Synergy terms in matrix format

as both the goal and the starting spatial point of task execution have an impact on the synergy.

Coupling effects between tasks also arise in Scenario 2, where the scaling factor acts more smoothly, changing continuously based on (27). Synergy estimation results of this scenario are reported in Figure 3b. The first row of the heatmap 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 cause faster execution than the average duration. The robot's task of picking the orange object (second row of the heatmap) is generally not affected by any human tasks; vice versa, the robot task of placing the blue box is significantly slowed down, except when the human performs the white box placement. The robot placing the orange object (last row) is slowed down more by the blue box-picking task of the human (first column) rather than the box-placing task (third column) because, in the first case, the relative speed of the robot toward the human is positive and thus aggravates the velocity scaling term.

Table III compares the makespan (28) and the synergy of executed plans. Figures 4 and 5 show the results obtained from the online simulations of both Scenarios 1 and 2. In particular, Figures 4a and 5a compare the total execution time (makespan) over 50 solutions for each planner. In both scenarios, STP outperforms the other methods by showing a percentage reduction to the average execution duration of the entire plan (about 18% with respect to *Baseline TP* and about 13% with respect to *Not Neighboring TP*). Intuitively, the performance of this method is related to the fact that not only it finds the best coupling between robot and human tasks but also adjusts the task time to the actual execution time due to the overlapping between parallel tasks and the synergy between them.

The Relaxed version of STP (R-STP) method achieves solutions with an average plan execution time comparable

with the best case of the Baseline and the Not Neighboring TP; the average makespan obtained with Relaxed-STP method overcomes the baseline (with a reduction of about 13% for both scenarios) and the solutions obtained from Not Neighboring TP (with a reduction about 7% for both Scenarios).

Figures 4b and 5b show the average cumulative probability distribution of the minimum human-robot distance (29), measured during plan executions. In Scenario 1, both the STP and Relaxed-STP are characterized by a cumulative probability distribution comparable with the Not Neighboring TP method for small distances (≈ 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 even more evident for larger distances. Relaxed-STP is the least likely to have distances below 1.8 m, 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.

Considerations on the Computational Complexity: Although this paper is not focused on providing real-time solutions, we discuss the proposed model's computational complexity. The proposed method is formalized using MINLP, which is generally \mathcal{NP} -hard; the curse of dimensionality of integer variables and the handling of nonlinearities can lead to suboptimal solutions with limited computation. For real-world problems such as those presented in the article, several parameters can affect the quality of the solution. Two parameters are manually tuned to find a good trade-off between computational speed and solution quality. 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 equals 0 when an optimal solution is found. The second parameter is

TABLE III: Comparison of Task Planners: Performance Summary of Optimized Plans

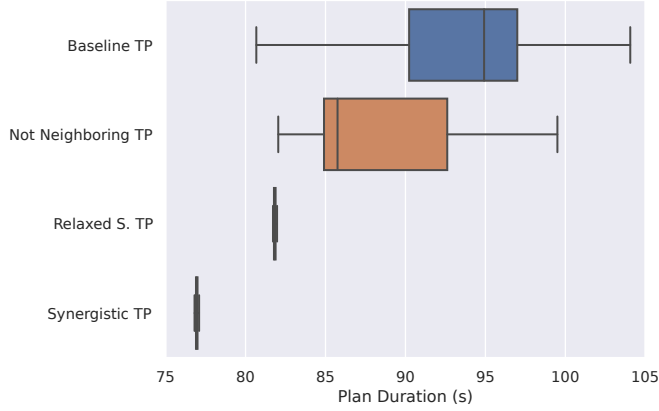
| Method | Makespan (s) | Average ΔS (s)* | Min ΔS (s)* | Max ΔS (s)* |
|--------------------|--------------|-------------------------|---------------------|---------------------|
| Baseline TP | 81.80 | 8.90 ± 7.80 (95%) | 0.16 | 14.96 |
| Not Neighboring TP | 81.80 | 1.93 ± 6.00 (95%) | -4.00 | 9.66 |
| Relaxed-STP | 81.80 | -4.65431 | -4.65 | -4.65 |
| Synergistic TP | 79.61 | -3.94048 | -3.94 | -3.94 |

(a) Scenario 1: Safety Areas.

