Robust assembly task assignment in Human Robot Collaboration as a Markov Decision Process problem

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 112 (2022) 174–179

www.elsevier.com/locate/procedia

15th CIRP Conference on Intelligent Computation in Manufacturing Engineering, CIRP ICME '21, Italy

Robust assembly task assignment in Human Robot Collaboration as a Markov Decision Process problem

Dario Antonelli, Khurshid Aliev

Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

* Corresponding author. Tel.: +39-011-0907202; fax: +39-011-0907299. *E-mail address:* dario.antonelli@polito.it

Abstract

Collaborative robots can work together with human workers in assembly workstations. Their drawback is the lack of flexibility that force human co-worker to bear the cognitive burden of strictly replicating every time the same tasks. To improve human-robot collaboration, human should be allowed to exchange tasks with the robot if this doesn't hinder the final assembly. The study proposes a robust real time optimization of the assembly task assignment through the modelling of the assignment problem as a Markov Decision Process with a randomly selected starting state.

Keywords: HRC;Adversarial Reinforcement learning; optimal assembly sequence planning

1. Introduction

Collaborative robots are increasingly applied in industrial context to the point that they are one of the enabling technologies of Industry 4.0 (I4.0).
The industrial applications of robots are frequent in the automotive field, see Michalos et al. (2010). Automotive assembly is a mass production sector; therefore, it is best suited for full automation with standard industrial robots. In the term assembly the following tasks are included: handling, pick and place, welding, gluing and even testing of the quality of the assembled joint.
Collaborative robots should fill a market niche characterized by small or medium volume assembly of parts where full automatization is difficult or impossible. Presently, this is the area of manual assembly. Introducing human robot collaboration (HRC) in small series assembly can lead to several benefits: robot can execute dangerous, tiresome or repetitive tasks, can offer better accuracy and less downtimes while human can dedicate to tasks that require dexterity or allows to overcome unexpected events due to incomplete standardization of the assembly operations.
The attention of robot manufacturers was completely focused on providing safe interaction between human and robot avoiding harms and damages.
On the contrary, the strategy to adopt in order to assure an efficient and effective collaboration between human and robot in the execution of the job is generally overlooked.
There are different levels of collaboration, according to ISO/TS 15066: with spatial separation, temporal separation or sharing workspace and time. Fig. 1 describes the different types of collaborative operations allowed by collaborative robots.
Presently, nearly all industrial applications apply the first two kinds of collaboration, with spatial or temporal separation between human and robot operations.
To achieve full collaboration between human and robot, it is necessary to define an effective and reliable collaboration strategy: distribute tasks to human and robot or to both simultaneously in order to exploit their reciprocal assets. This objective can be achieved by applying a task assignment method, as described in section 3.
The justification of present paper is in the assertion that a priori task planning is not sufficient when humans and robots are involved: a robust collaboration strategy must be able to question the task plan in real time during the job execution.

10.1016/j.procir.2022.09.068

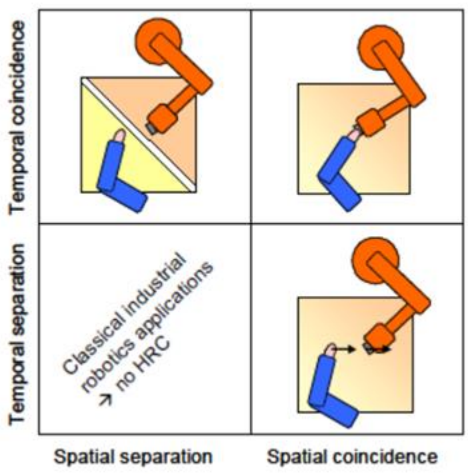

Figure 1: Types of collaborative operations (source ABB Group)

As a matter of fact, human workers, in manual operations for small series processing, do not adhere to a strict sequence of procedures. They can change the order of execution of some tasks, they can exchange tasks from time to time. This is the case if one task takes longer to complete to a worker while another worker is idle having completed his assignments.
A method is proposed to allow the task assignment strategy of the robot to adapt in real time during the work to the actual utilization state of operators or to an unscheduled action performed by the human. The method consists in modelling the task sequence as a discrete deterministic Markov Decision Process and to optimize it from random starting points. The outcome is that robot will be able to continue working to the assembly sequence in any condition by performing the tasks that allow to complete the work, even if those tasks are better fit for human or are not considered in the optimized assembly plan.

