

A capability-aware role allocation approach to industrial assembly tasks

Lamon, Edoardo; De Franco, Alessandro; Peternel, Luka; Ajoudani, Arash

DOI

[10.1109/LRA.2019.2926963](https://doi.org/10.1109/LRA.2019.2926963)

Publication date

2019

Document Version

Accepted author manuscript

Published in

IEEE Robotics and Automation Letters

Citation (APA)

Lamon, E., De Franco, A., Peternel, L., & Ajoudani, A. (2019). A capability-aware role allocation approach to industrial assembly tasks. *IEEE Robotics and Automation Letters*, 4(4), 3378-3385.
<https://doi.org/10.1109/LRA.2019.2926963>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

A Capability-Aware Role Allocation Approach to Industrial Assembly Tasks

Edoardo Lamon^{1,2}, Alessandro De Franco^{1,3}, Luka Peternel⁴, and Arash Ajoudani¹

Abstract—The deployment of industrial robotic cells based on lean manufacturing principles enables the development of fast-reconfigurable assembly lines in which human and robotic agents collaborate to achieve a shared task. To ensure the effective coordination of the shared effort, each task must be decomposed into a sequence of atomic actions that can be assigned either to a single agent or to the combination of more agents, according to a defined metric. While task allocation is a general problem and has been discussed intensively in other fields, less effort has been devoted in industrial scenarios involving mixed human-robot teams and in particular, to the factors that should be considered in allocating tasks among a heterogeneous set of agents in collaborative manufacturing scenarios. In this paper, we investigate the agent characteristics that should be considered in the task allocation problem of fast-reconfigurable systems in industrial assembly processes. First, we introduce a set of indices, namely task complexity, agent dexterity, and agent effort, to evaluate agent performance with respect to a task. Second, we propose an offline allocation algorithm that combines the performance indices to assign optimally the task to the team agents. Finally, we validate the framework in a proof-of-concept collaborative assembly of a metallic structure. The results show that the workload is shared through the agents according to their particular physical capabilities and skill levels. A subjective analysis of the proposed collaborative framework on 12 healthy participants also validated the intuitiveness-of-use and improved performance.

Index Terms—Physical Human-Robot Interaction, Assembly, Task Planning, Intelligent and Flexible Manufacturing.

I. INTRODUCTION

DESPITE the recent progress in developing powerful and reliable robotic arms for industry, most of the manufacturing tasks are not yet automated. The reason is the high cost of these complex robotic work cells, due to the requirements and expertise in project management, in developing custom solutions and in specialised programming. Moreover, when the robotic cell changes due to the assignment of a new product

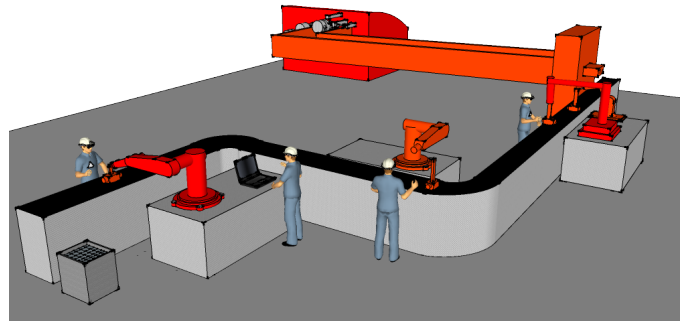


Fig. 1. Collaborative assembly task in a manufacturing factory: an example of a robotic work cell deployed with cobots.

or the replacement of the robot, a large amount of time and overhead costs are required to make the system operational. The development of lightweight collaborative robots (cobots) and their integration into robotic cells (see Fig. 1), according to lean manufacturing principles [1], brought several benefits in the manufacturing process. While increasing user-friendliness (ease of use), they not only improve production quality and save time but also eliminate human ergonomic constraints on the production line and prevent workers injuries [2]. Cobots are not only intrinsically safe due to the torque limitations on the actuation system, but also on the control level can feature impedance controllers that can regulate the level of compliance according to the task [3] and protocols to safely interact with unstructured environments [4]. In this way, cobots are enabled to share the same workspace with humans, adding power and high-precision motions to the superior task understanding and cognitive capabilities of humans. Moreover, they can be also easily programmed through learning by demonstration techniques [5]. These characteristics promoted development in the field of physical human-robot interaction (pHRI). A complete overview of the state of art and examples of industrial applications of pHRI can be found in [6]–[9].

In small and medium-size companies, which are characterised by small-scale production and limited resources, a robotic team is expected to be able to perform a variety of different tasks. In this context, collaborative robotic work cells allow flexible role interchange and fast adaptation of the cell itself to new products and configurations. However, traditional applications of pHRI involve mainly coexistence of human and robots in the same workspace. To achieve full collaboration, agents need to share not only the place but also the goals. To this aim, each agent, according to its capabilities, should be able to understand which actions are required to perform a complex task. In this paper, we will focus on the agent role

Manuscript received: February, 24, 2019; Revised May, 31, 2019; Accepted June, 25, 2019. This paper was recommended for publication by Editor Allison M. Okamura upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Amazon Research Awards 2018

¹ Edoardo Lamon, Alessandro De Franco, and Arash Ajoudani are with the Human-Robot Interfaces and Physical Interaction Laboratory, Department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy. edoardo.lamon@iit.it

² Edoardo Lamon is with Department of Information Engineering, Università degli Studi di Pisa, Via Girolamo Caruso, 16, 56122 Pisa, Italy.

³ Alessandro De Franco is with Department of Electronics, Information and Bioengineering, Politecnico di Milano, Via Giuseppe Ponzio, 34, 20133, Milano, Italy.

⁴ Luka Peternel is with Department of Cognitive Robotics, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands.