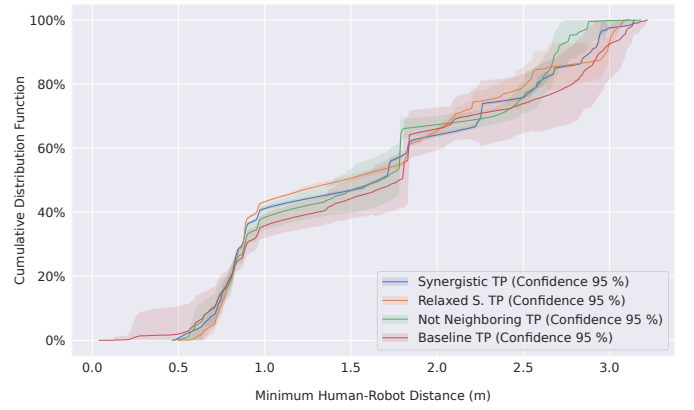
| Method | Makespan (s) | Average ΔS (s)* | Min ΔS (s)* | Max ΔS (s)* |
|--------------------|--------------|-------------------------|---------------------|---------------------|
| Baseline TP | 68.34 | 5.05 ± 13.67 (95%) | -6.63 | 16.37 |
| Not Neighboring TP | 68.34 | -4.01 ± 3.75 (95%) | -7.20 | -1.56 |
| Relaxed-STP | 68.34 | -8.76 | -8.76 | -8.76 |
| Synergistic TP | 62.11 | -6.29 | -6.29 | -6.29 |

(b) Scenario 2: Safety with continuous speed scaling.

*The term ΔS is computed according to the definition in (1);
The average, minimum and maximum values are calculated over all N_{plans} .

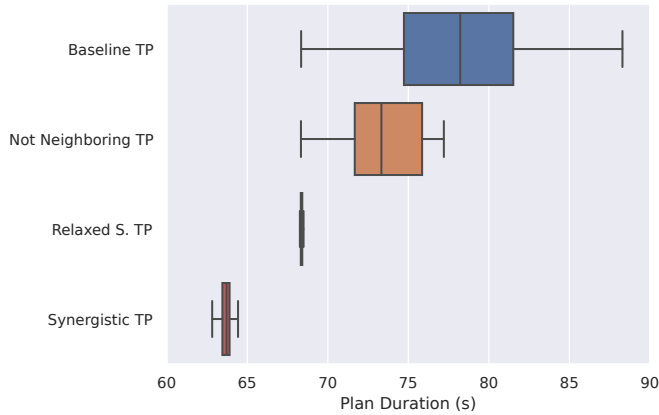


(a) Plan execution duration

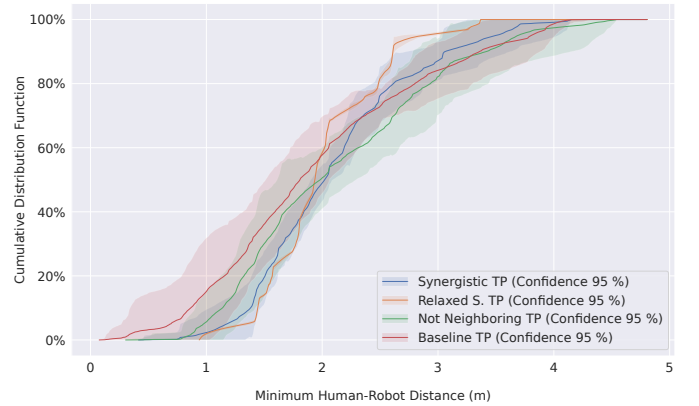


(b) Cumulative probability function of the minimum H-R distance

Fig. 4: Results of *Scenario 1*



(a) Plan execution duration



(b) Cumulative probability function of the minimum H-R distance

Fig. 5: Results of *Scenario 2*

TABLE IV: Optimization Parameters of Proposed Methods

| Proposed Method | MIP-GAP | Solution Time | Scenario |
|------------------|---------|---------------|----------|
| Relaxed-STP* | 1.70 % | 18 s | 1 |
| Synergistic TP** | 8 % | 192 s | |
| Relaxed-STP* | 2 % | 42 s | 2 |
| Synergistic TP** | 10.8 % | 240 s | |

* Timeout Time set to 60 s

** Timeout Time set to 240 s

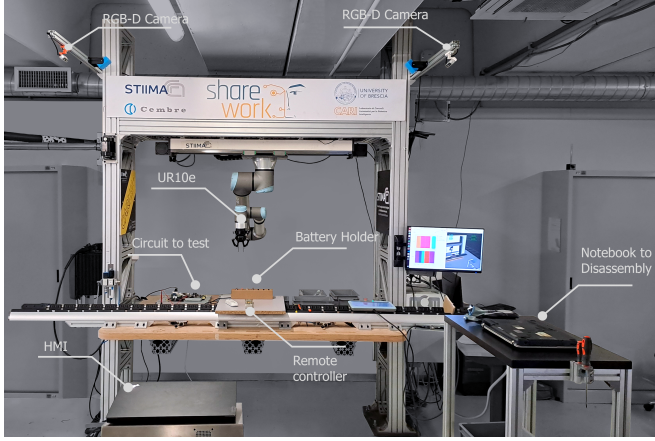


Fig. 6: Experimental setup.

the maximum available time, which typically depends on the available computation. Table IV shows the parameters used for the experiments and the solution time.