2. State of the art

There is a wide literature solving the task planning problem for robot teams and also for mixed teams of human and robots with or without collaboration. The task planning problem is the combination of task assignment with task scheduling.
A proper approach to the problem should start from the definition of the concept of task and the delimitation of the boundaries, as in Bänziger et al. (2018).
At the level of workload balancing, the problem is formalized by Ding et al. (2014) and solutions can be found in the study of Bruno and Antonelli (2018). Michalos et al. (2018) address specifically the task planning of HRC when human and robot execute shared tasks. By adopting a convenient representation of the product and of the assembly tasks (Tan, 2009) it has been possible to automatically build the assembly sequence for all the possible variants of the same product family. In this way the robot program can be executed as a composition of elementary tasks that have been programmed in advance, with the additional help of modern manual guidance programming (Massa et al., 2015).

A common characteristic of proposed solutions is that all of them adopt deterministic optimization methods and should be applied offline to plan the assembly process. However, in real life situations, robot would stop and display an exception message as soon as the human operator does not execute exactly the scheduled sequence of tasks.
To give a realistic and feasible solution to the task planning problem, the robot should be enabled to behave like a human teammate: perform tasks that have not been assigned to it, when there are delays or when the tasks assigned to it have already been performed. The robust solution will be likely suboptimal but will allow to complete the task sequence anyway in the actual context.
Markov Decision Process (MDP), coupled with Reinforcement Learning (RL) algorithms are already used in training collaborative robots to execute movements in conditions of uncertainty. In Akkaladevi (2016) the robotic platform guided by RL cognitive architecture, performs the main actions of assembly process, ‘picking’, ‘showing’, ‘placing’ and ‘handover’, on uncertain positions and respecting the manipulation preferences of the human teammate.
In the following, MDP will be used to model the task planning procedure and its solution will be used to provide a robust decision strategy to the robot during the assembly process.

3. Classification of tasks.

Every strategy for task assignment to humans and robots leads to a classification of tasks based on a scoring of how much the task is adequate either for robot or human execution. The procedure adopted here is illustrated in Bruno (2018).
As shown in fig.2, an assembly job (J) is the assembly of a set of parts in a specific position with respect each to the others. A complex assembly job can be decomposed in a hierarchical tree of sub-assembly (Makris et al., 2014). A job can be decomposed in a set of tasks (T), e.g., join two parts together. Tasks in turn are composed by an ordered set of operations (O). An operation is a basic building block, which can be programmed for robot execution or assigned to a human.

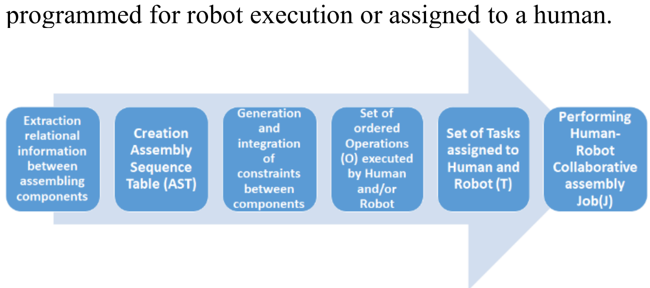

Figure 2: Hierarchy of the assembly job

For all the operations, a set of indicators can be defined. Indicators describe the features of the task that will be used as decision factor in the selection of the type of collaboration. They must be chosen in such a way to be easily assigned by the working team. Features that should be considered are the weight of the assembled part (W), the amount of displacement (Di), the presence of accuracy requirements (A) or dexterity requirements (De). Table 1 shows an example of application of the indicators to some significant operations.

