A Hierarchical Finite-State Machine-Based Task Allocation Framework for Human-Robot Collaborative Assembly Tasks

Ilias El Makrini[1,2], Mohsen Omidi[1,4], Fabio Fusaro[3,5], Edoardo Lamon[3],
Arash Ajoudani[3], and Bram Vanderborght[1,4]

Abstract— **Work-related musculoskeletal disorders (MSD) are one of the major cause of injuries and absenteeism at work. These lead to important cost in the manufacturing industry. Human-robot collaboration can help decreasing this issue by appropriately distributing the tasks and decreasing the workload of the factory worker. This paper proposes a novel generic task allocation approach based on hierarchical finite-state machines for human-robot assembly tasks. The developed framework decomposes first the main task into sub-tasks modelled as state machines. Based on capabilities considerations, workload, and performance estimations, the task allocator assigns the sub-task to human or robot agent. The algorithm was validated on the assembly of a crusher unit of a smoothie machine using the collaborative Franka Emika Panda robot and showed promising results in terms of productivity thanks to task parallelization, with improvement of more than 30% of the total assembly time with respect to a collaborative scenario, where the agents perform the tasks sequentially.**

I. INTRODUCTION

Human-robot collaboration (HRC) is part of industry 5.0 where a central role is given the human component [1]. It opens to new scenarios in terms of teamwork, where the efforts to achieve a common goal can be shared among the agents of the team. In the literature, the problem of assigning tasks, given a single shared job, in a mixed team composed of a arbitrary number of heterogeneous agents, such as human workers and robotic co-workers is defined as task allocation. Tasks can be assigned to humans or robot using different allocation schemes and according to distinct factors, such as capabilities, workload, execution time, performance, etc., that should capture the difference between the agents and evaluate the suitability of each agent to the task. A task allocation framework is usually made of three main components. First, a task partitioner, in charge of modeling the high-level job in smaller tasks and sub-tasks, until an atomic action that composes the job are reached. Second, a module is required to compute the suitability of the agent to the task. Finally, the proper task allocator assigns a task to each agent in a team according to the above-defined task/agent suitability.

Several task allocation schemes have been proposed to exploit the benefits of the human-robot teaming [2]–[4]. [5] developed a hierarchical framework for task allocation that assigns a full sequence of atomic tasks based on the capabilities of each agent. Similarly, in [6], complex tasks of a hybrid assembly cell are split into elementary sub-tasks that are allocated to the team members depending on their skills. The decision-making algorithms are often based on a multi-criteria approach using a cost function, such as in [7], [8]. Other methods incorporate the human and robot's models using Markov decision processes into the task allocation framework to predict human behaviour and determine the best execution plan [9], [10]. In [11], Bayesian inference is used to predict the human next goal and re-plan accordingly the robot's action in real time. Other allocation algorithms have been proposed to extend the task assignment scheme to the multi-human multi-robot context [12].

To evaluate the agent/task suitability, *Michalos et al.* [13] developed an algorithm that plans the tasks by evaluating each planning scenario to a set of parameters (ergonomics, quality and cell layout). In [7], a human-robot collaboration framework is studied for the task allocation of collaborative tasks in hybrid assembly cells. An intelligent decision-making algorithm is utilised to allocate the sequential tasks assigned to the robot or the human and to maximise the human-robot coexistence. *Ding et al.* [14] proposed a method to achieve optimal task distribution of assembly operations based on operator's efficiency, mutual influence, layout of the assembly task and setting of the workstation. *Lamon et al.* proposed to exploit agent capabilities, defined as task complexity, agent dexterity and agent effort [8].

Different methods have also been proposed to achieve task decomposition, i.e. extracting the possible set of robot actions for a given assembly tasks. In [15], a heuristic method is proposed that converts an assembly graph into a sequence of tasks. Planning formalisms are also studied to create causal assembly graphs [16]. Decision-making algorithms are built using behavior trees [17], [18] and decision trees [19]. Behavior trees are mathematical models of plan execution, they represent in a modular way the switching between a finite set of tasks. Decision trees, on the other hand, are composed of predicates and are a support tool to model decisions and their possible sequences using control statements.