V. CASE STUDY

We consider an E-waste disassembly and testing case study. We use the collaborative cell in Figure 6, which consists of a 6-degree-of-freedom manipulator, model Universal Robots UR10e, mounted upside down. The cell was designed within the EU 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.

The human pose estimation system is based on Mediapipe BlazePose [52] with an additional Kalman Filter (code available [53]) to enhance the reconstruction’s stability.

The experiments are also summarized in the attached video.

A. E-Waste Disassembly and Testing Scenario

The tasks assigned to the agents in the applied scenario are illustrated in Figure 7. The robot is tasked with testing an electrical circuit, which involves locating the board for inspection, probing the circuit with a multimeter probe, and disconnecting the power socket. Additionally, the robot shall place two batteries from the battery holder into an appropriate box (for recovery or disposal). The only precedence constraint for the robot is the localization task, which must be completed before any tasks involving the board. Concurrently, a human is responsible for disassembling a laptop: first, removing the covers and, subsequently, extracting the components. Moreover,

the human operator has to disassemble two remote controls and store the batteries in the battery holder, from which the robot retrieves them (not necessarily in a specific order, as it serves as a battery buffer). The remote controller and batteries are placed in a workspace closely shared with the robot.

The process was executed 30 times with random plans (generated by Baseline TP) to estimate synergy coefficients between pairs of tasks. The results of the offline synergy estimation are shown in Figure 8.

B. Experiments Protocol and Results

The experiments involved 15 people, including university students, PhD students, researchers, and other workers of age between 19 and 31. The experimental procedure was organized as follows. After initial training on the tasks to execute, the participants performed five plans for each method in a random order. During the execution of the plans, we recorded task durations 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 using the proposed method. This makes a comparative analysis of the different methods on the operators’ perceived feelings during collaboration possible. The questionnaire items are presented in Table V and pertain to overall satisfaction (Question 1), the perceived assessment of fluency (Question 5), safety (Question 2), robot interruptions (Question 6), idle time (Question 4), and human-robot distance (Question 3). The questionnaire follows a five-option response format (from “strongly disagree” to “strongly agree”).

Figure 9 compares execution times. Both the proposed methods lead to shorter execution times. STP (154.6 s on average) achieved a reduction of 14.7 % compared to the Baseline TP (181.2 s on average), and 10.9 % compared to the Not Neighboring TP (173.5 s on average). Similarly, Relaxed-STP (161.5 s) achieved a reduction of 10.9 % compared to the Baseline TP and 7.0 % compared to the Not Neighboring TP.

Figure 10 compares the human-robot distance measured during the executions; for each h-r distance, it shows the percentage of samples below that distance measured during the runs. Results show that the proposed methods are less likely to incur in 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, while Not Neighboring TP has 6.75 % and Baseline TP has 15.95 %. It is worth noting that the percentage of samples obtained below this distance with the proposed methods is comparable with that of false positives detected by Mediapipe.

Figure 11 displays user responses for each question, presenting response counts for each method in a stacked manner for each option. 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 [54] was performed for each item to assess whether statistically significant differences exist among the response populations corresponding to the four methods. The results of the test, presented in terms of p -values, are summarized in the last

