An Optimal Allocation and Scheduling Method in Human-Multi-Robot Precision Agriculture Settings

Martina Lippi, Jorand Gallou, Andrea Gasparri, Alessandro Marino

***Abstract*— Employing teams of robots to offer services to human operators enables the latter to reduce their physical workload. In this paper, we focus on the problem of optimally allocating and scheduling the robot tasks in order to serve human operators. We formulate a Mixed-Integer Linear Programming problem that aims to minimize the human waiting time and the energy spent by the robots, while ensuring that any velocity constraints of the robots are fulfilled and the task ordering is correct. In addition, we propose an online re-allocation strategy that takes into account the possibility of changing human parameters over time. This strategy determines whether a new optimal solution must be computed. We validate the proposed framework in a simulated precision agriculture setting composed of two robots and four human operators for a harvesting application.**

I. INTRODUCTION

Human-robot teams have demonstrated great potential in various application domains [1], spanning from industrial to domestic environments and, recently, to agricultural contexts. This technology enables the integration of the cognitive abilities of humans with the physical capabilities of robots. In these collaborative teams, a key challenge to address is how to effectively allocate and schedule tasks, with the primary objective of accommodating human needs.

In this regard, the work in [2] categorizes tasks based on the number and type of agents involved (human or robotic). A task assignment flow chart is then proposed, which, however, does not consider any measure of optimality. In the case of minimization of optimality indices, Mixed Integer Linear Programming (MILP) problems are commonly formulated to minimize the overall execution time, i.e., the makespan. Among these, the work in [3] proposes a MILP problem aimed at minimizing the makespan for a team consisting of a human operator and multiple robots. A human capability dynamics is defined to adapt the solution in case of changes. However, in [3] the time horizon is divided into discrete intervals and constraints are defined for each, leading to a number of decision variables that dramatically increase as the time horizon increases. This is overcome, for instance, in [4] where additional features, such as the possibility for humans to supervise robots when they are unable to operate autonomously, and further cost indices based on the workload and quality measures are also included.

M. Lippi, J. Gallou and A. Gasparri are with Roma Tre University, Italy, {*martina.lippi,jorand.gallou*}*@uniroma3.it, gasparri@dia.uniroma3.it*.

A. Marino is with University of Cassino and Southern Lazio, Italy, *al.marino@unicas.it*.

This work has been supported by the European Commission under the grant agreement number 101016906 – Project CANOPIES and by the Regione Lazio within the Programs POR FESR Lazio 2014-2020, Avviso Pubblico "Progetti di Gruppi di Ricerca 2020," AGR-o-RAMA, under Agreement B85F21001360006.

The optimization of the workload in the MILP formulation is also considered in [5], where, however, tasks are divided into several layers based on precedence constraints, and the cycle time of these layers is minimized. The same layer subdivision is exploited in [6] where tasks are prioritized according to the task dependency order. A further MILP application can be found in [7] where tasks are assigned to the agents on the basis of a Hungarian algorithm and, then, are scheduled according to a MILP.

Alternative approaches based on constraint programming and genetic algorithms are investigated in [8] to solve a MILP problem minimizing the makespan. However, no dynamic behavior of the human operators is taken into account. A genetic algorithm is additionally adopted in [9] where a simulation tool is developed to model an assembly procedure. Based on this tool, a fitness function is defined to optimize for the task progress, waiting time, and traveled distance. Repetitive tasks are considered instead in [10] where search algorithms based on branch and bound optimization and evolutionary search are explored. Recently, a multi-agent deep reinforcement learning approach has been proposed in [11] for a collaborative assembly application. However, no variability in the human parameters is examined.

In this work, we consider a human-multi-robot scenario where mobile robots perform service tasks for human agents. Inspired by the European project CANOPIES, we consider a precision agriculture setting where, as an example case study, box filling operations during table grape harvesting season are carried out. We formulate an optimization problem based on MILP to minimize a combination of human waiting time and robot energy and determine the allocation of the service tasks to the robots and their scheduling. Differently from most works cited above, we additionally consider that the velocity of the robotic agents represents an output of the optimization problem, which can be modulated to minimize the cost function. Furthermore, to tackle possible variability in the human parameters, we design an online checking strategy to establish if a new plan is needed. Compared to our previous work [4], we focus on a notably different scenario where robots provide services to human operators. Thus, we formulate an alternative optimization problem that aligns with the agriculture setting considering, also time-varying robot velocities. To the best of our knowledge, this is also the first contribution in the literature dealing with a human-multi-robot task allocation and scheduling problem in a precision agriculture setup. The proposed approach can be adapted to any service task and to different settings, such as assembly operations in industrial scenarios or assistive applications in domestic environments.