Aside from the classical task allocation resolution scheme, other methods are able to plan the robot behaviour based on

This work was supported by European Union's Horizon 2020 research and innovation program under Grant Agreement No. 871237 (SOPHIA) and Flanders Make through the programme "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" of the Flemish Government.

[1] Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium, www.brubotics.eu. Corresponding author: `Ilias.El.Makrini@vub.be`

[2] Flexible Assembly, Flanders Make, Belgium

[3] HRI^2 Lab, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy

[4] Interuniversity Microelectronics Centre (IMEC), Belgium

[5] Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milano, Italy

human intention detection and communication means [20], [21]. In [22], a task planner is designed for collaborative human-robot tasks where social acceptance is taken into account. Anticipation of human activities has also been studied using Markov decision processes from camera images [10]. In [23], a model for action preparation and decision making in cooperative tasks is proposed that integrates contextual cues, shared task knowledge and predicted outcome of other's motor behaviour.

This paper presents a novel task allocation algorithm using hierarchical finite-state machines (HFSM) that is based on a more human-centred approach with the goal of improving the working conditions of the worker by taking into account the human and the robot capabilities as well as workload estimates of the user during collaborative tasks. The proposed method improves the task assignment paradigm based on user workload (as in a previous work of the authors [24]) by providing a more comprehensive approach and integrating performance considerations to enhance the productivity of the human-robot assembly task. The developed framework is composed of four main modules that contain different sub-modules to perform the main operations of the task allocation procedure, namely the task selection, the agent capability assessment, the workload estimation, the task execution and communication of instructions to the human. The main contributions of the method are the following:

- The formulation of an online role allocation framework based on HFSM. With HFSM, the method earns in terms of generality (it is possible to model generic human-robot actions) and in easy-to-use (HFSM can be easily programmed by non-experts).
- The combination of a multi-criterion approach to the task allocation problem. Availability, capability, workload and performance are combined to obtain the desired allocation.

The method is evaluated with an industrial task, the assembly of a crusher unit of a smoothie machine, allocated to a human worker and to the collaborative Franka Emika Panda robot.

II. HIERARCHICAL FINITE STATE MACHINE-BASED TASK ALLOCATION

In this paper, we assume that each assembly can be decomposed into tasks, that represent the higher level goals that have to be assigned to an agent. Tasks could be further split recursively in sub-tasks until elementary tasks are reached. For example, the assembly of an electric motor (job) requires the "pick and place" of different objects (tasks). Each pick and place task can be decomposed into 2 sub-tasks: "pick object" and "place object". In turn, pick object is composed of "move to object" and "grasp object", while place is made of "move to target location" and "release object".

HFSM are one of the most used methods to model autonomous agent behaviours. HFSM are essentially finite state machines (FSM) whose states can be other basic FSM or HFSM. A basic FSM M is defined as following:

$$M = \{Q, q_0, F, \Sigma, E\} \tag{1}$$

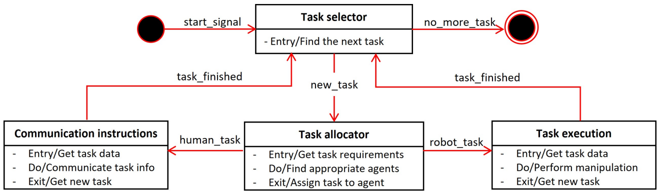


Fig. 1: Task allocation algorithm based on hierarchical finite-state machines. The framework is composed of four main modules (task selector, communication instructions, task allocator and task executor).

where Q is a finite set of states, $q0$ is an initial (or entry) state, F is a set of final (or exit states), Σ is a finite input alphabet, and E a set of transitions, $E \subseteq Q \times \Sigma \times Q$. Further details on HFSM can be found in [25].

The framework is composed of four modules that represent the states of the higher level of the HFSM, namely the task selector, the task allocator, the communication instructor, and the task executor. Figure 1 shows the state diagram with the structure of the proposed task allocation algorithm with the 4 aforementioned states.

When started the HFSM, the $start\_signal$ triggers the launching of the assembly task and activates the task selection among the assembly operations. Once a new task is found (new_task), the task allocator module is enabled to find the suitable agent based on the task requirements and agent information. If the sub-task is assigned to the human, instructions are provided in textual form in the communication instructor. Otherwise, the task executor controls the robot to perform the required operations. Once the task is performed, $task\_finished$ is sent back to the task selector to find the next available sub-task. The assembly finishes when no more tasks (no_more_task) are found. The operation of the 4 main modules will be analysed more in detail.