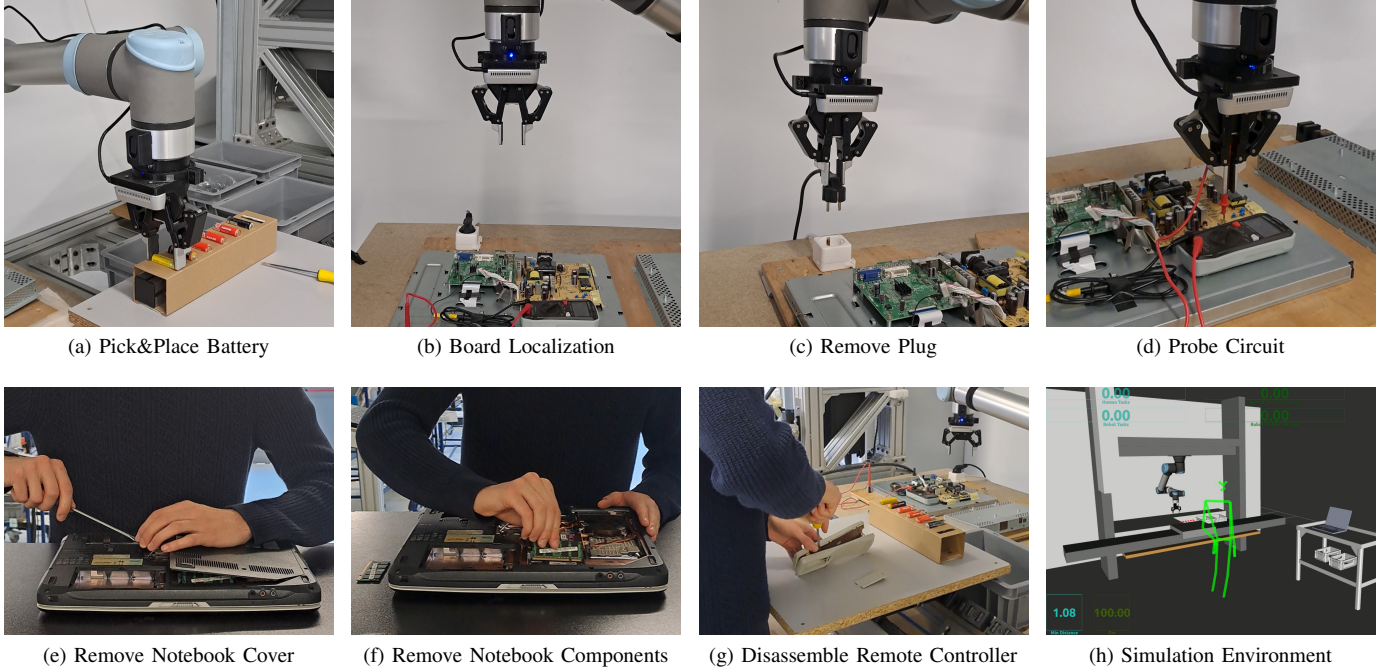


Fig. 7: Description of the robot (on top) and human tasks (on bottom), with the simulation environment

two columns of Table V. The four task planning methods are statistically equivalent for Question 2 (perceived safety) and Question 4 (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 (Q. 1), perceived human-robot proximity (Q. 3), fluency (Q. 5), and robot interruptions (Q. 6). 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. Analyzing the distributions in Figure 10 makes it evident that they favor the proposed methods: Q. 1 (overall satisfaction) has a median value of “Agree” for the proposed methods and “Neutral” for the baselines; Q. 3 (perceived human-robot proximity) has a median value of “Strongly Disagree” with STP, “Disagree” with Relaxed-STP, and “Neutral” for both baselines; as for Q. 5 (fluency) both proposed methods have a median value of “Strongly Agree”, while Baseline TP scored “Neutral” and Not Neighbouring TP scored “Agree”; finally, Q. 6 (robot interruptions) has a median value of “Strongly Disagree” for STP, “Disagree” for Relaxed-STP, “Neutral” for Not Neighbouring TP, and “Agree” for Baseline TP.

VI. CONCLUSIONS

The paper proposes a novel human-aware task allocation and scheduling model based on learned human-robot synergy.

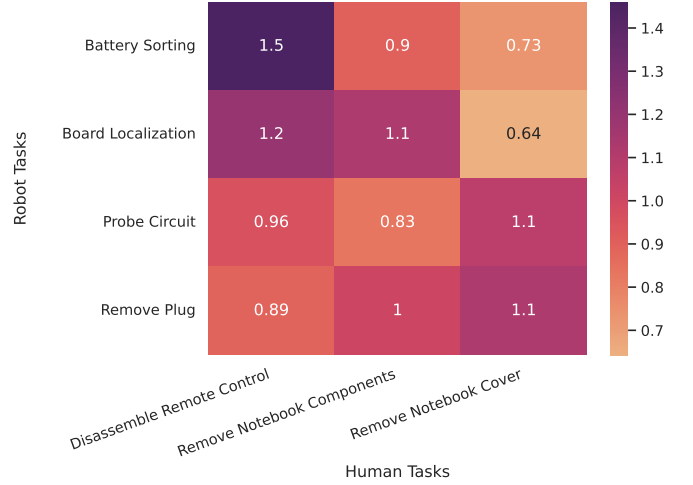


Fig. 8: Estimated Synergy Matrix of the real-case scenario.

Synergy learning enables the evaluation of synergy collaboration in concurrent task execution by the two agents. This allows for both more efficient and safe solutions as early as the task planning stage. Both the simulations and the case study experiments confirmed the effectiveness of the proposed models. Future works will focus on dynamic plan adaptation to human behavior at the task-planning level, exploring the potential of a sim2real approach for synergy estimation.

APPENDIX

This section presents a linearization of the MINLP models introduced in Section III.

The nonlinearity of the model in Section III-B is primarily related to the cost function, which can be linearized by

TABLE V: Statistical results of Kruskal-Wallis Test on user's questionnaire responses.