Digital Object Identifier (DOI): see top of this page.

allocation in collaborative work cell scenarios, where assembly processes are mastered and the desired duration of the tasks is well-known. The role of each agent of the team (in literature generally a single cobot and a single human) can be embedded at the control level [10]–[12] or could be planned beforehand. *Mortl et al.* [11] proposed and evaluated different dynamical task allocation in cooperative load transport tasks. To face the problem of load sharing, the authors presented a crafty policy analysis and task decomposition in steering and progressing. While this control algorithm optimised the performance of the cooperative load transport task in terms of human effort, it might not be optimal or applicable to a different task.

A more general approach to the dynamical role allocation problem is presented by *Peternel et al.* [12]. In this paper, the manipulator predicts the human motor intention through the measurements of EMG sensors placed on the muscles of the shoulder joint, enabling a dynamic adaptation of the robot effort during the task performance. Moreover, in the collaborative sawing example, the stiffness values of the robot impedance controller are related to the level of muscle activation. In this way, the robot control parameters do not require an offline tuning and the emerging robot behaviour will mimic human behaviour. Furthermore, to optimise robot performance, the control policy is selected depending on the particular task. From these examples, it is clear that the best control strategy should be chosen according to the task and it's not possible to give an optimal general control framework that is able to accomplish a large class of tasks. On the other hand, since industrial tasks are well understood, it is also possible to combine some prior knowledge on the role allocation at the planning level. The problem of coordination of a team of robots has been intensively studied in other fields, such as environment exploration and monitoring, surveillance, transport, and playing robotic soccer [13], [14]. A combinatorial formal analysis of task allocation in large and homogeneous multi-robot systems (MRTA) is presented by *Gerkey and Mataric* [15].

A common approach to the problem of the representation of a complex task and its decomposition into a sequence of atomic actions in industrial assembly processes exploits AND/OR graphs. This structure is an efficient implementation of a state transition graph, which requires fewer nodes and simplifies the search for feasible plans [16]. A recent application of AND/OR graphs to decompose assembly tasks in the field of pHRI is offered by *Johannsmeier and Haddadin* [17]. The authors introduced a framework on three different levels: the team-level assembly task planner, that solves the allocation problem of complex task through an A* search on the AND/OR graph, the agent-level skill planning, in which the planning of the action is implemented on the selected robot and the online skill execution level that consists of trajectory planning and control. One interesting feature is that the algorithm deals at the team-level with different agents with no explicit distinction between robots and workers: the discrimination is embedded in the heuristics, since they choose different cost functions depending on the agent, such as execution time and power consumption for robots and attention level and workload measurements for humans. While

AND/OR graphs represent a well-studied method to approach task decomposition problems, less attention has been devoted in understanding the agent characteristics and the factors that should be considered in solving the task allocation problem in industrial collaborative tasks.

In order to go beyond the state of art, we propose three different measurable indices to represent the agent features at the team-level. These indices reflect the agents' nature (human or robot) and their kinematic and dynamic characteristics. The first index, *task complexity*, enables the discrimination of human and robot capabilities in executing a high-level particular industrial task. Task complexity, by itself, does not distinguish between agents of the same type (i.e. between two robots, or two humans). For this reason, we introduced two other indices, i.e., *agent dexterity* and *agent effort* to take into account this aspect. Agent dexterity is a metric based on the kinematic properties of the agent, which is used to evaluate its capability to plan and move in the environment. Similarly, the agent effort index takes into account the dynamic behaviours reflecting the agents' ergonomic and safety constraints. In this paper, we propose kinematic reachability as the core component of agent dexterity, and human and robot fatigue, for the agent effort. To obtain the desired capability-based task allocation, all indices are combined together by means of a weighted sum to shape a reasonable heuristic for the search algorithm. These heuristics aim to associate each atomic action to an agent, based on the identified capabilities (see also Fig. 2). The method is evaluated with a proof-of-concept assembly of a metallic structure. First, the agent execution cost for each action is computed. Second, the role allocation algorithm assigns the actions to the agents in the team through a cost-minimisation principle. Finally, the task is executed by the agents relying on the algorithm results.

II. CAPABILITY-BASED TASK ALLOCATION

The main purpose of this paper is to define suitable metrics for the task allocation problem in collaborative industrial scenarios. The framework consists of three different layers. At the lower level, the *agent level*, actuation, sensing, control and trajectory planning are implemented on each agent. At the middle level, namely *team level*, the high-level actions are assigned to each agent and planned using state machines and advanced algorithms combined with the expertise of the process supervisors. Finally, at the higher level, called *assembly level*, the problem of the assembly task decomposition and assignment is handled.

We will focus now on a single mechanical assembly M and a single team of agents $W = \{w_1, \dots, w_n\}$. We define $A_i = \{a_1, \dots, a_m\}$ the set of skills (actions) that can be executed by an agent $w_i \in W$. At this level, there's no explicit difference between the skill of each agent. A proper representation of an assembly plan is offered by the AND/OR graph since it promotes the parallel execution of the actions. Using this representation, each assembly task can be decomposed into a sequence of actions a_j . What we want to obtain is the allocation of an agent w_i to each of the task actions a_j , denoted $w_i \rightarrow a_j$. In order to do that, it is necessary evaluate the cost $c(w_i, a_j)$ of each agent in executing the action a_j .