II. Problem setting

As introduced above, we consider a team of mobile robots and human operators working in an agricultural field, where a depositing station is available, as depicted in Figure 1. More specifically, every human performs operations in the field and, at the completion of each operation, a robot service task is necessary to enable the continuation of the person's activities. As an example case study, we assume that the humans are filling boxes with grape bunches during the harvesting season; once a box is filled, a robot is required to execute a service task that involves reaching the human operator, replacing the full box with an empty one, and transporting the full box to a depositing station where a new empty box is picked up. Therefore, our objective is to determine the allocation of the service tasks to the robots, i.e., which robot performs each task, as well as their scheduling, i.e., the start and end time of each task, while taking into account possible constraints on the robots and variations in the human behavior. In the remainder of the section, we introduce the notation to formulate the problem and state it formally.

In general, let $\mathcal{S} = \{s_1, s_2, .., s_n\}$ be a set, in the following we use the notation $i \in \mathcal{S}$ to denote the i-th element of the set and $|\mathcal{S}|$ for its cardinality. Similarly, in case two indices are utilized for the elements of the set, i.e., $\mathcal{S} = \{s_{1,2}, s_{1,2}, .., s_{n,m}\}$, we use the notation $(i, j) \in \mathcal{S}$ to denote the element $s_{i,j}$.

A. Agents and tasks

We denote by $\mathcal{R}$ the set of n mobile service robots, i.e., $|\mathcal{R}| = n$, and $\mathcal{H}$ the set of u human operators, i.e., $|\mathcal{H}| = u$. Minimum and maximum cruising velocities to navigate the field are considered for each robot r and denoted as $v_{\min,r}$ and $v_{\max,r}$, respectively. Moreover, we assume that an assigned position at the depositing station is defined. We refer to the latter as base position of the robot and denote it as b_r. For each human h, we denote his/her position in the field as x_h and define the length of the path that robot r has to follow to reach the human from its base position as $l_{r,h}$. Note that this length coincides with the one of the reverse travel, from the human operator to the robot base position.

As previously introduced, each human h performs a set of q ordered operations $\mathcal{O}_h = \{\tau^o_{h,1}, ..., \tau^o_{h,q}\}$, meaning that the start time of $\tau^o_{h,i}$ follows the end time of $\tau^o_{h,i-1}$, $\forall i \in \{2, ..., q\}$. Let $\underline{o}_{h,i}$ and $\overline{o}_{h,i}$ be the start and end time of the human operation $\tau^o_{h,i}$, respectively, it holds $\overline{o}_{h,i-1} \geq \underline{o}_{h,i}$. The overall set of human operations is given by $\mathcal{O} = \mathcal{O}_1 \cup ... \cup \mathcal{O}_u$. In addition, we define the time needed by the human h to perform a single operation as δ^o_h. For simplicity of notation, we assume that each human operator carries out q operations and all operations of a human operator h have the same duration δ^o_h. However, it is straightforward to include a different number of operations q_h for each human h and different durations. In general, the human parameters can vary over time, i.e., he/she might change his/her position (i.e., $l_{r,h}$ might change) or operation duration δ^o_h. For instance, when an operator's fatigue increases, his/her operating times will probably also increase.

For each human operation $(h, i) \in \mathcal{O}$, we define a respective service activity that a robot has to carry out. This is composed

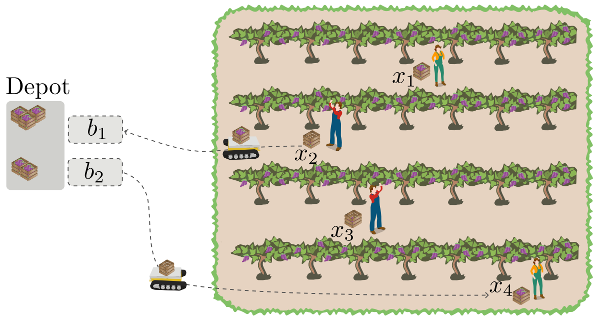

Fig. 1: Illustration of the field with two robots and four humans.

of two tasks: *i)* a *picking* task $\tau^p_{h,i}$, where a robot navigates the field to reach the human h and assists him/her by picking up any load provided by the human for the operation $\tau^o_{h,i}$, and *ii)* a subsequent *depositing* task $\tau^d_{h,i}$, in which the robot reaches a depositing station and releases any load provided by the human. We denote the times for the robot r to pick up a load at a human position and release it at the depositing station as δ^p_r and as δ^d_r, respectively. In our case study, during the picking task, the robot approaches the human and performs a box replacement, while during the depositing task, the same robot reaches the depositing station and releases the full box to grab an empty one. We define the set $\mathcal{P}$ collecting the picking tasks, i.e., $\mathcal{P} = \mathcal{P}_1 \cup ... \cup \mathcal{P}_u$ with $\mathcal{P}_h = \{\tau^p_{h,1}, ..., \tau^p_{h,q}\}$, and the set $\mathcal{D}$ collecting the respective depositing tasks, i.e., $\mathcal{D} = \mathcal{D}_1 \cup ... \cup \mathcal{D}_u$ with $\mathcal{D}_h = \{\tau^d_{h,1}, ..., \tau^d_{h,q}\}$. We denote the start and end time of each picking task $\tau^p_{h,i}$ as $\underline{p}_{h,i}$ and $\overline{p}_{h,i}$, respectively, and the start and end time of each depositing task $\tau^d_{h,i}$ as $\underline{d}_{h,i}$ and $\overline{d}_{h,i}$, respectively. Based on the above sets, we introduce the binary decision variable $P_{h,i,r} \in \{0, 1\}$, that is equal to 1 when the robot r performs the picking task $(h, i) \in \mathcal{P}$, and is 0 otherwise, and the binary decision variable $D_{h,i,r} \in \{0, 1\}$, that is equal to 1 when the robot r performs the depositing task $(h, i) \in \mathcal{D}$. Note that the cardinality of the sets $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{D}$ is identical. With an abuse of notation, we use the same indices for these sets, i.e., the index $(h, i) \in \mathcal{O}$ is used to refer to the human operation $\tau^o_{h,i}$ as well as the respective picking $\tau^p_{h,i}$ and depositing tasks $\tau^d_{h,i}$.