In green are p -values < 0.05 (statistically significant difference detected), in red are statistically equivalent (p -values > 0.05).

| Questions | Method C* VS Method A | Method C VS Method B | Method D VS Method A | Method D VS Method B | Statistical difference between methods | Kruskal-Wallis Test (p -value) |
|---|-----------------------------|----------------------------|----------------------------|----------------------------|---|--------------------------------------|
| Question 1: 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 |
| Question 2: I felt safe when working with the robot. | 7.0e-01 | 9.1e-02 | 2.6e-01 | 3.8e-02 | False | 1.1e-01 |
| Question 3: I felt that 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 |
| Question 4: Idle times bother me. | 2.3e-01 | 1.3e-01 | 4.0e-01 | 2.1e-01 | False | 3.5e-01 |
| Question 5: 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 |
| Question 6: 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 |

* Method A is the Baseline TP; Method B the Not Neighboring TP; Method C the Relaxed-STP; Method D the Synergistic TP.

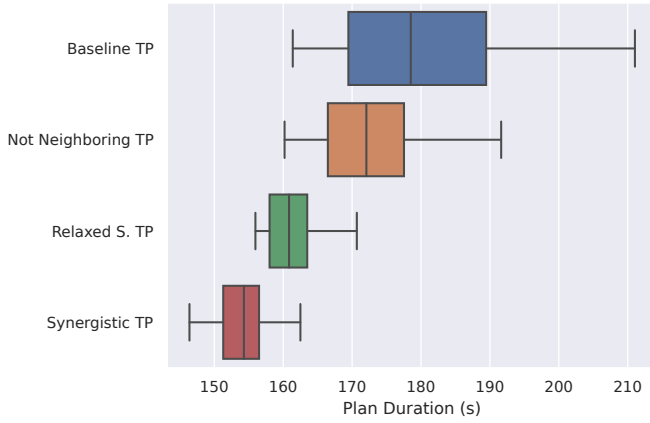


Fig. 9: Plan execution duration of the real-case scenario.

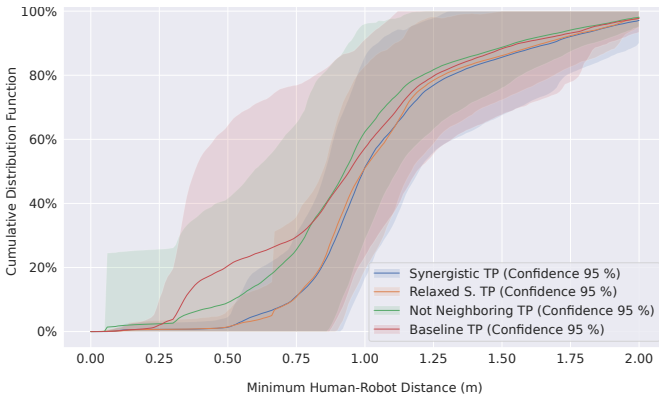


Fig. 10: Human-Robot distance in the real-case scenario.

introducing an auxiliary variable $t^{e-\max}$, i.e., the makespan, with the following linear constraints:

$$C_{\text{makespan}} := t^{e-\max} \geq t_i^e \quad \forall i \in \{1, \dots, |\mathcal{T}|\} \quad (30)$$

Thus, the non-linear cost function in (5) is replaced by:

$$\mathcal{J} = t^{e-\max} \quad (31)$$

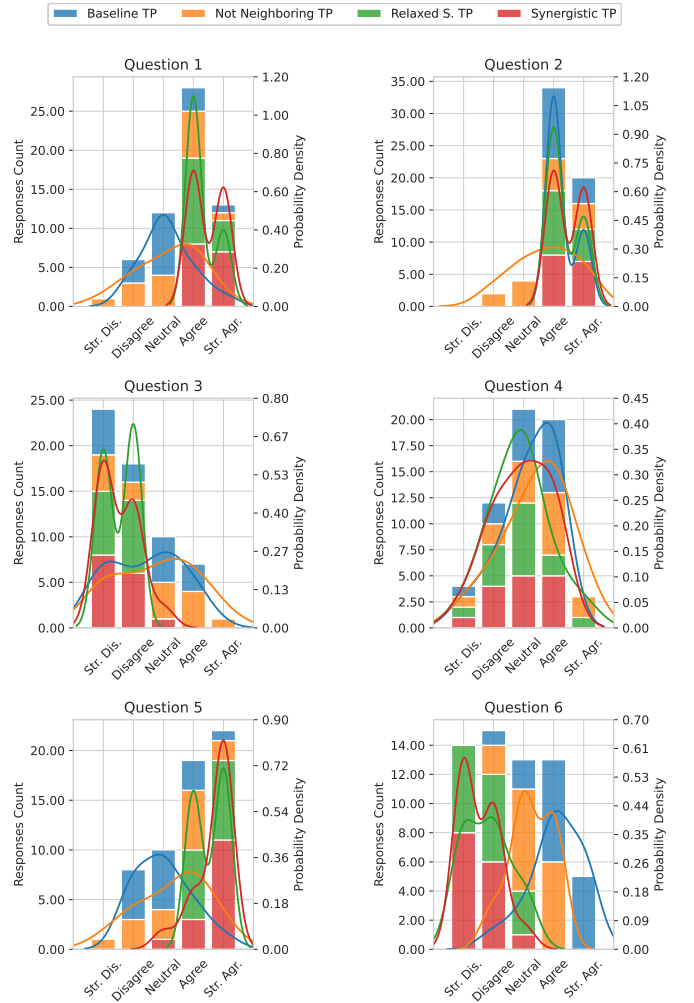


Fig. 11: Questionnaire Responses for each question. The x-axis displays user responses ranging from “strongly disagree” to “strongly agree”. Each response option's count is stacked for each method, with solid lines representing the estimated continuous probability densities.