A. Task Selection

The task selector module is triggered by either the $start\_signal$ or the $task\_finished$ signals. Every task is composed of state machines that represent the sub-tasks. This representation feature allows to model the assembly operations in a comprehensive way with respect to FSM. This enables the grouping of sub-tasks into relevant assembly tasks such as the "shaft assembly" or "motor assembly" tasks shown in Figure 2. Every task outputs a $finished\_task$ signal based on the states of its composing sub-tasks. Each state can be connected to other states in series or in parallel. The states of parallel tasks are combined using vertical forks. The output of the module is a sequence made of actions and sub-actions that presents the assembly operations in a simple and intuitive way. Once all the tasks have been completed, the task selector module will output the no_more_task signal. An example of this module applied to a crusher unit assembly (see section III) is shown in Figure 2.

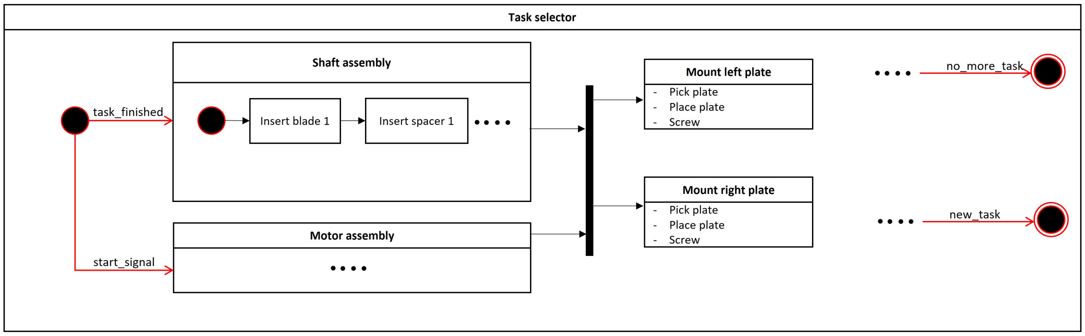


Fig. 2: Task selector finite-state machine, applied to the crusher unit assembly (see Section III). The module is triggered by the $start\_signal$ or the $task\_finished$ signals. Assembly sub-tasks are modelled as finite-state machines.

TABLE I: Example of information about agents[1] .

Agent Info	unit
Payload	[kg]
Precision	[mm]
Reach	[mm]
Speed	[mm/s]
Force range	[N]
Gripper type	[-]
Gripping force	[N]

TABLE II: Example of information about objects.

Object Info	unit
Weight	[kg]
Position	[mm]
Dimension	[mm]
Grasping status	[-]

B. *Task Allocation*

This module is in charge of allocating the task input to the most suitable agent. The structure is shown in Figure 3. First, the available agents are determined. If some agents are available, then, based on the task requirements and agent information, capability of these agents is evaluated. Only the subset of capable agents will be then used to compute the allocation. Finally, the task is assigned considering the current human workload and the agents performance.

1) Availability evaluation: Availability is an important parameter to consider during the task allocation procedure. One of the agent might be busy, e.g. currently performing a task. The module determines the availability state of every agent based on their current task completion. The user informs the system that his/her task is achieved by pressing a button.

2) Capability evaluation: Using the requirements for every elementary task and the agents' and objects' characteristics (Table I and Table II), this module determines the capabilities for the manipulation tasks. Agents' characteristics include information such as payload, speed, reach, equipped gripper type, and gripping force. Basic objects' characteristics include weight, dimension, position, and current grasping status (1 = grasped, 0 = not grasped).

TABLE III: Example of capability table for a pick and place operation with 0 = non-capable and 1 = capable. The final capability values are determined based on equation (2) according on their capabilities on elementary tasks.