B. Cost indices and problem formulation

We define the waiting time $w_{h,i}$ for the human operation (h, i) as the difference between the starting time of the operation and the end of the previous picking task by a robot, i.e.,

$$w_{h,i} = \underline{o}_{h,i} - \overline{p}_{h,i-1} \quad \text{with } i > 1. \tag{1}$$

Indeed, a human cannot start an operation if the previous picking has not been performed by a robot, e.g., in our setting, a robot needs to replace a full box with an empty one in order for the human to start a new box filling operation. In the case of first operation of a human operator, i.e., $i = 1$, we consider $w_{h,1} = \underline{o}_{h,1}$, $\forall h \in \mathcal{H}$.

We introduce the average traveling velocity $v^p_{h,i}$ of the picking task (h, i) as the ratio between the length of the path followed by the robot and the respective travel time, i.e.,

$$v^p_{h,i} = \frac{\sum_{r\in\mathcal{R}} P_{h,i,r} l_{r,h}}{\left(\overline{p}_{h,i} - \sum_{r\in\mathcal{R}} P_{h,i,r}\delta^p_r - \underline{p}_{h,i}\right)}. \tag{2}$$

Note that the term $\sum_{r\in\mathcal{R}} P_{h,i,r}\delta^p_r$ provides the time spent in assisting the human and picking up the load for the operation (h,i), which is excluded for the calculation of the robot's actual travel time. Based on the above, we define an energy-like term $e^p_{h,i}$ for the picking task (h,i) which increases with higher velocities of the robot performing the task [12], i.e.,

$$e^p_{h,i} = k_e \left(1/v_{\min} - 1/v^p_{h,i}\right), \tag{3}$$

with k_e a positive constant, and $v_{\min}$ the minimum robot velocity of the team, that is $v_{\min} = \min_{r\in\mathcal{R}} v_{\min,r}$. Therefore, the lower the robot velocity during the picking traveling, the lower the energy cost $e^p_{h,i}$, which reaches 0 when $v^p_{h,i} = v_{\min}$.

Similarly, we define an energy-like term $e^d_{h,i}$ for the depositing task (h,i) defined as

$$e^d_{h,i} = k_e \left(1/v_{\min} - 1/v^d_{h,i}\right), \tag{4}$$

where it holds

$$v^d_{h,i} = \frac{\sum_{r\in\mathcal{R}} D_{h,i,r} l_{r,h}}{\overline{d}_{h,i} - \sum_{r\in\mathcal{R}} D_{h,i,r}\delta^d_r - \underline{d}_{h,i}}. \tag{5}$$

The problem addressed in this work is formalized as follows.

Problem 1. Consider a system composed of u humans, performing agricultural operations $\mathcal{O}$, and n mobile robots, performing picking $\mathcal{P}$ and depositing tasks $\mathcal{D}$ to assist the humans. The objective is to determine the allocation variables $P_{h,i,r}$ and $D_{h,i,r}$, as well as the tasks start and end times of the picking and depositing tasks, $\underline{p}_{h,i}$, $\overline{p}_{h,i}$, $\underline{d}_{h,i}$, $\overline{d}_{h,i}$, in such a way to minimize a combination of the human waiting time, $w_{h,i}$, and robots energy, $e^p_{h,i}$, $e^d_{h,i}$, $\forall (h,i)\in\mathcal{O}, r\in\mathcal{R}$, while fulfilling the robots' velocity constraints and the correct order of the tasks execution. Furthermore, the solution must adapt to any changing human parameter.

To solve the above problem, we design: *i)* an optimal allocation and scheduling module and *ii)* a re-allocation strategy that updates, if needed, the allocation and scheduling to account for the variation of time-varying human parameters.