The overlapping definition in (14) 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) and 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) :

$$C_{\max-\text{start}} := \begin{aligned} t_{i,k}^{s-\max} &\geq t_i^s \end{aligned} \quad (32)$$

$$t_{i,k}^{s-\max} \geq t_k^s \quad (33)$$

$$t_{i,k}^{s-\max} \leq t_i^s + M(1 - \sigma_{i,k}^I) \quad (34)$$

$$t_{i,k}^{s-\max} \leq t_k^s + M(\sigma_{i,k}^I) \quad (35)$$

$$C_{\min-\text{end}} := \begin{aligned} t_{i,k}^{e-\min} &\leq t_i^e \end{aligned} \quad (36)$$

$$t_{i,k}^{e-\min} \leq t_k^e \quad (37)$$

$$t_{i,k}^{e-\min} \geq t_i^e - M(1 - \sigma_{i,k}^{II}) \quad (38)$$

$$t_{i,k}^{e-\min} \geq t_k^e - M(\sigma_{i,k}^{II}) \quad (39)$$

$$C_{\text{overlapping}} := \begin{aligned} \text{OV}_{i,k} &\leq (t_{i,k}^{e-\min} - t_{i,k}^{s-\max}) + M\sigma_{i,k}^{III} \end{aligned} \quad (40)$$

$$\text{OV}_{i,k} \leq M(1 - \sigma_{i,k}^{III}) \quad (41)$$

$$\text{OV}_{i,k} \geq -M(1 - \sigma_{i,k}^{III}) \quad (42)$$

$$\text{OV}_{i,k} \geq 0 \quad (43)$$

$$\forall i, k \in \{1, \dots, |\mathcal{T}|\}, k \neq i$$

Thus, (14)-(15) are replaced by (40)-(43). To linearize the bilinear term in (13) that defines the actual end time of a task, it can be rewritten as follows:

$$t_i^e = t_i^s + \sum_{j \in \mathcal{A}} \hat{d}_i a_i^j + \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{T}|} \alpha_{i,k} (s_{i,k} - 1) \quad \forall i \in \{1, \dots, |\mathcal{T}|\} \quad (44)$$

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

$$C_{\text{adaptation-term}} := \begin{aligned} \alpha_{i,k} &\leq \text{OV}_{i,k} + M(1 - a_i^j) \end{aligned} \quad (45)$$

$$\alpha_{i,k} \geq \text{OV}_{i,k} - M(1 - a_i^j) \quad (46)$$

$$\alpha_{i,k} \leq M a_i^j \quad (47)$$

$$\alpha_{i,k} \geq -M a_i^j \quad (48)$$

$$\forall i, k \in \{1, \dots, |\mathcal{T}|\}, k \neq i, j = R$$

To linearize the model in section III-C, the same approach can be followed as for the model III-B. Thus, by adding (30) to the model, it is possible to rewrite the non-linear cost function in (17) with:

$$\mathcal{J} = t^{e-\max} + \Delta S \quad (49)$$

Moreover, by adding the constraints in (32)-(43) and is possible to replace the non-linear definition of ΔS in (18) with:

$$\Delta S = \sum_{i=1}^{|\mathcal{T}|} \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{T}|} \alpha_{i,k} (s_{i,k} - 1) \quad (50)$$

A similar approach can be followed for the linearization of the Human Multi-Robot model in section III-D, so by adding (32)-(43) and (45)-(48). In this case, (45) and (46) have to apply the overlapping effect only if the (i, k) task couple is executed by a human and a robot, respectively, so they must be rewritten as:

$$\alpha_{i,k} \leq \text{OV}_{i,k} + M \left(2 - a_i^H - \sum_{j \in \mathcal{A}^R} a_k^j \right) \quad (51)$$

$$\alpha_{i,k} \geq \text{OV}_{i,k} - M \left(2 - a_i^H - \sum_{j \in \mathcal{A}^R} a_k^j \right) \quad (52)$$

Thus, (19) can be written linearly for all i -task as:

$$t_i^e = t_i^s + \sum_{j \in \mathcal{A}} \hat{d}_i a_i^j + \sum_{r \in \mathcal{A}^R} \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{T}|} \alpha_{i,k} (s_{i,k} - 1) \quad (53)$$

and (20) with:

$$\Delta S = \sum_{r \in \mathcal{A}^R} \sum_{i=1}^{|\mathcal{T}|} \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{T}|} \alpha_{i,k} (s_{i,k} - 1) \quad (54)$$