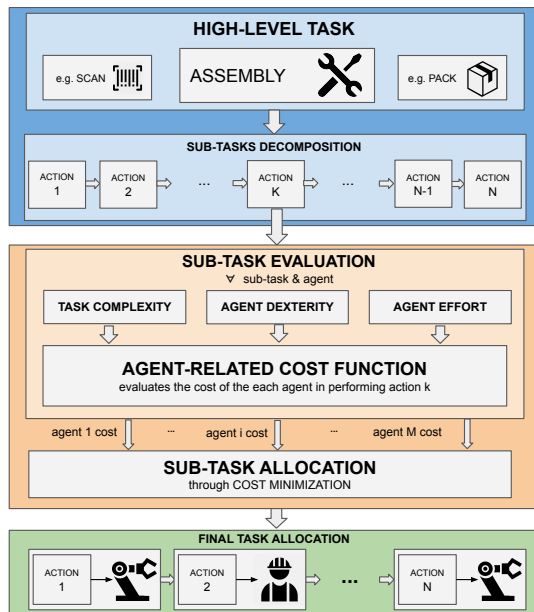


Fig. 2. Flow diagram from the high-level task to the assigned sequence of actions. Each high-level task is first decomposed into a sequence of actions required to achieve the task. Then the algorithm evaluates the suitability of each agent in performing each action, according to the proposed metrics. Finally, the action allocation is performed according to the minimisation of the overall performance cost.

The desired allocation is obtained when the total assignment cost is minimised. In this paper, the cost function $c(w_i, a_j)$ is designed according to the agent's capability. A more detailed explanation of this three-layer structure and a way to represent assembly plans with AND/OR graphs can be found in [17].

III. METRICS FOR THE COST FUNCTION

One of the first attempts to distinguish between human and robot capabilities was proposed by *Fitts et al.* at the American National Research Council in 1951 [18], in which a list of what man can do better than a machine is presented. Although this list is outdated and some assumptions might not be true anymore (machine learning algorithms emulates inductive reasoning for instance), the basics concepts are still valid: robots perform hard, repetitive and precise tasks better than humans, while humans can execute cognitive complex tasks thanks to their capability to learn quickly and react to unseen situations. Recent research tried to make the decision process objective and automatic, considering mainly corporate goals as capability indicators. For example, *Ranz et al.* [19] proposed an algorithm to compute these indicators from process time, additional investment and process quality, estimated from work measurements techniques like MTM (*method time measurement*) and RTM (*robot time and motion*). These indicators are not able to discriminate, among a team composed of more than a robot (or more than a human), which robot (or human) should perform the task.

To account the general heterogeneity of the agents in collaborative work cells, we propose three different metrics that belong to the physical characteristics of the agents: *task complexity*, *agent dexterity* and *agent effort*. Task complexity measures if an agent is able to perform the current task according to the specification required by the production

TABLE I
TAXONOMY OF MANUFACTURING ASSEMBLY ACTIONS.

Action	Atomic primitives
Align	MOVE
Pickup	MOVE, GRASP
Insert	MOVE
Slide	MOVE, APPLY a FORCE
Retract	MOVE
Transport	MOVE
Place	GRASP, MOVE
Tool action	MOVE, APPLY a FORCE
Drill	APPLY a FORCE
Hold	GRASP, APPLY a FORCE

process. In particular, some actions may not be feasible for a cobot, or may not be implemented. While task complexity deals with the set of skills of each agent separately, there might be a case where two or more agents are suitable for the same task. For this reason, we introduce also agent dexterity and effort. The first is required to favour agents with higher motion capability, while the second to distribute the efforts equally among the agents. Moreover, each metric can be described by one or more core components. A similar approach, based on agent capabilities, is proposed by *Tsarouchi et al.* [20] and *Michalos et al.* [21]. In our scenario, the assembly time is dictated by the manufacturing process and will not be considered as one of the metrics. Indeed, we are not just interested in increasing the productivity of the work cell. The main goal is the improvement of the work ergonomics by minimisation of human risk factors, through a capability-based approach.

A. Task complexity

In a fast-reconfigurable work cell, the approach is to exploit general purpose robots (like cobots) to execute as many tasks as possible, within platform limits. In general, a cobot is able to successfully perform point-to-point motions, apply forces/torques, and grasp simple objects. On the other hand, complex coordinated motions, precise grasps, or in-hand manipulation, for instance, require more sophisticated perception and manipulation control, that might not be in agreement with the current levels of robot decisional autonomy. For these reasons, it is more convenient (and time-efficient) to assign such complex tasks to human workers, that are able to achieve high performances with lower planning effort. Having that in mind, to describe cobot contributions, we consider the taxonomy for manufacturing assembly tasks proposed by *Huckaby and Christensen* in [22], listed in Tab I. Most of the skills presented in the taxonomy can be further decomposed into 3 main action primitives: MOVE, GRASP and APPLY a FORCE¹.

A possible choice of the task complexity metric is the following:

$$T(w_i, a_j) = \begin{cases} 0, & \text{if } a_j \in A_{w_i}; \\ \infty, & \text{otherwise,} \end{cases} \quad (1)$$

¹Some skills may require also perception-based primitives that are not considered in this analysis.

where $T(w_i, a_j)$ is the cost of the worker w_i in performing action a_j and A_{w_i} is the set of agent feasible/known actions. The aim of the binary metric is to discriminate the actions that the agent is able to perform from the ones that cannot perform or has not been programmed for. Nevertheless, this metric could be designed in such a way that also performance or action duration are considered.

B. Agent dexterity

In general, in a heterogeneous team, more than a single agent might be able to execute an atomic action. Each of these agents may also have different kinematic and dynamic characteristics, especially in teams composed of different humans and robots. Moreover, even in a set composed by the two agents of the same type (two robots or two humans), one might be more suitable to accomplish the task due to multiple factors, such as proximity to the goal, higher manipulability, or grasping capabilities. To take this into account, we group the kinematic characteristics into the agent dexterity metric.