The first module is based on the formalization of a MILP problem. Specifically, its input is given by the robots' velocity bounds, $v_{\min,r}, v_{\max,r}$, the lengths $l_{r,h}$ of the paths from the base positions of the robots to the humans, the durations δ^p_r and δ^d_r of the load picking and releasing activities, respectively, as well as the duration of the human operations δ^o_h, $\forall h\in\mathcal{H}$, $r\in\mathcal{R}$. Based on these parameters, the module determines the optimal allocation variables and the start and end times of the picking and depositing tasks. It also provides the start and end times of the human operations $\underline{o}_{h,i}$, $\overline{o}_{h,i}$ $\forall (h,i)\in\mathcal{O}$ based on the picking times. We consider that each human starts a box filling operation as soon as an empty box is available to him/her. We refer to the output variables of the MILP problem as decision variables. Note that in the MILP problem we assume constant human parameters, while their variations are tackled by the second module.

For the second module, we assume that a human awareness system is available which estimates the position x_h of each human in the field and the duration δ^o_h of the operations. Such a system is out of the scope of this paper but many works can be employed such as [13], [14]. Based on these estimations, we evaluate the variation with respect to the parameters used at allocation time. We also examine if they lead to any violation of the constraints for task execution. In this case or if a significant variation is recorded, re-allocation and re-scheduling are performed to generate a new optimal task allocation plan.

III. Optimal allocation and scheduling

In the following, we describe the cost function and constraints involved in the MILP formulation. As far as the cost function is concerned and according to Problem 1, the following is defined

$$c = \sum_{(h,i)\in\mathcal{O}} \alpha\, w_{h,i} + \beta\, e^p_{h,i} + \gamma\, e^d_{h,i}, \tag{6}$$

where α, β, γ are positive weights. The rationale behind this cost function is that, for each human operation (h,i) we aim to minimize the total human waiting time as well as the overall energy spent by robots to serve human operations, given by the energy for the picking phase, $e^p_{h,i}$, and the energy for the depositing phase, $e^d_{h,i}$. It is worth noticing that in general, in order to reduce the humans' waiting time, the robot velocity has to be increased, leading to higher energy consumption; therefore, the objective terms related to the human waiting time and the robots' energy are antagonistic and their priority is balanced by α, β, γ weights.

With regard to the constraints, the following are defined.

1) Picking tasks assignment:

$$\sum_{r\in\mathcal{R}} P_{h,i,r} = 1, \quad \forall (h,i)\in\mathcal{P}. \tag{7}$$

The above constraint ensures that each human operation (h,i) is served by only one robot, i.e., by considering all robots in $\mathcal{R}$, only for one of them it must hold $P_{h,i,r} = 1$.

2) Depositing tasks assignment:

$$D_{h,i,r} = P_{h,i,r}, \quad \forall (i,r)\in\mathcal{P}, r\in\mathcal{R}. \tag{8}$$

This equality enforces that, if a robot performs a picking task, it also performs the associated depositing task to release any load at the depositing station. Therefore, in case the robot r executes the picking task (h,i), i.e., $P_{h,i,r} = 1$, it must hold $D_{h,i,r} = 1$; in the same way, if the robot is not assigned to the picking task, i.e., $P_{h,i,r} = 0$, it will not perform any corresponding depositing task, i.e., $D_{h,i,r} = 0$.

3) Picking tasks duration:

$$\overline{p}_{h,i} - \underline{p}_{h,i} \geq P_{h,i,r}\frac{l_{r,h}}{v_{\max,r}} + P_{h,i,r}\delta^p_r, \tag{9a}$$

$$\overline{p}_{h,i} - \underline{p}_{h,i} \leq P_{h,i,r}\frac{l_{r,h}}{v_{\min,r}} + P_{h,i,r}\delta^p_r, \tag{9b}$$

$\forall (h,i)\in\mathcal{P}, r\in\mathcal{R}$. These inequalities establish the duration of each picking task. In detail, the quantities $l_{r,h}/v_{\max,r}$ and

$l_{r,h}/v_{\min,r}$ represent the minimum and maximum travel time for the robot r to reach the human h, respectively. Therefore, the inequality (9a) ensures that the duration of the picking task (h,i), i.e., $\overline{p}_{h,i} - \underline{p}_{h,i}$, is greater than the minimum travel time required by the robot r assigned to the task, i.e., such that $P_{h,i,r} = 1$, plus the load picking time δ_r^p. Similarly, the inequality (9b) ensures that the duration of the picking task (h,i) is lower than the maximum travel time required by the robot assigned to the task plus the load picking time.

4) Depositing tasks duration:

$$\overline{d}_{h,i} - \underline{d}_{h,i} \geq D_{h,i,r}\frac{l_{r,h}}{v_{\max,r}} + D_{h,i,r}\delta_r^d, \tag{10a}$$

$$\overline{d}_{h,i} - \underline{d}_{h,i} \leq D_{h,i,r}\frac{l_{r,h}}{v_{\min,r}} + D_{h,i,r}\delta_r^d, \tag{10b}$$

$\forall (h,i) \in \mathcal{D}, r \in \mathcal{R}$. By following a similar approach to constraints (9a)-(9b), the above inequalities define the duration of the depositing tasks: the constraint in (10a) ensures that the duration of each depositing task (h,i) is greater than the minimum travel time required by the robot r assigned to the task, i.e., such that $D_{h,i,r} = 1$, plus the time to release the load δ_r^d; the inequality (10b) enforces the duration of the depositing task (h,i) to be lower than the maximum travel time required by the assigned robot plus the time to release the load.

