

General Framework for the Optimization of the Human-Robot Collaboration Decision-Making Process Through the Ability to Change Performance Metrics

Mélodie Hani Daniel Zakaria, Sébastien Lengagne, Juan Antonio Corrales Ramón, Youcef Mezouar

▶ **To cite this version:**

Mélodie Hani Daniel Zakaria, Sébastien Lengagne, Juan Antonio Corrales Ramón, Youcef Mezouar. General Framework for the Optimization of the Human-Robot Collaboration Decision-Making Process Through the Ability to Change Performance Metrics. Frontiers in Robotics and AI, 2021, 8, 10.3389/frobt.2021.736644 . hal-03474401

HAL Id: hal-03474401
https://uca.hal.science/hal-03474401

Submitted on 10 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 25 October 2021
doi: 10.3389/frobt.2021.736644

General Framework for the Optimization of the Human-Robot Collaboration Decision-Making Process Through the Ability to Change Performance Metrics

Mélodie Hani Daniel Zakaria[1]*, *Sébastien Lengagne*[1], *Juan Antonio Corrales Ramón*[2] and *Youcef Mezouar*[1]

[1]CNRS, Clermont Auvergne INP, Institut Pascal, Université Clermont Auvergne, Clermont-Ferrand, France, [2]Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain

OPEN ACCESS

Edited by:
Yanan Li,
University of Sussex, United Kingdom

Reviewed by:
Xiaoxiao Cheng,
Imperial College London, United Kingdom
Selma Music,
Technical University of Munich, Germany

***Correspondence:**
Mélodie Hani Daniel Zakaria
Melodie.HANI_DANIEL_ZAKARIA@uca.fr
melodie.daniel@yahoo.fr

Specialty section:
This article was submitted to Human-Robot Interaction, a section of the journal Frontiers in Robotics and AI

Received: 05 July 2021
Accepted: 28 September 2021
Published: 25 October 2021

Citation:
Hani Daniel Zakaria M, Lengagne S, Corrales Ramón JA and Mezouar Y (2021) General Framework for the Optimization of the Human-Robot Collaboration Decision-Making Process Through the Ability to Change Performance Metrics. Front. Robot. AI 8:736644. doi: 10.3389/frobt.2021.736644

This paper proposes a new decision-making framework in the context of Human-Robot Collaboration (HRC). State-of-the-art techniques consider the HRC as an optimization problem in which the utility function, also called reward function, is defined to accomplish the task regardless of how well the interaction is performed. When the performance metrics are considered, they cannot be easily changed within the same framework. In contrast, our decision-making framework can easily handle the change of the performance metrics from one case scenario to another. Our method treats HRC as a constrained optimization problem where the utility function is split into two main parts. Firstly, a constraint defines how to accomplish the task. Secondly, a reward evaluates the performance of the collaboration, which is the only part that is modified when changing the performance metrics. It gives control over the way the interaction unfolds, and it also guarantees the adaptation of the robot actions to the human ones in real-time. In this paper, the decision-making process is based on Nash Equilibrium and perfect-information extensive form from game theory. It can deal with collaborative interactions considering different performance metrics such as optimizing the time to complete the task, considering the probability of human errors, etc. Simulations and a real experimental study on "an assembly task" -i.e., a game based on a construction kit- illustrate the effectiveness of the proposed framework.

Keywords: human-robot collaboration, decision-making, game theory, Nash equilibrium, interaction optimality

1 INTRODUCTION

Nowadays, Human-Robot Collaboration (HRC) is a fast-growing sector in the robotics domain. HRC aims to make everyday human tasks easier. It is a challenging research field that interacts with many others: psychology, cognitive science, sociology, artificial intelligence, and computer science (Seel, 2012). HRC is based on the exchange of information between humans and robots sharing a common environment to achieve a task as teammates with a common goal (Ajoudani et al., 2018).

HRC applications can have social and/or physical benefits for humans (Bütepage and Kragic, 2017). Social collaboration tasks include social, emotional and cognitive aspects (Durantin et al.,

2017) such as care for the elderly (Wagner-Hartl et al., 2020), therapy (Clabaugh et al., 2019), companionship (Hosseini et al., 2017), and education (Rosenberg-Kima et al., 2019). Social robots, such as Nao, Pepper, iCub, etc., are dedicated to this type of task; however, their physical abilities are very limited (Nocentini et al., 2019). For the physical HRC (pHRC), physical contacts are necessary to perform the task. They can occur directly between humans and robots or indirectly through the environment (Ajoudani et al., 2018). pHRC applications are mainly used in industrial environments [e.g., assembly, handling, surface polishing, welding, etc., (Maurtua et al., 2017)]. pHRC is also used in the Advanced Driver-Assistance Systems (ADAS) for autonomous cars (Flad et al., 2014).