REFERENCES

- [1] R. Awad, M. Fechter, and J. van Heerden, "Integrated risk assessment and safety consideration during design of hrc workplaces," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–10.
- [2] T. C. I. 299, "ISO/TS 15066-2016 Robots and robotic devices: Collaborative robots," International Organization for Standardization, Geneva, CH, Standard, 2016.
- [3] P. Song, Y. Yu, and X. Zhang, "A tutorial survey and comparison of impedance control on robotic manipulation," *Robotica*, vol. 37, no. 5, pp. 801–836, 2019.
- [4] P. Franceschi, N. Pedrocchi, and M. Beschi, "Adaptive impedance controller for human-robot arbitration based on cooperative differential game theory," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7881–7887.
- [5] M. Faroni, M. Beschi, and N. Pedrocchi, "Safety-aware time-optimal motion planning with uncertain human state estimation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 219–12 226, 2022.
- [6] R. Laha, W. Wu, R. Sun, N. Mansfeld, L. F. Figueredo, and S. Haddadin, "S*: On safe and time efficient robot motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 758–12 764.
- [7] C. Tonola, M. Faroni, M. Beschi, and N. Pedrocchi, "Anytime informed multi-path replanning strategy for complex environments," *IEEE Access*, 2023.
- [8] S. Sandrini, M. Faroni, and N. Pedrocchi, "Learning action duration and synergy in task planning for human-robot collaboration," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–6.
- [9] M. H. Christoph Petzoldt and M. Freitag, "Review of task allocation for human-robot collaboration in assembly," *International Journal of Computer Integrated Manufacturing*, vol. 36, no. 11, pp. 1675–1715, 2023.
- [10] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, 2016.
- [11] L. Homem de Mello and A. Sanderson, "And/or graph representation of assembly plans," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 188–199, 1990.
- [12] E. Lamon, A. De Franco, L. Peternel, and A. Ajoudani, "A capability-aware role allocation approach to industrial assembly tasks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3378–3385, 2019.