Capability	Agent 1	Agent 2
Pick object (c_1)	0	1
Move to object ($c_{1,1}$)	0	1
Grasp object ($c_{1,2}$)	1	1
Place object (c_2)	1	1
Move to target location ($c_{2,1}$)	1	1
Release object ($c_{2,2}$)	1	1
Pick and place object (C)	0	1

The capability of agent i for a task composed of l sub-tasks, each composed of m_j elementary tasks, is defined as:

$$C_i = \prod_j^l c_j \tag{2}$$

where the capability of task j, c_j, given the capability of the elementary task k of the sub-task j, $c_{j,k}$, reads as:

$$c_j = \prod_k^{m_j} c_{j,k} \tag{3}$$

From the above equation, one can observe that an agent becomes non-capable of realising a task if any of its sub-tasks cannot be executed. For example, "Pick object" is composed of two sub-tasks: "Move to object", $c_{1,1}$ and "Grasp object", $c_{1,2}$. Its capability results as $c_1 = c_{1,1}c_{1,2}$ (Equation (3)). See Table III for a complete "pick and place" operation.

3) Workload: In order to determine an approximation of the user workload during human-robot collaboration, a dynamic workload model was established based on virtual spring systems from our previous work [26]. As shown in Figure 4, the human skeleton is modelled by kinematics chains with virtual mechanical elements, creating corresponding joint torques. Torsional springs are attached to the body joints as depicted in Figure 4. The developed model considers the upper body part of the human. The joint torsional springs free position is set to the joint angle that leads to the most ergonomic posture. In this case, these values have been selected based on the Rapid Entire Body Assessment

[0] The griping force represents the maximum holding force of the gripper in Newtons.

[1] The griping force represents the maximum holding force of the gripper in Newtons.

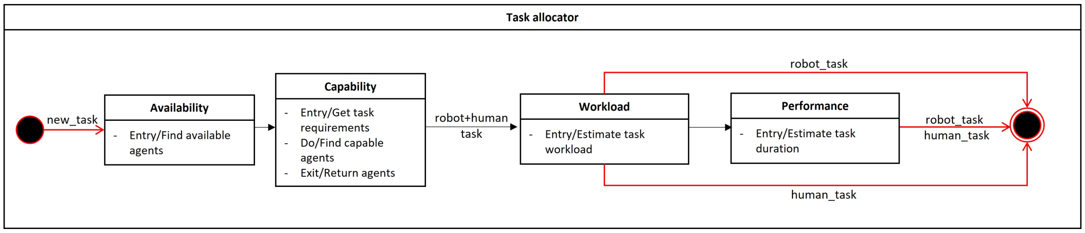


Fig. 3: Task allocation finite-state machine. The module is triggered by the new_task signal. After the checking of agent availabilities, capabilities are determined. The situation where both the human and the robot are capable is handled by the workload state machine that determines if the workload limit is reached. If this is not exceeded, the best agent will be selected based on the estimated task duration determined by task performance machine.

method (REBA), a standard human ergonomics assessment method [27]. The figure also shows the locking of the user's hand using linear springs in the case of extra constraints from the manipulated objects. In this work, however, the stiffness is considered as equal to zero as the hand is allowed to freely move in space.

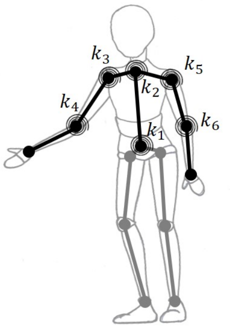


Fig. 4: Human body model using virtual torsional springs attached to the joints: k_1, k_2, k_3, k_4, k_5 and k_6 are the joint's spring stiffness [26].

The workload is calculated using the sum of the elastic energy of the springs [26] that reads as follows:

$$E = \sum_i \frac{1}{2} k_i (\theta_i - \theta_i^*)^2 \tag{4}$$

where θ_i and θ_i^* are respectively the angle and free position of joint i. k_i denotes the stiffness of joint i. The cumulative energy at t=T (cumulative workload) is:

$$W = \frac{1}{T} \int_0^T (E(t) - E_L)\, dt \tag{5}$$

Where E_L represents the workload limit. The latter allows to set a threshold to avoid non-ergonomic situations that might involve risks to the worker's health. In case the human workload exceeds the limit, the task will be assigned to the robot.