To measure agent dexterity, we exploited the concept of *reachability map*, introduced by *Zacharias et al.* [23]. This map describes the agent Cartesian workspace structure, in terms of solutions to the inverse kinematics algorithm. To compute the map, the Cartesian space is first approximated with a set of equidistant spheres. For each sphere, a fixed number of randomly distributed points N according to a uniform distribution is sampled on the surface of the sphere. Then, a frame is assigned to each of these points, with the z -axis aligned with the radial direction. These frames represent possible Cartesian Tool Centre Point (TCP) poses. For each of these frames is computed the inverse kinematic, if the solution exists. If it does not exist, the frame is rotated around the z -axis of a fixed step size. The number of valid inverse kinematics solutions for each sphere is R . Accordingly, the *reachability index* D is computed by:

$$D = \frac{R}{N}, \quad \text{with } R \leq N. \quad (2)$$

The main advantage of this map, instead of, e.g., the manipulability map, is that it describes the characteristics of the Cartesian workspace, instead of the configuration space. For each valid pose of a redundant agent in the Cartesian space, there exists an infinite number of possible solutions to the inverse kinematics problem. We are interested in just one of them and the inverse kinematics algorithm already selects it. Hence, instead of describing the Cartesian space from a much larger configurations space, we restrict the problem to a subspace of it that is formed by the solutions of the inverse kinematics. In the end, for each point in the Cartesian space we have an index that describes how close we are to the singularities. The smaller the index, the closer it gets to the singularity.

To be able to compute this metric on human workers, we developed a 31 DOFs model of a human. The kinematic parameters of the human model, such as links length or joint angle limits, have been extracted from anthropometric tables [24]. The resultant kinematic structure has been translated into a Unified Robot Description Format (URDF) file. In this way,

we could treat the human model in the same way as the model of a floating-base humanoid robot. The inverse kinematics solution is obtained with the modular whole-body control library *OpenSoT* [25]. OpenSoT solves the nonlinear inverse kinematics as a quadratic programming optimisation problem based on different solvers. It includes also the possibility to specify prioritised constraints on the controller, that is particularly useful in the case of highly redundant robots, and to consider joint velocity limits as well as joint position limits. In the case of manipulators, we set as constraints the Cartesian TCP pose and, with a lower priority, a postural at the joint level. For the human model, we also add, with a higher priority, Cartesian constraints on the feet and on the waist link, to ensure the stability of the computed configuration. The constraint on the feet, in particular, allows the inverse kinematics algorithm to generate configurations with at least one foot placed on the ground. We fixed the TCP on the gripper of the cobot and on the hand of the human.

In our framework, we assume that each action has to be performed in a particular position in space $x(a_j)$, defined by the production process. For each agent in the team, $x(a_j)$ is first mapped in its own reference frame and then the reachability index is computed. For the human model, the reachability index is computed twice, once with the TCP Cartesian constraint on the right hand and once with the left one. The final reachability index is the maximum between the two previously computed. The radius of the sphere centred in the $x(a_j)$ is 25 mm. For each sphere, 100 points are sampled on the surface and each frame is rotated around its z -axis by 60 degrees. The computation of the whole reachability map, for all the Cartesian space, is considerably expensive and the best approach would be to compute the map offline once, then save the values for future use. Moreover, in the proposed metric, we considered just the relative motion of the end-effector with respect to a frame fixed on the body of the agent, that we could call *static dexterity*. However, for humans and mobile robots, we can also consider a *mobile dexterity* as an additional cost, that is actually the motion of the floating base in the space. A simple way to compute this cost is presented by *Vahrenkamp et al.* [26].

C. Agent effort

In manufacturing scenarios, repetitive and force-demanding assembly tasks represent one of the risk factors to human health. Awkward postures, exerting high forces to achieve a task, using heavy or vibrating tools, pressing on hard surfaces are just some of the possible cause of fatigue, pain and sometimes injury (musculoskeletal disorders) in the workplace. What we would like to have is a metric that is able to estimate some features of the dynamic behaviour of the agent in executing each action. For instance, it should recognise the repetitiveness or heaviness of the task. As a core component of this metric, we selected fatigue. We exploited a model introduced first by *Ma et al.* [27] and then used by the authors of the paper to characterise both human muscle activation [12] and also robot motor usage [28] in human-robot collaborative scenarios. In addition, it has been also exploited for dynamic job rotation to balance the workload of human workers in

flexible assembly work cells [29]. The dynamic behaviour of a fully actuated fixed-base n-DOFs robot is described by:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q})\mathbf{f}_{ext}, \quad (3)$$

where $\mathbf{q} \in \mathbb{R}^n$ is the joint angles vector, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ is the vector of Coriolis and centrifugal torques and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ denotes the gravity component. Moreover, $\mathbf{J}^T(\mathbf{q})\mathbf{f}_{ext} \in \mathbb{R}^n$, represents the torque due to the action of an external force $\mathbf{f}_{ext} \in \mathbb{R}^m$ (contact forces, external loads, etc.), with $m \leq 6$, projected at joint level by the geometric Jacobian matrix $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$. The dynamical characteristics of the performed action are summarised in $\boldsymbol{\tau} \in \mathbb{R}^n$, the vector of input joint torques. The dynamic behaviour of a floating base robot is slightly more complex, since we should consider an augmented model with virtual joints and contact forces [30]. But, for the sake of clarity, we will consider the simpler model (3), without any loss of generality.