- [13] E. Merlo, E. Lamon, F. Fusaro, M. Lorenzini, A. Carfi, F. Mastrogiovanni, and A. Ajoudani, "Dynamic human-robot role allocation based on human ergonomics risk prediction and robot actions adaptation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2825–2831.
- [14] E. Merlo, E. Lamon, F. Fusaro, M. Lorenzini, A. Carfi, F. Mastrogiovanni, and A. Ajoudani, "An ergonomic role allocation framework for dynamic human-robot collaborative tasks," *Journal of Manufacturing Systems*, vol. 67, pp. 111–121, 2023.
- [15] I. E. Makrini, K. Merckaert, J. D. Winter, D. Lefeber, and B. Vanderborght, "Task allocation for improved ergonomics in human-robot collaborative assembly," *Interaction Studies*, vol. 20, no. 1, pp. 102–133, 2019.
- [16] B. Busch, M. Toussaint, and M. Lopes, "Planning ergonomic sequences of actions in human-robot interaction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1916–1923.
- [17] F. Fusaro, E. Lamon, E. De Momi, and A. Ajoudani, "An integrated dynamic method for allocating roles and planning tasks for mixed human-robot teams," in *IEEE 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 2021, pp. 534–539.
- [18] M. Iovino, E. Scukins, J. Styrd, P. Ögren, and C. Smith, "A survey of behavior trees in robotics and ai," *Robotics and Autonomous Systems*, vol. 154, p. 104096, 2022.
- [19] A. Monguzzi, M. Badawi, A. M. Zanchettin, and P. Rocco, "A mixed capability-based and optimization methodology for human-robot task allocation and scheduling," in *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2022, pp. 1271–1276.
- [20] C. T. Landi, V. Villani, F. Ferraguti, L. Sabattini, C. Secchi, and C. Fantuzzi, "Relieving operators' workload: Towards affective robotics in industrial scenarios," *Mechatronics*, vol. 54, pp. 144–154, 2018.
- [21] S. Zhang, Y. Chen, J. Zhang, and Y. Jia, "Real-time adaptive assembly scheduling in human-multi-robot collaboration according to human capability," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3860–3866.
- [22] A. Pupa, C. T. Landi, M. Bertolani, and 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.
- [23] A. Pupa, W. Van Dijk, and C. Secchi, "A human-centered dynamic scheduling architecture for collaborative application," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4736–4743, 2021.
- [24] A. Pupa, W. Van Dijk, C. Brekelmans, and C. Secchi, "A resilient and effective task scheduling approach for industrial human-robot collaboration," *Sensors*, vol. 22, no. 13, p. 4901, 2022.
- [25] M. Faroni, M. Beschi, S. Ghidini, N. Pedrocchi, A. Umbrico, A. Orlandini, and A. Cesta, "A layered control approach to human-aware task and motion planning for human-robot collaboration," in *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, pp. 1204–1210.
- [26] M. Faroni, A. Umbrico, M. Beschi, A. Orlandini, A. Cesta, and N. Pedrocchi, "Optimal task and motion planning and execution for multiagent systems in dynamic environments," *IEEE Transactions on Cybernetics*, pp. 1–12, 2023.
- [27] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.
- [28] A. T. Le, P. Kratzer, S. Hagenmayer, M. Toussaint, and J. Mainprice, "Hierarchical human-motion prediction and logic-geometric programming for minimal interference human-robot tasks," in *IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 2021, pp. 7–14.
- [29] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and 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.
- [30] I. Georgievski and M. Aiello, "Htn planning: Overview, comparison, and beyond," *Artificial Intelligence*, vol. 222, pp. 124–156, 2015.
- [31] R. Lallement, L. De Silva, and R. Alami, "Hatp: An htn planner for robotics," *arXiv:1405.5345*, 2014.
- [32] R. Lallement, L. de Silva, and R. Alami, "Hatp: hierarchical agent-based task planner," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2018.
- [33] L. De Silva, R. Lallement, and R. Alami, "The hatp hierarchical planner: Formalisation and an initial study of its usability and practicality," in *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2015, pp. 6465–6472.
- [34] G. Buisan and R. Alami, "A human-aware task planner explicitly reasoning about human and robot decision, action and reaction," in *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 544–548.
- [35] A. Favier, S. Shekhar, and R. Alami, "Robust planning for human-robot joint tasks with explicit reasoning on human mental state," *arXiv:2210.08879*, 2022.
- [36] —, "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.
- [37] A. Ham and M.-J. Park, "Human-robot task allocation and scheduling: Boeing 777 case study," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1256–1263, 2021.
- [38] 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," *Applied Sciences*, vol. 12, no. 14, p. 6839, 2022.
- [39] A. Umbrico, M. Anasagasti, S.-O. Bezrucav, F. Canale, A. Cesta, B. Corves, N. Mandischer, M. Mondragon, C. N. Rappis, and A. Orlandini, "Enhanced cognition for adaptive human-robot collaboration," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–7.
- [40] Y. Liu, Z. Jiang, and C. Ke, "A task allocation method in human-robot collaboration (hrc) for the disassembly of automotive traction batteries," in *IEEE International Conference on Cyber-Physical Social Intelligence (ICCSI)*, 2022, pp. 332–337.
- [41] K. Bogner, U. Pferschy, R. Unterberger, and H. Zeiner, "Optimised scheduling in human-robot collaboration—a use case in the assembly of printed circuit boards," *International Journal of Production Research*, vol. 56, no. 16, pp. 5522–5540, 2018.
- [42] Y. Y. Liao and K. Ryu, "Task allocation in human-robot collaboration (hrc) based on task characteristics and agent capability for mold assembly," *Procedia manufacturing*, vol. 51, pp. 179–186, 2020.
- [43] M. Lippi and A. Marino, "A mixed-integer linear programming formulation for human multi-robot task allocation," in *IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 2021, pp. 1017–1023.
- [44] M. Lippi, P. Di Lillo, and A. Marino, "A task allocation framework for human multi-robot collaborative settings," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7614–7620.
- [45] A. Pupa and C. Secchi, "A safety-aware architecture for task scheduling and execution for human-robot collaboration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1895–1902.
- [46] M. Faccio, I. Granata, and R. Minto, "Task allocation model for human-robot collaboration with variable cobot speed," *Journal of Intelligent Manufacturing*, pp. 1–14, 2023.
- [47] M. D. Hoffman, A. Gelman, *et al.*, "The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, 2014.
- [48] C. Györfi, R. Györfi, G. Pecherle, and A. Olah, "A comparative study: MongoDB vs. mysql," in *International Conference on Engineering of Modern Electric Systems (EMES)*, 2015, pp. 1–6.
- [49] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: <https://www.gurobi.com>
- [50] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *J. Mach. Learn. Res.*, vol. 20, pp. 28:1–28:6, 2019.
- [51] P. Karagiannis, N. Kousi, G. Michalos, K. Dimoulas, K. Mparis, D. Dimosthenopoulos, Önder Tokçalar, T. Guasch, G. P. Gerio, and S. Makris, "Adaptive speed and separation monitoring based on switching of safety zones for effective human robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 77, p. 102361, 2022.
- [52] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," 2020.
- [53] S. Sandrini and M. Beschi, "A ros-based human pose tracking system based on mediapipe and kalman filter." [Online]. Available: https://github.com/SamueleSandrini/skeleton_trajectory
- [54] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.