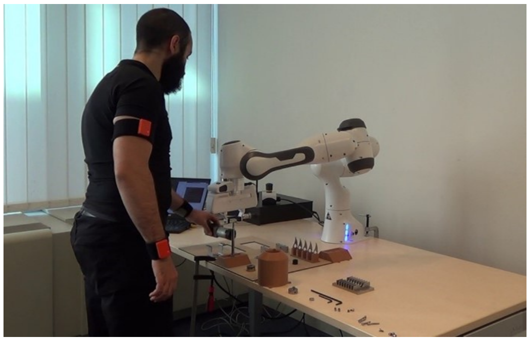


Fig. 5: Experimental setup. The assembly task is performed jointly by the Emika Panda robot and human wearing the Xsens sensors. Link: `https://youtu.be/EQXSCXVDQkI`

4) Performance: The performance of every task is evaluated using an estimate of the task duration. This is measured during fully manual assembly in the case of the human agent and during the collaborative assembly for the robot. In case the human and the robot are both available and capable, and the human workload does not exceed the limit, the agent with the lowest expected task duration will be assigned the sub-task.

C. Task execution and instructions

Depending on the task assignment, i.e. human or robot agent, the task executor or the communication instructor is enabled. The former sends commands to the robot to execute the desired actions based on the received task data such as trajectories, gripper openings, torque/force profiles. In order to assist and guide the user during his/her task, instructions are displayed to the screen in textual format.

III. COLLABORATIVE ASSEMBLY

The assembly task is realized with the collaborative robot Panda from Franka Emika as shown in Figure 5. The dismounted parts of the assembly are laid down on a table along with the screws and the screwdrivers. The human and the robot jointly assemble the crusher unit. The sub-tasks are assigned to one of the agents by taking into account their capability and availability. Since the robot is not equipped, in this case, with a screwdriver, the robot is considered as not capable of performing the screwing task.

The robot is position-controlled at a frequency of 1 kHz and runs under the Robot Operating System (ROS). The master node runs on Matlab and receives the online joint

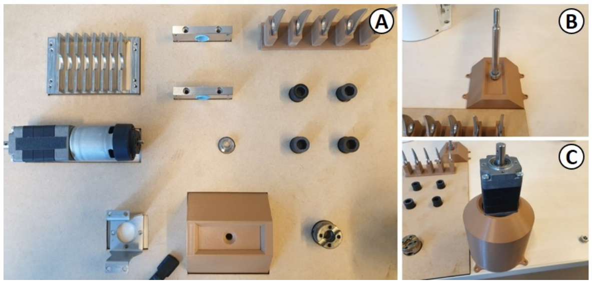


Fig. 6: The dismounted parts of the crusher unit are placed on the table at predetermined positions (A). Kitting trays are used to facilitate the assembly and the manipulation of the pieces by the robot such as the shaft (B) and motor (C) sub-assemblies.

angles from the Xsens sensors through a ROS layer. A fixed-position approach is used where the objects have a determined position on the table. The locations are saved for every part. During human interventions, instructions are given via textual information to the operator. The mapping from the 3D coordinates to the robot joint angles is performed with the MoveIt planning framework[2].

The assembly task consists in assembling the crusher unit of a smoothie machine by both the human and the robot. The latter is composed of several parts: blades, spacers, couplers, etc, as shown in Figure 6. Kitting trays are utilised to set the position of the objects on table and to orient some parts so that they can be grasped by the robot (as it can be seen in Figure 6).

IV. EXPERIMENTAL VALIDATION

The crusher unit assembly consists of 28 tasks, as described in Table IV. The related task sequence, from which it is possible to devise the task selector structure, diagram is depicted in Figure 7. Sub-tasks are grouped in the form of "pick and place" operations to avoid the unreal situations where the pick tasks and concurrent place tasks could be performed by two different agents.

The cooperative assembly is executed with two different workload limits. 0.4 and 0.25, respectively. Figure 8 give the task allocation results of the crusher unit assembly in these two settings. One can notice that some tasks cannot be performed by the robot such as the screwing (T16, T17) or the manipulation of sub-assemblies (T26, T27). In the case where the workload limit equal to 0.25, the task T20 (pick and place motor attachment), is performed by the robot since the workload threshold is exceeded (Figure 9). Thanks to the hierarchical state machine, it is possible to run parallel tasks where both the robot and the human perform the required operations. Thanks to the performance module, the total assembly time is improved by 31% (workload limit $= 0.4$) and 34% (workload limit $= 0.25$), i.e. compared to a solution allocated to the same two agents where no parallel tasks are implemented.