The complete *joint fatigue-recovery* model is based on a first-order dynamic system:

$$\frac{dV_i(t)}{dt} = \begin{cases} (1 - V_i(t)) \frac{|\tau_i(t)|}{C_i}, & \text{if } \tau_i(t) \geq \tau_{i,th} \\ -V_i(t) \frac{r}{C_i}, & \text{if } \tau_i(t) < \tau_{i,th} \end{cases} \quad (4)$$

where V_i , C_i , and τ_i are the fatigue level, endurance capacity, and the torque of joint i , respectively. The capacity C_i is a physical parameter related to the maximum torque that could be supplied by the actuators, namely muscles in humans and motors in robots, that scales the contribution of the torques. In the recovery equation, r is the recovery rate of the actuator. Finally, $\tau_{i,th}$ is the torque threshold, that represents the value of the torque that discriminates fatigue and recovery situations. A possible choice of $\tau_{i,th}$ could be the 10% of the maximum torque supplied by the actuator or the nominal torque in static "rest" conditions. The closed-form solution of model (5) clearly shows how the fatigue of the current action is related to the previous accumulated fatigue (up to the action j-1). The j-action related accumulated fatigue $V_{j,i}$ (the subscript j highlights the dependency on the action a_j), defined in the period (t_0, t) , can be computed through:

$$V_{j,i}(t_0, t) = \begin{cases} 1 - (1 - V_{j-1,i}) \exp\left(-\frac{\int_{t_0}^t |\tau_i(\sigma)| d\sigma}{C_i}\right), & \text{if } \Delta\tau_i(t_0, t) \geq \tau_{i,th} \\ V_{j-1,i} \exp\left(-\frac{r(t-t_0)}{C_i}\right), & \text{if } \Delta\tau_i(t_0, t) < \tau_{i,th} \end{cases} \quad (5)$$

where $\Delta\tau_i(t_0, t)$ represents the sequence of torques from t_0 to t required to achieve action a_j . In this paper, for the sake of simplicity, we will focus on the fatigue caused by external loads in static conditions. This assumption simplifies both the model (3) and the floating-base robot dynamic model, leading to $\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{f}_{ext}$, where \mathbf{f}_{ext} now is the gravity force due to the presence of an external load either on the end-effector of the robot or held by the human. At this point, the computation of $\boldsymbol{\tau}$ requires the kinematic parameters of the agent and the mass of the load.

The overall agent fatigue accumulated for action a_j is computed by averaging the fatigue at joint level:

$$V(a_j) = \frac{1}{n} \sum_{i=1}^n V_{j,i}. \quad (6)$$

Instead of this simple average, it is possible to introduce a weighted average with different weights on some specific joints, for instance low-back or the shoulder of the human, depending on the particular task.

IV. COST FUNCTION DESIGN

The desired cost function should incorporate the previous indices into a single scalar value that should represent how much an agent is suitable for the action. In particular, we would like to assign an infinite cost in some predefined situations. For instance, task complexity $T(w_i, a_j)$ defined in (1) already takes the infinite value if the action is not feasible or known by the agent. Similarly, we design the overall cost function $c(w_i, a_j)$ by

$$c(w_i, a_j) = \beta_{T_i} T_i(a_j) + \beta_{D_i} |\log D_i(a_j)| + \beta_{V_i} |\log(1 - V_i(a_j))|, \quad (7)$$

where $T_i(a_j)$, $D_i(a_j)$, $V_i(a_j)$ are the indices defined in (1), (2) and (6) and $\beta_{T_i}, \beta_{D_i}, \beta_{V_i} \geq 0$ are the indices weights². The term $|\log D_i(a_j)|$ is needed to ensure that actions with null reachability index, for instance outside the agent workspace, have a high cost. On the other hand $|\log(1 - V_i(a_j))|$ avoids that heavy loads are assigned to agents with low payload. It is important to highlight that, in collaborative assembly scenarios, most of the tasks can be performed manually (even faster!), but the quality of the execution may decrease over time until eventual harm to the human worker. For this reason, to reduce the risk of injury, the general approach consists in assigning repetitive and force-demanding tasks to the robotic agents. This behaviour is obtained by a higher weighting of the fatigue component in the cost function of human agents, while in case of robotic agents the reachability index is associated with a higher weight. This is possible since the weights in (7) can be tuned depending on the particular agent.

V. EXPERIMENTAL RESULTS

The proposed metric-based allocation were tested on a fast-reconfigurable and flexible assembly line scenario. We reproduced a collaborative assembly task with a mixed team of agents and their role have been assigned by the algorithm described above. For the proof-of-concept demonstration we physically performed the assembly of a metallic structure made of two aluminium profiles of different weight (392 g and 865 g) and sizes, held together through a corner joint (58 g) with screws and nuts³. The team is composed of

²With the current choice of task complexity metric the only reasonable value for the weight is $\beta_{T_i} = 1$.

³For the sake of simplicity, in the following graphs and tables the name of the objects has been abbreviated in LP (Long Profile), SP (Small Profile) and CJ (Corner Joint).

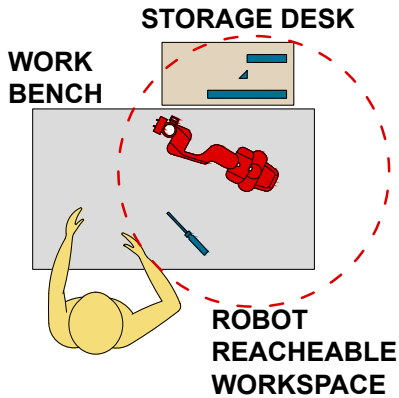


Fig. 3. Setup of the experiment. The worker was additionally equipped by an augmented reality interface, to ensure and raise worker awareness of the allocated tasks and the ongoing robot actions.

a human worker and a collaborative Franka Emika Panda equipped with just the standard gripper. This solution reflects the lean manufacturing principles, exploiting immediately the available resources instead of developing highly customised tools, that might not be convenient in a small production batch. To improve the synchronisation and coordination between the agents, the participants were provided of a pair of Microsoft HoloLens, a mixed reality smartglasses technology. The device features, among other sensors, an inertial measurement unit (IMU), a depth camera, a video camera, and microphones, enabling a natural way to interact with the environment, with holographic contents and with the surrounding information. The embedded gesture capture system and the voice commands of the AR device were exploited to promote an active collaboration between the agents: the device allowed to display the assembly status on the viewer and trigger particular moments through the gesture capture system.