5) Depositing tasks start time:

$$\underline{d}_{h,i} = \overline{p}_{h,i}, \forall (h,i) \in \mathcal{P}. \tag{11}$$

This equality, together with (8), ensures that after a picking task the respective depositing is made, i.e., the start time of the depositing task (h,i) is equal to the end time of the picking task (h,i).

6) Picking tasks end time:

$$\overline{p}_{h,i} \geq \overline{o}_{h,i} + \textstyle\sum_{r\in\mathcal{R}} P_{h,i,r}\delta_r^p, \quad \forall (h,i) \in \mathcal{P}. \tag{12}$$

The above establishes when the robot has to arrive at the human position for the picking tasks. More specifically, it requires that the end time of the robot travel for the picking task (h,i), that is given by the $\overline{p}_{h,i} - \sum_{r\in\mathcal{R}} P_{h,i,r}\delta_r^p$, is greater than or equal to the end time $\overline{o}_{h,i}$ of the human operation (h,i). In this way, no robot will wait in proximity to humans, but they will approach the operators only when necessary to perform the service and pick up a load.

7) Human sequence:

$$\overline{o}_{h,i} = \underline{o}_{h,i} + \delta_h^o, \tag{13a}$$

$$\underline{o}_{h,i} \geq \overline{p}_{h,i-1}, \tag{13b}$$

$\forall (h,i) \in \mathcal{O}$. The above constraints impose the correct sequence of human operations. Firstly, (13a) imposes that the final time of the operation (h,i), i.e., $\overline{o}_{h,i}$, is equal to the start time of the same operation, i.e., $\underline{o}_{h,i}$, plus the duration of the human operation, i.e., δ_h^o. Secondly, (13b) requires that, in order for the human h to start the operation i, the picking of the previous operation $i-1$ must be completed, i.e., the start time of the operation (h,i) must be greater than or equal to the end time of the picking task $(h,i-1)$. For instance, in our scenario, each human operator needs the full box to be replaced with an empty one in order to start the next harvesting operation.

8) Execution of one task at a time:

$$\underline{p}_{s,k} - \overline{p}_{h,i} \geq -M(2 - P_{h,i,r} - P_{s,k,r}) - M(1 - U_{h,i,s,k,r}), \tag{14a}$$

$$\overline{p}_{s,k} - \underline{p}_{h,i} \geq -M(2 - P_{h,i,r} - P_{s,k,r}) - M\, U_{h,i,s,k,r}, \tag{14b}$$

$$\underline{d}_{s,k} - \overline{d}_{h,i} \geq -M(2 - D_{h,i,r} - D_{s,k,r}) - M(1 - Q_{h,i,s,k,r}), \tag{14c}$$

$$\overline{d}_{s,k} - \underline{d}_{h,i} \geq -M(2 - D_{h,i,r} - D_{s,k,r}) - M\, Q_{h,i,s,k,r}, \tag{14d}$$

$$\underline{d}_{s,k} - \overline{p}_{h,i} \geq -M(2 - P_{h,i,r} - D_{s,k,r}) - M(1 - V_{h,i,s,k,r}), \tag{14e}$$

$$\overline{d}_{s,k} - \underline{p}_{h,i} \geq -M(2 - P_{h,i,r} - D_{s,k,r}) - M\, V_{h,i,s,k,r}, \tag{14f}$$

$\forall (h,i),(s,k) \in \mathcal{O}$, $r \in \mathcal{R}$, where M is an arbitrarily large positive constant, and $U_{h,i,s,k,r}$, $Q_{h,i,s,k,r}$, $V_{h,i,s,k,r} \in \{0,1\}$ represent auxiliary binary decision variables. These constraints specify that each robot can perform only one task at a time, i.e., at each time instant, a robot can be either idle or involved in a picking or a depositing task. In detail, let us consider the case where a robot r is assigned to both the picking tasks (h,i) and (s,k), i.e., $P_{h,i,r} = P_{s,k,r} = 1$. Then, (14a) and (14b) state that it must either hold that (s,k) is executed after (h,i) is completed, i.e., $\underline{p}_{s,k} \geq \overline{p}_{h,i}$ with $U_{h,i,s,k,r} = 1$, or the opposite, i.e., $\underline{p}_{h,i} \geq \overline{p}_{s,k}$ with $U_{h,i,s,k,r} = 0$, but it is not possible that the two tasks are executed simultaneously. Note that if the robot r is not involved in both the picking tasks, i.e., it holds that at least one of the allocation variables is zero, $P_{h,i,r} = 0$ or $P_{s,k,r} = 0$, then no constraints are enforced by (14a) and (14b) with respect to the relative start and end times. Similarly, (14c) and (14d) state that, if a robot performs two depositing tasks (h,i) and (s,k), then it must either hold that (s,k) is executed after (h,i) is completed, or the opposite. Finally, (14e) and (14f) state that also in case a robot performs a picking task and a depositing one, they cannot be simultaneously executed by the same robot. Hence, the constraints in (14) overall guarantee that each robot only executes at most one task at a time.