Figure 9 show the workload throughout the assembly process with workload limits set to 0.25 and 0.4. This is computed from the joint angles measured by the Xsens sensors and determined by the human spring model. One can observe that the workload limit of 0.25 is rapidly exceeded in Figure 9a. The robotic intervention during task T20 at $t = 20\ s$ allows to decrease the workload below the latter value. One can also notice that the workload value at the end of the assembly is smaller in the lower workload limit case (0.8 vs 0.9). It should be noted that the performance has been favoured over the workload. Therefore, to minimise inefficient waiting times, every time an agent is allocated once an agent becomes available, a task will be assigned to him.

V. CONCLUSIONS AND FUTURE WORK

The working condition of the operator is an important factor during human-robot collaboration, especially in the context of human-centered approach of industry 5.0. By assisting the user with the robot, it is possible to reduce the risk of MSDs and physical fatigue.

This work presented a novel framework that takes into account multiple aspects of human-robot teams such as capabilities, availabilities, workload and performances. Hierarchical finite-state machines are used to manage the logic of the task allocation processes to assign the assembly sub-tasks to either the human or the robot.

As shown by the collaborative assembly of the smoothie machine's crusher unit, the proposed framework is able to adapt the level of the human assistance during the tasks by setting the workload limit to the desired level.

Current limitation of the method lies in the use of workload and task duration estimations of the user intervention. Possible improvements include the learning of the task parameters to better predict and adapt the task allocation to the human behavior.

REFERENCES

[1] S. Nahavandi, "Industry 5.0—a human-centric solution," *Sustainability*, vol. 11, no. 16, p. 4371, 2019.

[2] T. Yu, J. Huang, and Q. Chang, "Mastering the working sequence in human-robot collaborative assembly based on reinforcement learning," *IEEE Access*, vol. 8, pp. 163 868–163 877, 2020.

[3] U. Kartoun, H. Stern, and Y. Edan, "A human-robot collaborative reinforcement learning algorithm," *Journal of Intelligent & Robotic Systems*, vol. 60, no. 2, pp. 217–239, 2010.

[4] T. Yu, J. Huang, and Q. Chang, "Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning," *Journal of Manufacturing Systems*, vol. 60, pp. 487–499, 2021.

[5] 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, 2017.

[6] F. Chen, K. Sekiyama, F. Cannella, and T. Fukuda, "Optimal subtask allocation for human and robot collaboration within hybrid assembly system," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1065–1075, 2014.

[7] P. Tsarouchi, G. Michalos, S. Makris, T. Athanasatos, K. Dimoulas, and G. Chryssolouris, "On a human–robot workplace design and task allocation system," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 12, pp. 1272–1279, 2017.

[8] 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.

[2] https://moveit.ros.org/

TABLE IV: Task descriptions of the crusher unit assembly.

Task	Description	Task	Description	Task	Description
T1	Pick and place shaft	T11	Pick and place left side plate	T20	Pick and place motor attachment
T2	Pick and place blade 1	T12	Pick and screw nut	T21	Screw motor attachment
T3	Pick and place spacer 1	T13	Pick and place case	T22	Pick and place motor coupler
T4	Pick and place blade 2	T14	Pick and place assembled shaft	T23	Screw motor coupler
T5	Pick and place spacer 2	T15	Pick and place left side plate	T24	Pick and place shaft coupler
T6	Pick and place blade 3	T16	Screw left side plate	T25	Screw shaft coupler
T7	Pick and place spacer 3	T17	Pick and place right side plate	T26	Pick and place assembled motor
T8	Pick and place blade 4	T18	Screw right side plate	T27	Connect motor to shaft
T9	Pick and place spacer 4	T19	Pick and place motor	T28	Screw motor to shaft
T10	Pick and place blade 5				