The experimental setup is displayed in Fig. 3. The cobot was placed on a work bench where the assembly task had to be performed, while the components of the structure were placed on a storage desk. The Allen key, screws, and nuts were already available on the work bench.

The determined sequence of actions required to achieve the assembly task, independent of the agent skills, was⁴:

- 1) Pick the small aluminium profile from the storage desk;
- 2) Place the small aluminium profile in the work bench;
- 3) Pick the corner joint from the storage desk;
- 4) Place the corner joint in the assembly work bench;
- 5) Insert a nut in the small profile and attach it to the corner joints, aligning the screw with the nut, and fasten it with an Allen key;
- 6) Pick the long aluminium profile from the storage desk;
- 7) Place the long aluminium profile in the assembly area;
- 8) Insert a nut in the big profile and attach it to the other side of the corner joint, aligning the screw with the nut, and fasten it with the Allen key.

In agreement with the taxonomy of actions of Tab. I, we selected the skills required to achieve the task. In the first three columns of Tab. II, the required skills and their feasibility are

⁴The order of the actions is not unique.

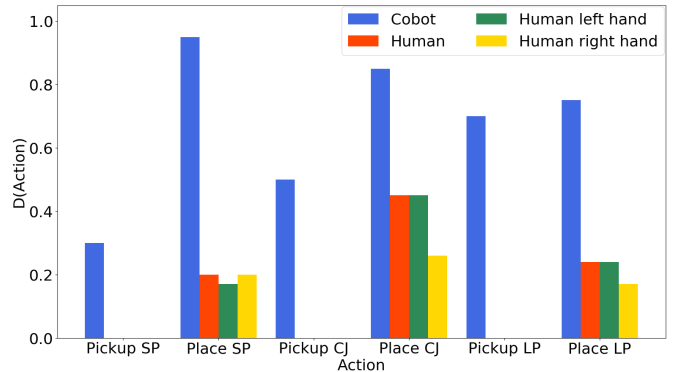


Fig. 4. Reachability values computed with Eq. (2) for Pickup and Place skills.

listed for each agent. Because of the absence of any perception features and advanced tools for screwing, the actions Align and Tool Action were considered as not feasible for the cobot. All the other actions could be performed by both agents.

The cost function values and the role assignments in Table II, computed with a simplified version of the A* algorithm [17], show that cobot precision and power were exploited in the repetitive Pick & Place actions, cobot endurance to carry the weight of the objects. On the other hand, human hand-eye coordination and task understanding to align the screws with the nuts (see Fig. 7 for photos of the experiment). To evaluate the task assignment we computed the cost function with the proposed metrics, for each action and for each agent. For the sake of simplicity, all the cost function weights $\beta_{T_i}, \beta_{D_i}, \beta_{V_i}$ were set to 1. For the computation of the task complexity metric, just the agent set of skills is required (see Table II). For the computation of the agent dexterity metric, the algorithm requires as input, the agent URDF file, a SRDF (Semantic Robot Description Format) file for the kinematics chains, and a parameter file with the constraints for the inverse kinematics algorithm, as well as action-related parameters, like the position in space where the action has to be performed. In particular, Fig. 4 displays the value of the reachability index D for the actions of Pickup and Place of the small profile, the corner joint, and the long profile. The Pickup action, when performed by the human agent, has 0 reachability index because the storage desk was placed outside the worker workspace. For this reason, the most suitable agent to accomplish the Pickup action, and therefore the Place action, based on this metric, was the cobot.

On the other hand, for the agent effort metric, in addition to other action-related parameters such as the weight of the loads and the nominal execution time of each action, the agent joint capacity vector \mathcal{C} was taken into account. In the experiment we fixed the same value for all the joints of the same agent, $C_{cobot} = 100$ and $C_{human} = 50$, to reflect the fact that the cobot can produce higher torques than the human, with less fatigue. This is clear in Fig. 5. In the absence of a proper dynamic modelling of the human agent, we focused on the fatigue due to the presence of external loads. Because of that, the metric was computed only in the actions that involved loads carrying (e.g. Transport, Hold). To compute the fatigue values, the nominal time of execution of the tasks, imposed by the production requirements, was set to 20 s for

TABLE II
ALUMINIUM PROFILE ASSEMBLY TASK.

Action	Cobot	Human	Cobot Cost	Human Cost	Role Assignment
Pickup + Transport + Place SP	✓	✓	1.360	∞	Cobot
Pickup + Transport + Place CJ	✓	✓	0.870	∞	Cobot
Align CJ with SP	✗	✓	∞	1.708	Human
Tool Action (Screwing)	✗	✓	∞	0.799	Human
Pickup + Transport + Place LP	✓	✓	0.842	∞	Cobot
Align LP with CJ	✗	✓	∞	1.736	Human
Hold CJ	✓	✓	0.313	0.454	Cobot
Tool Action (Screwing)	✗	✓	∞	1.427	Human

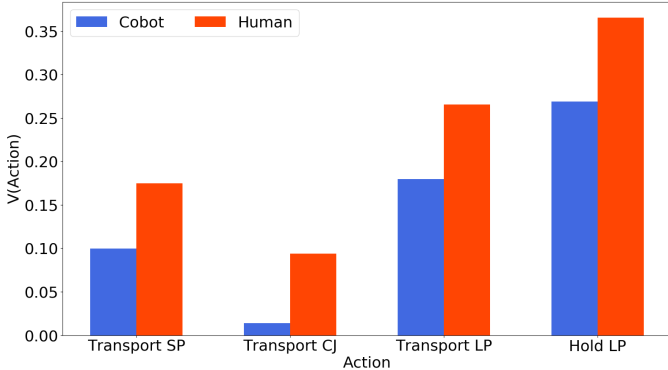


Fig. 5. Fatigue values computed with Eq. (6) for Transport and Hold skills.

the Transport action of the 3 objects and 60 s for the Hold action of the long profile. The fatigue indices were computed without considering the previous accumulated fatigue levels, but the general approach should consider the fatigue related to the previous achieved actions, as using the algorithm in [28]. Finally, to make human fatigue index comparable with the robot fatigue, the average in Eq. (6) considers just the fatigue of the arm holding the load.