In summary, the decision variables are obtained by solving the following MILP problem:

$$\min \sum_{(h,i)\in\mathcal{O}} \alpha\, w_{h,i} + \beta\, e_{h,i}^p + \gamma\, e_{h,i}^d \tag{15a}$$

$$\text{s.t.} \quad \text{Constraints (7)-(14)} \tag{15b}$$

IV. Online re-allocation and -scheduling

Since the human parameters can generally vary over time, online re-allocation and re-scheduling might be needed to adapt the solution. Specifically, for each human operator h, a variation of the respective duration of the operation, δ_h^o, and his/her location in the field, x_h, can occur. Clearly, the latter case leads to a change in the lengths $l_{r,h}$ of the paths from the robot base positions to the human h, $\forall r \in \mathcal{R}$. Let t_s be the time when the solution in use is computed, i.e., the decision variables are set by solving (15), and t_c be the time when a change in the human parameters is recorded. In the following, we denote with $y(t)$ the value of variable y computed at time t, e.g., $\delta_h^o(t_c)$ represents the value of δ_h^o computed at time t_c.

We disregard the solution, i.e., the decision variables, defined at time t_s when one of the following conditions occur:

1) Any of the problem constraints is violated.
2) The change in the human parameters is higher than a certain threshold with respect to the values used at time t_s.

Obviously, these conditions are only verified for operations or tasks that are not completed by time t_s. To assess if condition 1) is verified, we consider that the updated final time of each human operation (h, i) is given by

$$\overline{o}_{h,i}(t_c) = \underline{o}_{h,i}(t_s) + \delta_h^o(t_c), \quad \forall h \in \mathcal{H}$$

where $\underline{o}_{h,i}(t_s)$ is the start time of the operation (h, i) determined at time t_s and $\overline{o}_{h,i}(t_c)$ is the final time of the operation (h, i) updated according to the variation recorded at time t_c. Next, we check if any of the constraints (9), (10), and (12) is violated by using the distances $l_{r,h}(t_c)$ and the final times $\overline{o}_{h,i}(t_c)$, i.e., we evaluate the following inequalities

$$\begin{aligned}
&\overline{p}_{h,i}(t_s) - \underline{p}_{h,i}(t_s) \geq P_{h,i,r}(t_s)\frac{l_{r,h}(t_c)}{v_{\max,r}} + P_{h,i,r}(t_s)\delta_r^s,\\
&\overline{p}_{h,i}(t_s) - \underline{p}_{h,i}(t_s) \leq P_{h,i,r}(t_s)\frac{l_{r,h}(t_c)}{v_{\min,r}} + P_{h,i,r}\delta_r^s,\\
&\overline{d}_{h,i}(t_s) - \underline{d}_{h,i}(t_s) \geq D_{h,i,r}(t_s)\frac{l_{r,h}(t_c)}{v_{\max,r}} + D_{h,i,r}(t_s)\delta_r^d,\\
&\overline{d}_{h,i}(t_s) - \underline{d}_{h,i}(t_s) \leq D_{h,i,r}(t_s)\frac{l_{r,h}(t_c)}{v_{\min,r}} + D_{h,i,r}(t_s)\delta_r^d,\\
&\overline{p}_{h,i}(t_s) \geq \overline{o}_{h,i}(t_c) + \textstyle\sum_{r\in\mathcal{R}} P_{h,i,r}(t_s)\delta_r^p,
\end{aligned} \tag{16}$$

where the reference time of each variable has been made explicit for the sake of clarity. If any constraint is violated, the decision variables must be re-computed according to the parameters at time t_c.

Regarding condition 2), we evaluate whether it holds

$$|\delta_h^o(t_c) - \delta_h^o(t_s)/\delta_h^o(t_s)| \geq \omega_\delta, \tag{17}$$

with ω_δ a positive constant, or

$$|l_{r,h}(t_c) - l_{r,h}(t_s)/l_{r,h}(t_s)| \geq \omega_l, \tag{18}$$

with ω_l a positive constant, for any $h \in \mathcal{H}, r \in \mathcal{R}$. The rationale behind the above inequalities is that we evaluate the relative variation of the human parameters with respect to the ones at time t_s. If this variation, for the path length or the human duration, exceeds certain thresholds, the optimality of the solution in use may be compromised. Hence, a re-computation of the decision variables is made.