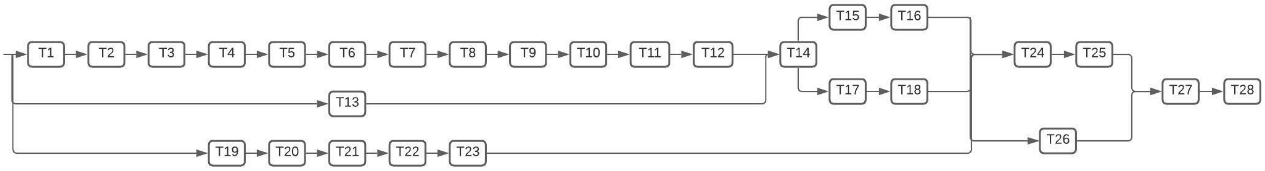
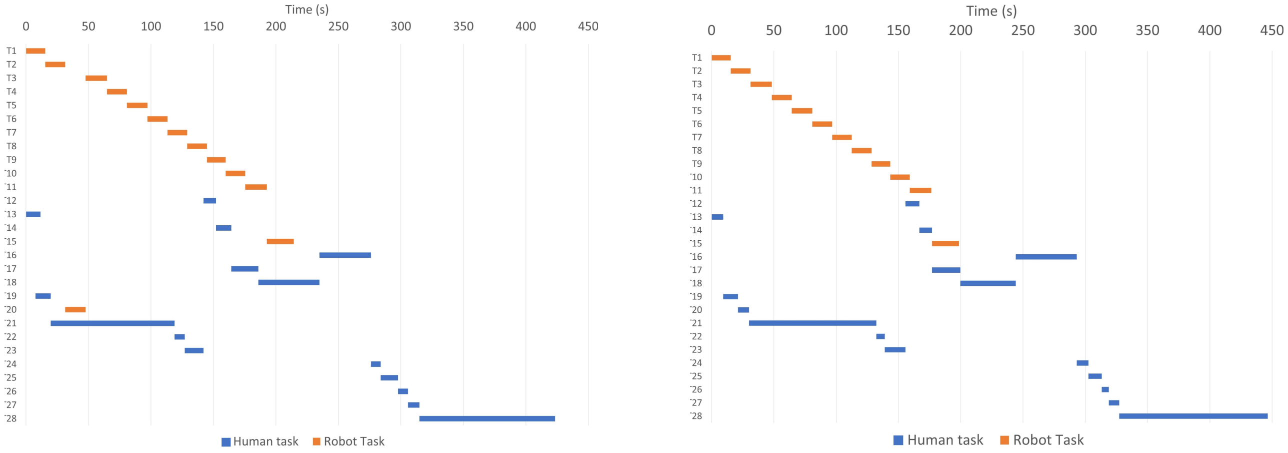


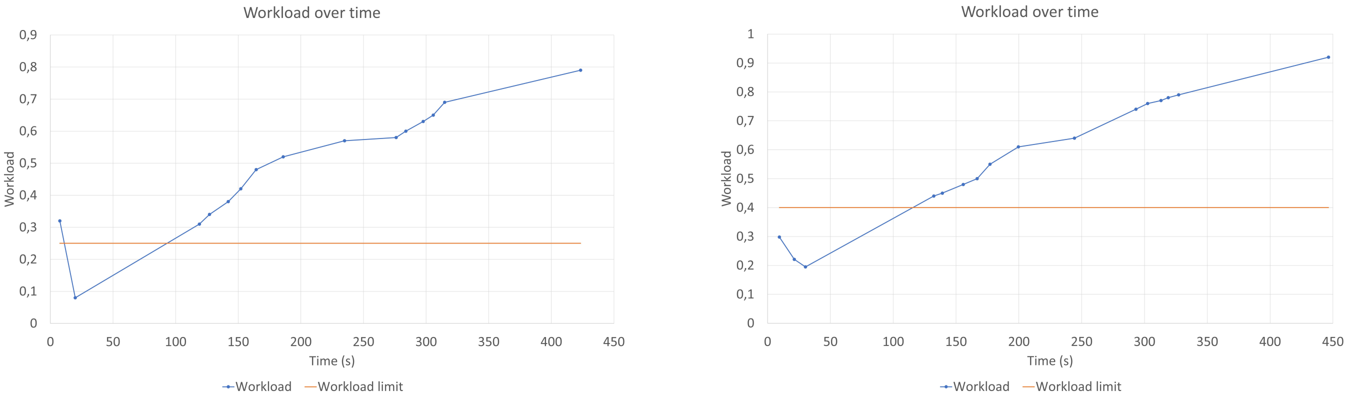
Fig. 7: Assembly task sequence of the smoothie machine’s crusher unit



(a) Workload limit = 0.25.