To evaluate how the task selection affects the human's subjective measure of the teamed task, we asked 12 subjects of different sex, unaware of the scope of the experiment, to perform the described assembly task in different conditions. First, they performed it manually, and then using the collaborative setup and the AR device. Later, we asked them to compile a Likert scale-based questionnaire. The manual performance is useful since, in this way, the subjects could understand task requirements and the complexity.

The questionnaire included 9 statements. Q.1 The performed actions of the assembly were easy to perform; Q.2 It was physically tiresome to accomplish the task; Q.3 It was psychologically tiresome to accomplish the task; Q.4 Overall, I felt satisfied with the current task performance; Q.5 It was intuitive to understand the current action to perform; Q.6 I felt safe in performing the actions of the assembly; Q.7 It was easy to keep the focus on the task execution; Q.8 Overall, I think that using the current collaborative framework I could perform the same task for a longer duration and better quality time; and Q.9 Given the current task performance, I think that collaborative robots do not help to improve assembly tasks performance. The statements [5-9] are related just to the proposed approach. The possible answers ranged from strongly disagree to strongly agree, with an assigned score of -5 and +5, respectively. The results are displayed in Fig. 6.

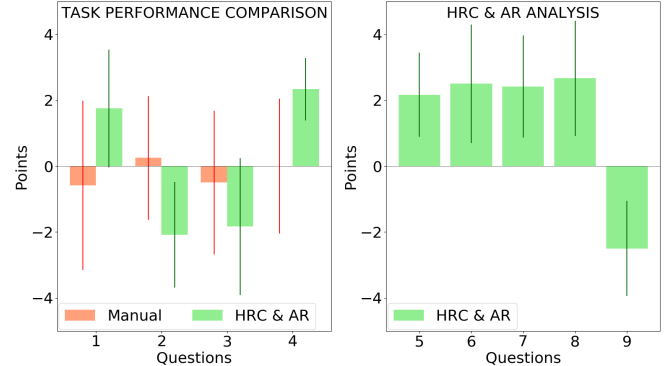


Fig. 6. Likert scale-based questionnaire scores for the assembly task. The first 4 statements were evaluated for both the manual and collaborative setup with the AR device, while from number 5 to 9 they are related just to the latter. The results show that the subjects felt satisfied with the proposed collaborative system, stating that it could actually help workers to improve performance and simplify task complexity.

The participants stated that the task execution with collaborative setup and the AR device involves less physical and psychological effort than the manual approach (Q.2-3). This perception led them to feel satisfied with the proposed collaborative system (Q.4) while in the manual setup the task was more difficult to perform (Q.1). Moreover, the subjects confirmed that with the help of an AR interface it was intuitive to understand the current action to be performed (Q.5) and to keep the focus on the task execution (Q.7). Therefore, they agreed with the facts that the worker could perform the same task for a long time with the current setup (Q.8), and that collaborative robots, could help workers to improve also task performance (Q.9). Noteworthy, the experiments were approved by the ethics committee Azienda Sanitaria Locale Genovese (ASL) N.3 (Protocollo IIT_HRII_001 (rif. interno:108/2018)).

VI. CONCLUSION

The proposed framework presented a modular capability-aware solution to the task allocation problem of collaborative assembly tasks in fast-reconfigurable industrial scenarios. The main advantage of the framework consists of the achievement of an optimal task assignment, according to the physical characteristics of the agents. We selected a role allocation behaviour based on ergonomic principles, but also other core components can be adopted. In this work, we considered a limited set of robot skills, since they were sufficient to complete repetitive and force-demanding tasks. Nevertheless, this set of actions can be adapted and extended to comply with requirements of the new assembly tasks. On the human