When a re-computation of the decision variables is required, we solve the optimization problem in (15) by using the updated parameters, i.e., we use the values $\delta_h^o(t_c)$ and $l_{r,h}(t_c)$ in place of δ_h^o and $l_{r,h}$, respectively. The tasks in execution at time t_c remain allocated to the same robots as in the previous solution.

V. VALIDATION RESULTS

To validate the proposed framework, we considered a simulated table grape harvesting application, as depicted in Figure 1, with a team composed of $n = 2$ mobile robots and $u = 4$ human operators performing grape box filling tasks. We considered five vine rows of 30m long in the field and planting pattern of 3m × 3m. The depositing station was centered at $[5, 1]$ m. Regarding the humans, their initial positions were randomly chosen within the vineyard rows. The times δ_h^o to fill boxes with grape bunches were obtained with a uniform distribution in the interval $[200, 400]$ s, while we considered that each human performs $q = 3$ box filling operations. Regarding the robots, their base positions were set at $[4, 2]$ m and $[6, 2]$ m. The times for the box replacements during picking tasks δ_r^p and for box releases during depositing tasks δ_r^d were set to 10 s and 5 s, respectively. Minimum $v_{\min,r}$ and maximum $v_{\max,r}$ velocities were set to 0.2 m/s and 0.6 m/s, respectively, $\forall r \in \mathcal{R}$. The paths, along with their lengths, from base positions to human locations were computed according to a RTT planner. Regarding the cost function in (6), we normalized the terms in it and selected $\alpha = \beta = \gamma = 0.5$, i.e., same weight for all the contributions. In addition, we used the thresholds $\omega_\delta = 0.2$ and $\omega_l = 0.1$. We implemented the simulation software with MATLAB interfaced with Gurobi solver. As mentioned in Section II-B, we assumed the availability of a human awareness method to estimate the changes in the human parameters. A video showing the simulation results can be found at the link[1].

Figure 2-left presents the initial allocation and scheduling obtained by solving the problem. The first two columns are associated with the robots and the last four to the humans, while the y axis represents the time evolution. Each colored segment is associated with a task (thinner for the depositing tasks). Specifically, the same color is used to denote the generic human operation $\tau_{h,i}^o$ and the respective picking $\tau_{h,i}^p$ and depositing tasks $\tau_{h,i}^d$. The figure shows that all the constraints for the correct execution of the tasks are fulfilled: the human operators wait for the robots to pick up the previous boxes to start a new box filling operation, and the robots return to the depositing station after each picking. Furthermore, the robot velocities respect their allowed range. For instance, by considering the first task of robot 2, the picking of the first box filled by human 3, i.e., $P_{3,1,2} = 1$, is assigned. The completion time of this task is 294 s, which is given by the finishing time of human filling task $\overline{o}_{3,1}$ plus the assistance time of 10 s to exchange the human full box with an empty one. Then, the robot has to reach its base station to release the full box and complete the depositing task at time 324 s. Afterwards, the robot has to start serving the human 2 and so on. Regarding the humans, each operation is followed by a minimum idle time of 10 s to enable the robot to provide assistance and exchange the box. Additional waiting times equal to 56 s and 90 s are recorded for the human operations $(2, 1)$ and $(4, 1)$, respectively, as no robots are available to immediately pick the filled boxes.

We simulated a change in the human parameters at time $t_{c,1} = 284$ s. Specifically, we considered that human 3 moved

[1]https://youtu.be/ZMBGHU7clTY

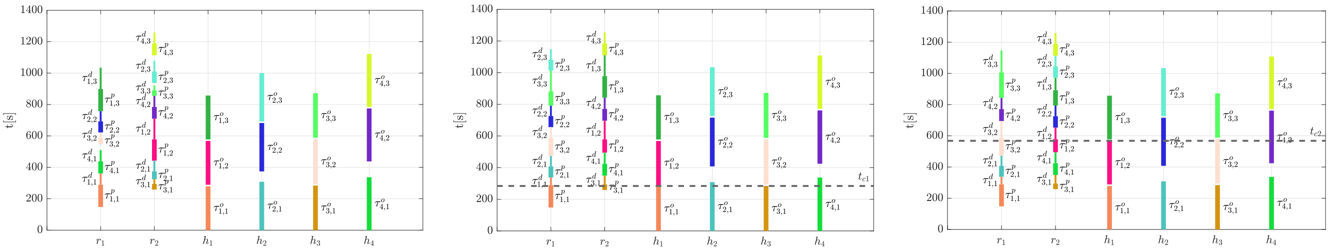

Fig. 2: Left: initial plan of the tasks on the left; middle: second plan updated at time $t_{c,1}$; right: third plan updated at time $t_{c,2}$.