Table 1: Example of indicator values for four operations

Task	W	Di	De	A
Tool retrieval	0	1	0	0
Inserting clamp	1	0	0	1
Welding	0	0	0	1
Fixing support	0	0	0	0

The job of the classifier is to classify operations depending on the agent to whom they should be assigned. Therefore, the classes are: executable only by human (H), only by robot (R), indifferently by human or robot (H/R), by both human and robot in shared work (HRC). The classifier is trained by using a training set made of previous classified data, like the ones in Tab.2. For this purpose, a C4.5 decision tree, Quinlan (1993), was used as classifier.

Table 2: Example of classified data used as training set

Task	W	Di	De	A	Class
Tool retrieval	0	1	0	0	H
Inserting clamp	1	0	0	1	HRC
Welding	0	0	0	1	R
Fixing support	0	0	0	0	H/R

After the classification, a scheduler assigns operation and estimated execution time to the workers. If an operation can be assigned either to human or to the robot, a choice is made based on time scheduling, balancing the workload on the workers. Obviously, if both the workers are idle, the operation is assigned to the robot.

4. Modelling the task assignment problem as a Markov Decision Process

Assembly process may be described as a collection of states (S), events (V) and relations (R). S defines the individual tasks of the assembly process. V drives the progress of the assembly process from one step to another. R specifies the effect of a given event V_m on a given state S_t in progressing the assembly process (Akkaladevi et al. 2016).
A state S_t is Markov if and only if:

$$P[S_{t+1}|S_t] = P[S_{t+1}|S_t, \dots, S_1]$$

P is the state transition matrix, which can describe the transition probability of two states and reflect the uncertainty of the task outcomes.

Markov decision process (MDP) is a mathematical model of the procedure of decision making in situations where outcomes not completely under the control of the decision maker. MDP describes a decision problem that can be solved by dynamic programming (DP), when the computational effort is limited, by reinforcement learning when the optimal solution requires excessive computational time (Shoham, 2003). A standard MDP is a 5-tuple (S, A, P, R, γ), where, S is a finite set of states; A is a finite set of actions; P is the state transition matrix:

$$P_{SS'}^a = P[S_{t+1} = s' | S_t = s, A_t = a]$$

R is the reward function:

$$R_S^a = E[R_{t+1}|S_t = s, A_t = a]$$

γ is the discount rate.
In initial state S_0, the agent performs the action a_0, and the state changes to S_1 correspondingly due to the action. At the same time, the instant reward r_0 has been given. Then, the agent performs action a_1 according to the state of S_1, and the system state changes to S_2, and the agent obtains the instant reward r_1, as shown in figure 3. Agents interact with the environment constantly.

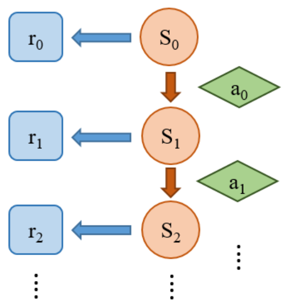

Figure 3: Markov Decision Process

In present study, the state transition is obtained by the solution of the assembly sequence problem. The reward function is induced by the value of process time with a minus sign (penalty function).
MDP states in present research correspond to the assembly operations that constitute the tasks. Only the feasible assembly tasks are considered after the application of assembly constraints, as explained in section 5.

5. Application of MDP to the case study

The case study is a laboratory experiment in which some flanges are mounted on the top of a box. In the chosen case study, the assembled part is made up of different components, i.e., a base (B) on which three flanges (F1, F2, S) are mounted and joined by screwed bolts (Aliev et al. 2019). Human and the collaborative robot need to collaborate joining the flanges by using screws and nuts. Uncountable variants of this assembly are possible, allowing for the generation of continuously new assembly sequences. The graphic visualization of the assembly diagram derives from the standard representation proposed by De Mello (1986).