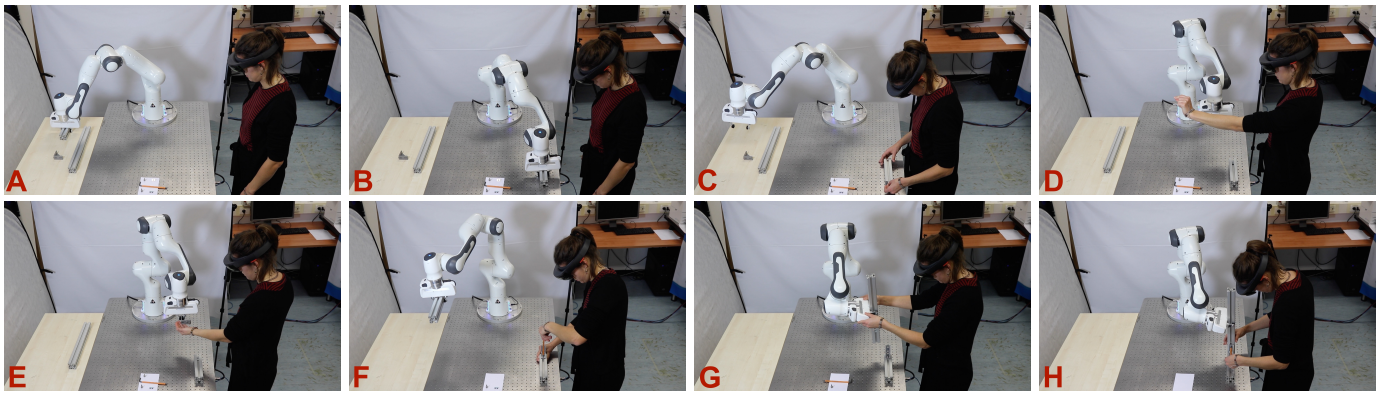


Fig. 7. Snapshots of the experiment. Fig. 8A-8B: Pickup, Transport and Place actions of the small profile. While the cobot picks the corner joint, the human worker can insert the nuts in the profile (Fig. 8C). The worker monitors the system through a pair of HoloLens, used to increase the situational awareness and to recognise human gestures (Fig. 8D). The gesture triggers the release of the corner joint from the gripper (Fig. 8E). The worker can start screwing and, at the same time, the cobot picks the long profile (Fig. 8F). The cobot then holds the profile in gravity compensation, so that the worker can easily align it to the mounted piece (Fig. 8G) and complete the task (Fig. 8H). The worker, successively, exploits the gesture to acknowledge the achievement of the assembly task. A Video of the experiment is available in the multimedia extension and in https://youtu.be/_Axmnu95TyQ.

side, we did not restrict the range of actions that can be executed by a human worker. Although age, gender, and particular disorders may limit human performance, they can be embedded in a function that discriminates if an action can be performed by an agent. Future work will focus on the application of the system in multi-agent real factory settings.

REFERENCES

- [1] J. P. Womack and D. T. Jones, *Lean Thinking : Banish Waste and Create Wealth in Your Corporation*. Simon & Schuster, 2003.
- [2] W. Kim, J. Lee, L. Peternel, N. Tsagarakis, and A. Ajoudani, "Anticipatory robot assistance for the prevention of human static joint overloading in human-robot collaboration," *IEEE Robotics and Automation Letters*, 2018.
- [3] P. Balatti, D. Kanoulas, N. G. Tsagarakis, and A. Ajoudani, "Towards robot interaction autonomy: Explore, identify, and interact," in *2019 IEEE International Conference on Robotics and Automation*, vol. published, 2019.
- [4] A. D. Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006.
- [5] S. Wrede, C. Emmerich, R. Grünberg, A. Nordmann, A. Swadzba, and J. Steil, "A user study on kinesthetic teaching of redundant robots in task and configuration space," *J. Hum.-Robot Interact.*, 2013.
- [6] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human-robot collaboration," *Autonomous Robots*, vol. 42, pp. 957–975, 2018.
- [7] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on humanrobot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248 – 266, 2018.
- [8] I. Maurtua, A. Iburguren, J. Kildal, L. Susperregi, and B. Sierra, "Humanrobot collaboration in industrial applications: Safety, interaction and trust," *International Journal of Advanced Robotic Systems*, 2017.
- [9] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, "Collaborative manufacturing with physical humanrobot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 40, 2016.
- [10] N. Jarrass, V. Sanguineti, and E. Burdet, "Slaves no longer: review on role assignment for human-robot joint motor action," *Adaptive Behavior*, vol. 22, no. 1, pp. 70–82, 2014.
- [11] A. Mörtl, M. Lawitzky, A. Kucukylmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The role of roles: Physical cooperation between humans and robots," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1656–1674, 2012.
- [12] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani, "Robot adaptation to human physical fatigue in human-robot co-manipulation," *Autonomous Robots*, Jun 2018.
- [13] P. Stone and M. Veloso, "Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork," *Artificial Intelligence*, vol. 110, no. 2, pp. 241 – 273, 1999.
- [14] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: a classification focused on coordination," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 5, pp. 2015–2028, Oct 2004.
- [15] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [16] L. S. H. de Mello and A. C. Sanderson, "And/or graph representation of assembly plans," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 188–199, April 1990.
- [17] L. Johansmeier 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, Jan 2017.
- [18] P. M. Fitts *et al.*, "Human engineering for an effective air-navigation and traffic-control system," Tech. Rep., 1951.
- [19] F. Ranz, V. Hummel, and W. Sihn, "Capability-based task allocation in human-robot collaboration," *Procedia Manufacturing*, vol. 9, pp. 182 – 189, 2017, 7th Conference on Learning Factories, CLF 2017.
- [20] 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.
- [21] 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, 2018.
- [22] J. O. Huckaby and H. I. Christensen, "A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics." in *AAAI*, 2012.
- [23] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007.
- [24] D. A. Winter, "Biomechanics of human movement," *Biomechanics*, pp. 65–83, 1979.
- [25] E. Mingo Hoffman, A. Rocchi, A. Laurenzi, and N. G. Tsagarakis, "Robot control for dummies: Insights and examples using opensot," in *2017 IEEE-RAS 17th Humanoids*, Nov 2017.
- [26] N. Vahrenkamp, H. Arnst, M. Wchter, D. Schiebener, P. Sotiropoulos, M. Kowalik, and T. Asfour, "Workspace analysis for planning human-robot interaction tasks," in *2016 IEEE-RAS 16th Humanoids*, 2016.
- [27] L. Ma, D. Chablat, F. Bennis, W. Zhang, and F. Guillaume, "A new muscle fatigue and recovery model and its ergonomics application in human simulation," *Virtual and Physical Prototyping*, 2010.
- [28] E. Lamon, L. Peternel, and A. Ajoudani, "Towards a prolonged productivity in industry 4.0: A framework for fatigue minimisation in robot-robot co-manipulation," in *2018 IEEE-RAS 18th Humanoids*, Nov 2018.
- [29] G. Michalos, S. Makris, L. Rentzos, and G. Chryssolouris, "Dynamic job rotation for workload balancing in human based assembly systems," *CIRP Journal of Manufacturing Science and Technology*, 2010.
- [30] A. Laurenzi, E. Mingo Hoffman, M. Parigi Polverini, and N. G. Tsagarakis, "Balancing control through post-optimization of contact forces," in *2018 IEEE-RAS 18th Humanoids*, Nov 2018.