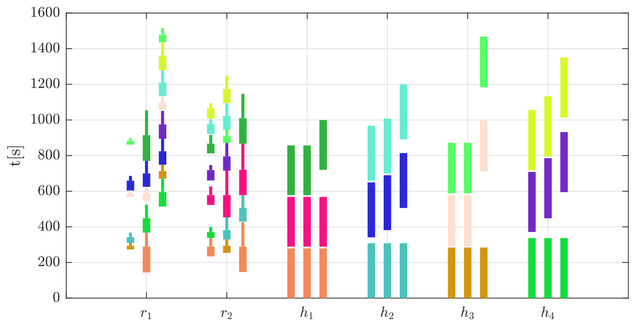

Fig. 3: Comparison of the plans obtained with different weights (columns for each agent) in the cost function.

by 25 m reaching the end of the next row in the field. This led to the violation of constraint (9). Therefore, a reallocation was performed according to Sec. IV leading to the updated plan in Figure 2-middle, where the grey dotted line highlights the time $t_{c,1}$. No allocation changes were made for the tasks started before the reallocation time $t_{c,1}$, while the remaining ones were updated. For instance, $\tau^p_{4,1}$ was assigned to r_1 at the initial time and was then allocated to r_2 after reallocation. We then simulated a further change at time $t_{c,2} = 568$ s in the position of human 1 by 5 m. This change did not lead to any violation of the constraints, but fulfilled the condition in (18). A new plan was then computed as shown in Figure 2-right. By doing so, an overall cost equal to 0.3476 was achieved, compared to a cost equal to 0.3148 of the previous solution.

Finally, we compared the results obtained using different weights in the cost function to show their influence on the solution. Figure 3 reports the plans of each agent obtained with three sets of weights: *i)* $\alpha = 1$ and $\beta = \gamma = 0$ (first column of each agent), *ii)* $\alpha = \beta = \gamma = 0.5$ (second column), *iii)* $\alpha = 0$ and $\beta = \gamma = 1$ (third column). We can observe that in case *i)*, low human waiting times are recorded for the humans and the robots' tasks have short durations (high velocity). In case *ii)*, higher human waiting times can be observed compared to the first case due to slowest robots. Moreover, different allocations of some tasks are obtained. In case *iii)*, the durations of the robots' tasks increase significantly compared to case *i)* leading to higher human waiting times and overall makespan.

VI. Conclusion

In this work, we proposed a framework for allocating and scheduling tasks in a human-multi-robot scenario. Given the set of human operations, a MILP problem was defined to establish which robot and when it should provide service for each operation. We optimized the waiting time of the humans and the energy consumption of the robots, while ensuring compliancy with the robot constraints and the correct task ordering. To take into account the human dynamic behavior, we designed an online strategy which defines if a re-allocation is necessary based on the changes in the human parameters. We validated the approach in an agricultural table grape harvesting application, where humans are responsible for filling boxes with grape bunches, while robots replace the full boxes with empty ones and transport the full boxes to a designated depositing station. Future work aims to define adaptive weights in the cost function and validate the approach on a real-world setup.

References

[1] T. B. Sheridan, "Human–robot interaction: status and challenges," *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[2] G. Bruno and D. Antonelli, "Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells," *Int. J. Adv. Manuf. Tech.*, vol. 98, pp. 2415–2427, 2018.

[3] 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 Int. Conf. Robot. Autom.*, 2020, pp. 3860–3866.

[4] M. Lippi and A. Marino, "A mixed-integer linear programming formulation for human multi-robot task allocation," in *IEEE Int. Conf. Robot and Human Interactive Comm.*, 2021, pp. 1017–1023.

[5] A. Pupa and C. Secchi, "A safety-aware architecture for task scheduling and execution for human-robot collaboration," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2021, pp. 1895–1902.

[6] S. Alirezazadeh and L. A. Alexandre, "Dynamic task scheduling for human-robot collaboration," *IEEE Robot. Autom. Letters*, vol. 7, no. 4, pp. 8699–8704, 2022.

[7] 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 Int. Conf. Robot and Human Interactive Comm.*, 2022, pp. 1271–1276.

[8] C. Ferreira, G. Figueira, and P. Amorim, "Scheduling human-robot teams in collaborative working cells," *Int. J. Prod. Econ.*, vol. 235, p. 108094, 2021.

[9] T. Bänziger, A. Kunz, and K. Wegener, "Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions," *J. Intell. Manuf.*, vol. 31, pp. 1635–1648, 2020.

[10] R. Liu, M. Natarajan, and M. C. Gombolay, "Coordinating human-robot teams with dynamic and stochastic task proficiencies," *J. Hum.-Robot Interact.*, vol. 11, no. 1, 2021.

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

[12] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *IEEE Int. Conf. Adv. Robot.*, 2005, pp. 492–497.

[13] R. Maderna, P. Lanfredini, A. M. Zanchettin, and P. Rocco, "Real-time monitoring of human task advancement," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2019, pp. 433–440.

[14] G. Maeda, M. Ewerton, G. Neumann, R. Lioutikov, and J. Peters, "Phase estimation for fast action recognition and trajectory generation in human–robot collaboration," *Int. J. Robot. Res.*, vol. 36, no. 13-14, pp. 1579–1594, 2017.