a

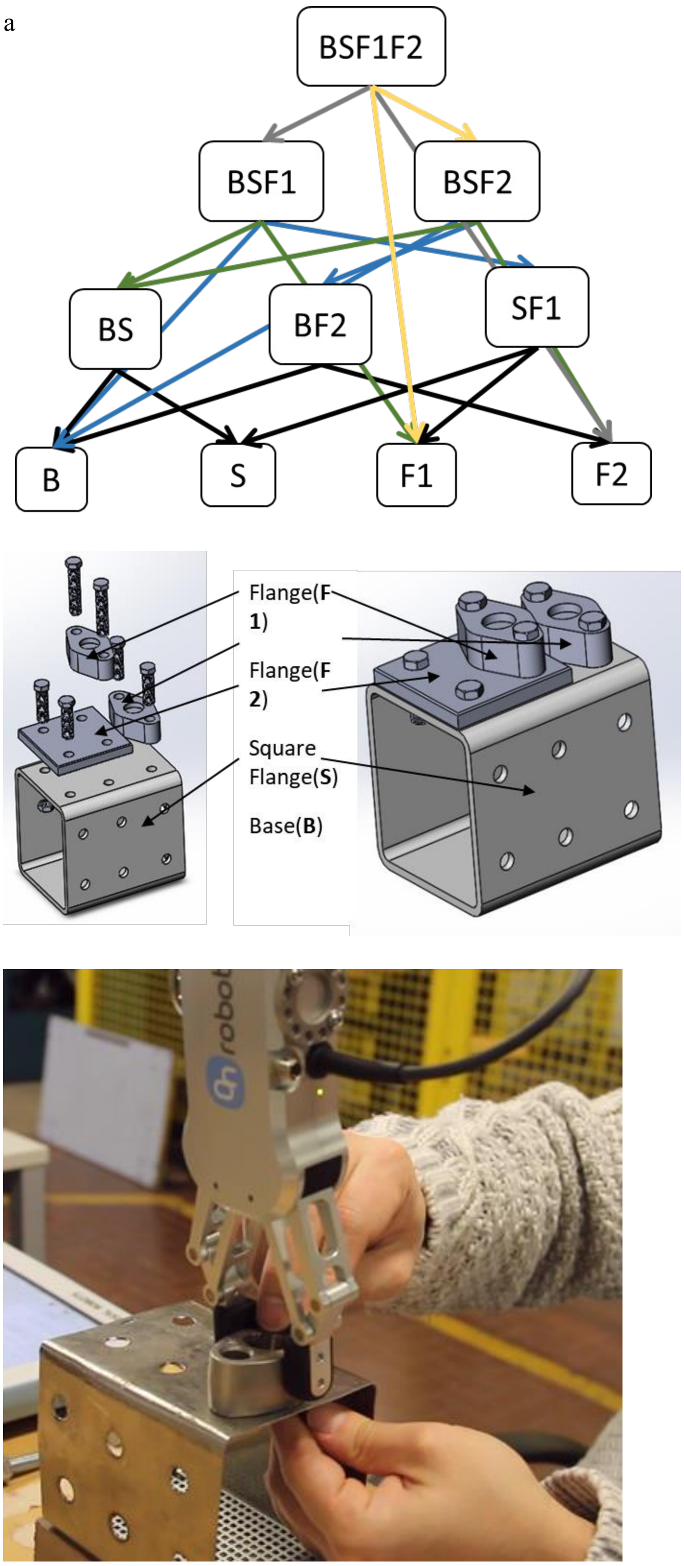

Fig. 3. Synthetic description of one variant of assembly case study: (a) assembly diagram, (b) drawing of the components to be assembled, and (c) example of assembly task performed collaboratively.

The optimal assembly sequence is obtained after the application of assembly constraints, by minimizing the completion time. For the formal definition and the application of the constraints, the assembly sequence generation method of Gottipolu and Ghosh (2003) and their formalism was applied. The formalism is composed of a set of operations and of topological, functional and stability constraints. Constraints are represented as matrices with assembly pairs in the rows and contact and translation functions in the column. In Antonelli (2019) the method has been already applied to this case study. The optimal assembly sequence was found as:

$$\{B,S\} \to \{B,S\},\{F_1\} \to \{B,S,F_1\},\{F_2\}$$

The meaning of the formalism is that there are 3 tasks to be executed in sequence: mount S on B, F1 on the sub-assembled of S and B, F2 on the sub-assembled made of S, B and F1.
Every task can be decomposed in a number of elementary operations that will be programmed using the manual guidance programming, a functionality present in all industrial collaborative robots.
After the evaluation of the collaborative indicators and the application of the classifier, the resulting classification can be found in Table 3. The last column of the table describe the distribution of workload between human and robot.