(b) Workload limit = 0.4.

Fig. 8: Gantt chart of the crusher unit assembly.



(a) Workload limit = 0.25.

(b) Workload limit = 0.4.

Fig. 9: Human workload over time during execution of the crusher unit assembly.

[9] A. Roncone, O. Mangin, and B. Scassellati, “Transparent role assignment and task allocation in human robot collaboration,” in *Robotics*

and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017, pp. 1014–1021.

[10] H. S. Koppula, A. Jain, and A. Saxena, "Anticipatory planning for human-robot teams," in *Experimental Robotics*. Springer, 2016, pp. 453–470.

[11] C. Liu, J. B. Hamrick, J. F. Fisac, A. D. Dragan, J. K. Hedrick, S. S. Sastry, and T. L. Griffiths, "Goal inference improves objective and perceived performance in human-robot collaboration," in *Proceedings of the 2016 international conference on autonomous agents & multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 940–948.

[12] M. S. Malvankar-Mehta and S. S. Mehta, "Optimal task allocation in multi-human multi-robot interaction," *Optimization Letters*, vol. 9, no. 8, pp. 1787–1803, 2015.

[13] G. Michalos, J. Spiliotopoulos, S. Makris, and G. Chryssolouris, "A method for planning human robot shared tasks," *CIRP Journal of Manufacturing Science and Technology*, vol. 22, pp. 76 – 90, 2018.

[14] H. Ding, M. Schipper, and B. Matthias, "Optimized task distribution for industrial assembly in mixed human-robot environments-case study on io module assembly," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2014, pp. 19–24.

[15] M. Helmert, "A planning heuristic based on causal graph analysis." in *ICAPS*, vol. 16, 2004, pp. 161–170.

[16] C. Lenz, S. Nair, M. Rickert, A. Knoll, W. Rosel, J. Gast, A. Bannat, and F. Wallhoff, "Joint-action for humans and industrial robots for assembly tasks," in *RO-MAN 2008-The 17th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2008, pp. 130–135.

[17] F. Fusaro, E. Lamon, E. D. Momi, and A. Ajoudani, "An integrated dynamic method for allocating roles and planning tasks for mixed human-robot teams," in *2021 30th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2021, pp. 534–539.

[18] T. Wang, D. Shi, and W. Yi, "Extending behavior trees with market-based task allocation in dynamic environments," in *Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control*, 2020, pp. 1–8.

[19] W.-Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[20] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, "Gestures for industry: intuitive human-robot communication from human observation," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, 2013, pp. 349–356.

[21] F. Fusaro, E. Lamon, E. D. Momi, and A. Ajoudani, "A human-aware method to plan complex cooperative and autonomous tasks using behavior trees," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, 2021, pp. 522–529.

[22] S. Alili, M. Warnier, M. Ali, and R. Alami, "Planning and plan-execution for human-robot cooperative task achievement," in *19th international conference on automated planning and scheduling (ICAPS)*, 2009.

[23] E. Bicho, W. Erlhagen, L. Louro, and E. C. e Silva, "Neuro-cognitive mechanisms of decision making in joint action: A human–robot interaction study," *Human movement science*, vol. 30, no. 5, pp. 846–868, 2011.

[24] I. El Makrini, K. Merckaert, J. De 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.

[25] M. Yannakakis, "Hierarchical state machines," in *IFIP International Conference on Theoretical Computer Science*. Springer, 2000, pp. 315–330.

[26] I. El Makrini, G. Mathijssen, S. Verhaegen, T. Verstraten, and B. Vanderborght, "A virtual element-based postural optimization method for improved ergonomics during human-robot collaboration," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2022.

[27] S. Hignett and L. McAtamney, "Rapid entire body assessment (reba)," *Applied ergonomics*, vol. 31, no. 2, pp. 201–205, 2000.