Table 3: Classification of the case study operations

N	Operation	Time [s]	Prec.	W	Di	De	A	Class
1	Fill workspace	60		0	1	0	0	H
2	Mounting the tool	128		0	0	1	0	H
3	Fetch flange S	20	2	0	0	0	0	H/R
4	Place flange S	20	3	0	0	0	1	R
5	Pick flange 1	6	4	0	0	0	0	H/R
6	Position flange1	4	5	1	0	0	1	R
7	Hold flange 1	14	6	1	0	0	0	R
8	Pick screw 1	3	6	0	0	0	0	H/R
9	Insert screw 1	4	8	0	0	1	0	H
10	Pick screw 2	3	6	0	0	0	0	H/R
11	Insert screw 2	4	10	0	0	1	0	H
12	Pick nut 1	3	9	0	0	0	0	H/R
13	Screw nut 1	40	12	0	0	1	0	H
14	Pick nut 2	3	10	0	0	0	0	H/R
15	Screw nut 2	40	14	0	0	1	0	H
16	Pick flange 2	12	7	0	0	0	0	H/R
17	Position flange2	19	16	1	0	0	1	R
18	Hold flange 2	14	17	1	0	0	0	R
19	Pick screw 3	3	16	0	0	0	0	H/R
20	Insert screw 3	4	19	0	0	1	0	H
21	Pick screw 4	3	16	0	0	0	0	H/R
22	Insert screw 4	4	21	0	0	1	0	H
23	Pick nut 3	3	20	0	0	0	0	H/R
24	Screw nut 3	40	23	0	0	1	0	H
25	Pick nut 4	3	22	0	0	0	0	H/R
26	Screw nut 4	40	25	0	0	1	0	H

Now the next step is the scheduling of tasks, based on the process times in Table 3. The choice between human and robot, whenever an operation can be performed by either one, follows the simple policy to load the robot if idle, otherwise the human. The Gantt table in fig. 4 shows the scheduling of the assembly and the assignment, cyan for human, green for robot.

The last step is now to return to Table 3 and to convert all and only the operations that can be assigned by choice to human or robot in Markov states.

The state in Table 4 is represented as a tuple made by the task, the slot position where the two parts are assembled and the actions that can be executed from this state. The possible actions are: executed by human or executed by robot. Some tasks are terminal, corresponding to the completion of the assembly. To understand the logic behind task generation it is important to remind that every new transition in MDP is determined only by the preceding state and action. Therefore, every Markov state must keep memory of the whole assembly sequence so far. As an example, the states S5 and S7 are different because they have been obtained from a different operation sequence.

Table 4. State list: feasible assembly tasks

State	Task	Prec.	Action (Reward)
1	Fetch flange S	-	H(1) , R(2)
2	Pick flange 1	1	H(1) , R(2)
3	Pick screw 1	2	H(2) , R(1)
4	Pick screw 2	3	H(2) , R(1)
5	Pick nut 1 after screw 1	3	H(2) , R(1)
6	Pick nut 2 after screw 1	3	R(1)
7	Pick nut 1 after screw 2	4	R(1)
8	Pick nut 2 after screw 2	4	H(2) , R(1)
9	Pick flange 2	6,8	H(1) , R(2)
10	Pick screw 3	9	H(2) , R(1)
11	Pick screw 4	10	H(2) , R(1)
12	Pick nut 3 after screw 3	10	H(2) , R(1)
13	Pick nut 4 after screw 3	10	R(1)
14	Pick nut 3 after screw 4	11	R(1)
15	Pick nut 4 after screw 4	11	H(2) , R(1)

MDP in Table 4 is the list of states. Transition among states happens when an action is decided. The action is to assign the operation to the human or to the robot. The reward the system receives by choosing a given action in a given state is shown in parenthesis in the column of the action. In this application of MDP both actions lead to the same transition of states.

If we compare Table 4 with Fig.4 we will see that the robot is given better rewards in the operations that are assigned to it in the scheduling. Conversely the human receives better rewards in the states corresponding to human executed operations.

This MDP problem is a finite-horizon discrete deterministic optimization and can be solved by Dynamic Programming (DP) with a backward induction algorithm. The algorithm is used in the version implemented in Matlab by the decision team of the Biometry and Artificial Intelligence Unit of INRA Toulouse (Marie-Josee Cros, 2021). The optimality equations allow to recursively evaluate function values starting from the terminal stage. For every stage the function displays the current stage and the corresponding optimal policy that is the policy that provides the maximum cumulative reward. The policy for the decision maker is a function π that specifies the action π (s) that the decision maker will choose when in state s.

If the human would always follow the assigned operations in the right order, the robot could follow the optimal policy with the maximum cumulative reward. Sometimes the human decides other ways, by picking nut 2 after having picked screw 1, as an example. The system will be in state 6 that is a state non considered in the optimal task sequence. The optimal policy starting from state 6 is that the robot picks nut 1 and assists the human in completing the joining. This is not the overall optimal policy, but it is the best policy starting from such a sub-optimal state.

6. Conclusions

The paper addresses the problem of enhancing industrial HRC through robust task planning of assembly operations, making the robot program more flexible. In the case of a collaborative assembly job executed by human and robot in a shared workspace, flexibility at trajectory level is obtained by making robot movements insensitive to disturbances. Concerning task management, flexibility is obtained adapting task sequence to the actual human decisions in terms of tasks, at the price of giving up optimality.

While MDP with RL has been extensively exploited to improve robot flexibility at the trajectory level, it has not been considered until now as a way for obtained robust solutions to the task planning problem in presence of disturbances caused by unpredictable and variable human behaviour.

This study obtains a flexible management of task assignment to human and robot having the goal of real time updating the assembly tasks schedule to meet the actions of the human teammate. MDP problem, coupled with DP not only finds the best task assignment, but also explore the solutions space for viable alternatives when the system is in a state out of the optimal track. The robot can execute the tasks that maximize the result starting from the human moves.

Despite the overall good performances of the method, results have required a consistent amount of manual work in the set-up of the MDP. It is apparent that this is unpractical in a realistic full complexity assembly process.

Next research step will be therefore the automatic generation of MDP, starting from feasible assembly tasks.

References

Akkaladevi, S. C., Plasch, M., Pichler, A., & Rinner, B. (2016, August). Human Robot Collaboration to Reach a Common Goal in an Assembly Process. In STAIRS (pp. 3-14). [16]

Aliev K, Antonelli D, Bruno G. (2019). Task-based Programming and Sequence Planning for Human-Robot Collaborative Assembly[C]// 9th IFAC Conference on manufacturing modelling, management and control MIM, Berlin, 52(13): 1638-1643.

Antonelli, D., & Bruno, G.: Dynamic distribution of assembly tasks in a collaborative workcell of humans and robots. FME Transactions, 47(4), 723-730, 2019

Bänziger, T., Kunz, A., & Wegener, K. (2018). Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions. Journal of Intelligent Manufacturing, 1-14.

Bruno, G., & Antonelli, D. (2018). Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells. The International Journal of Advanced Manufacturing Technology, 98(9), 2415-2427.

De Mello, L.S. and Sanderson, A.C., 1986. AND/OR Graph representation of assembly plans (no. cmu-ri-tr-86-8). carnegie-mellon univ pittsburgh pa robotics inst.

Ding, H., Schipper, M., Bjoern, M. (2014) Optimized task distribution for industrial assembly in mixed human robot environments case study on IO module assembly, IEEE International Conference on Automation Science and Engineering

Freedy, A., DeVisser, E., Weltman, G., & Coeyman, N. (2007, May). Measurement of trust in human-robot collaboration. In Collaborative Technologies and Systems, 2007. CTS 2007. International Symposium on (pp. 106-114). IEEE.

Gottipolu, R.B. and Ghosh, K., 2003. A simplified and efficient representation for evaluation and selection of assembly sequences. Computers in Industry, 50(3), pp.251-264.

Howard, Ronald A. (1960). Dynamic Programming and Markov Processes. The M.I.T. Press.

Marie-Josee Cros (2021). Markov Decision Processes (MDP) Toolbox (https://www.mathworks.com/matlabcentral/fileexchange/25786-markov-decision-processes-mdp-toolbox), MATLAB Central File Exchange. Retrieved June 16, 2021.

Massa, D., Callegari, M., Cristalli, C. (2015) Manual guidance for industrial robot programming. Industrial Robot: An International Journal, 42(5), 457-465. [19]

Michalos, G., Makris, S., Papakostas, N., Mourtzis, D., & Chryssolouris, G. (2010). Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach. CIRP Journal of Manufacturing Science and Technology, 2(2), 81-91.

Michalos, G., Spiliotopoulos, J., Makris, S., & Chryssolouris, G. (2018). A method for planning human robot shared tasks. CIRP journal of manufacturing science and technology, 22, 76-90.

Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

Shoham, Y., Powers, R., & Grenager, T. (2003). Multi-agent reinforcement learning: a critical survey (pp. 1-13). Technical report, Stanford University.

Tan, J. T. C., Duan, F., Zhang, Y., & Arai, T. (2009, February). Extending task analysis in HTA to model man-machine collaboration in cell production. In Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on (pp. 542-547). IEEE.

ID	Task	Duration [s]	Precedent	Classification	Assignment
1	Positioning parts	60	-	H	H
2	Mounting tool	168	-	H	H
3	Fetch base	20	1;2	H/R	R
4	Place base	20	3	R	R
5	Pick flange 1	6	4	H/R	R
6	Positioning flange 1	4	5	R	R
7	Hold flange 1	100	6	R	R
8	Pick screw 1	3	6	H/R	H
9	Insert screw 1	4	8	H	H
10	Pick screw 2	3	6	H/R	H
11	Insert screw 2	4	10	H	H
12	Pick nut 1	3	9	H/R	H
13	Screw nut 1	40	12	H	H
14	Pick nut 2	3	10	H/R	H
15	Screw nut 2	40	14	H	H
16	Pick flange 2	12	4;7	H/R	R
17	Positioning flange 2	19	16	R	R
18	Hold flange 2	100	17	R	R
19	Pick screw 3	3	16	H/R	H
20	Insert screw 3	4	19	H	H
21	Pick screw 4	3	16	H/R	H
22	Insert screw 4	4	21	H	H
23	Pick nut 3	3	20	H/R	H
24	Screw nut 3	40	23	H	H
25	Pick nut 4	3	22	H/R	H
26	Screw nut 4	40	25	H	H

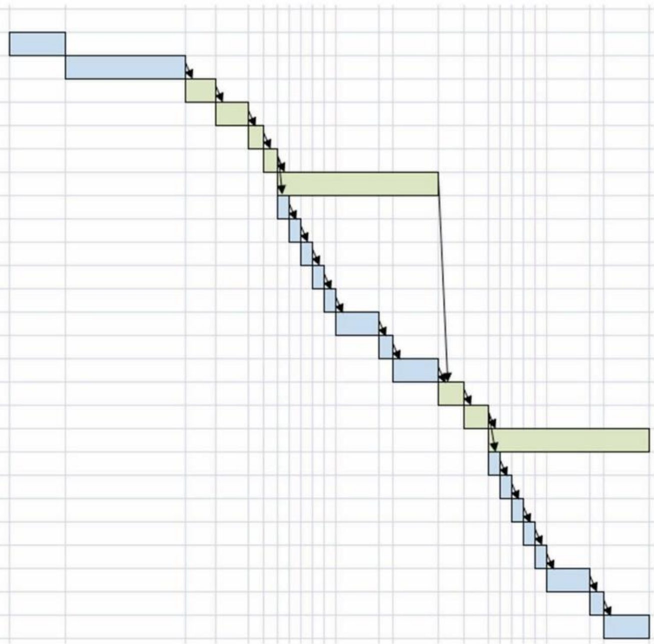

Fig. 4. Assignment of tasks and corresponding Gantt chart (in blue tasks assigned to the human, in green tasks assigned to the robot)