Robots can adapt to humans in different situations by implementing five steps in a decision-making process (Negulescu, 2014): 1) gathering relevant information on possible actions, environment, and agents, 2) identifying alternatives, 3) weighing evidence, 4) choosing alternatives and selecting actions, and 5) examining the consequences of decisions. These steps are usually modeled in computer science using a decision-making method with a strategy and a utility function (Fülöp, 2005). The decision-making method models the whole situation (environment, actions, agents, task restrictions, etc.). The strategy defines the policy of choosing actions based on the value of their reward. The utility function (i.e., reward function) evaluates each action for each alternative by attributing a reward to it.

On the one hand, previous works, known as leader-follower systems (Bütepage and Kragic, 2017), focused the decision process on choosing the actions that increase the robot's abilities to accomplish the task without considering how the collaboration is done. Such as in DelPreto and Rus (2019), where a human-robot collaborative team is lifting an object together, or in Kwon et al. (2019), where robots influence humans to change the pre-defined leader-follower agents to rescue more people when there is a plane or ship crash in the sea.

On the other hand, other works deal with maximizing the collaboration performance by promoting mutual adaptation (Nikolaidis et al., 2017b; Chen et al., 2020) or reconsidering the task allocation (Malik and Bilberg, 2019). However, they only consider one or two unchangeable performance metrics for this evaluation in their utility function: postural or ergonomic optimization (Sharkawy et al., 2020), time consumption (Weitschat and Aschemann, 2018), trajectory optimization (Fishman et al., 2019), cognitive aspects (Tanevska et al., 2020), and reduction of the number of human errors (Tabrez and Hayes, 2019).

In this paper, we optimize and quantitatively assess the collaboration between robots and humans based on the resulting impact of some changeable performance metrics on human agents. Hence, an optimized collaboration aims to bring a benefit to humans, such as getting the task done faster or reducing the effort of human agents. However, an unoptimized collaboration will bring nothing to humans or, on the contrary, will represent a nuisance, such as slowing them down or overloading them, even if the task is finally accomplished. The main contribution of this paper is that the proposed framework allows optimizing the performance, based on some changeable metrics, of the collaboration between one or more humans and one or more robots. Contrary to previous works, our framework allows us to easily change the performance metrics without changing the whole way the task is formalized since we isolate the impact of the metrics in the utility function.

The benefit of this contribution is to increase the collaboration performance without having to ameliorate the robot's abilities. This is important in relevant practical cases: for instance, when using social robots that have great limitations (e.g., slowness in their movements and/or reduced dexterity), and it is not easy or even possible to ameliorate their abilities drastically. Therefore, our work provides an interesting solution to enhance collaboration performance with such limited robots.

Our framework uses the state-of-the-art decision-making process composed of: a decision-making method, a strategy, and a utility function. We divide the utility function into two main parts: the collaboration performance evaluated by a reward according to one or several performance metrics, and the task accomplishment, which is considered as a constraint since we only deal with achievable tasks.

The paper is organized as follows. First, we review related work in **Section 2**. Then, we present our framework formalization in **Section 3**. **Section 4** includes all the details regarding the decision-making process. The effectiveness of our new formalization is shown in **Section 5** based on simulated and experimental tests of an assembly task (i.e., a game[1] that involves placing cubes to build a path between two figurines) shown in **Figure 1**. Finally, we sum up the effectiveness of our contribution and discuss the possible improvements in **Section 6**.

2 RELATED WORK

In this section, we present the most popular methods, strategies, utility functions, and performance metrics used in the decision-making process of human-robot collaboration to place our contributions with regard to them. A decision-making method models the relationship between the agents, the actions, the environment, the task, etc. A strategy defines how to select the optimal actions each agent can choose based on the reward (utility) calculated by the utility function for each action. An optimal action profile is made up of the best actions each agent can choose. All methods and strategies can be used to perform different tasks, and there is no pre-written rule that implies that one will necessarily perform better than the others.

2.1 Decision-Making Methods

Decision-making methods are used, as mentioned before, to model the relationship between the task, the agents accomplishing it, their actions, and their impact on the environment. Probabilistic methods, deep learning, and game

[1]Camelot Jr. is a game created by Smart Games: https://www.smartgames.eu/uk/one-player-games/camelot-jr.

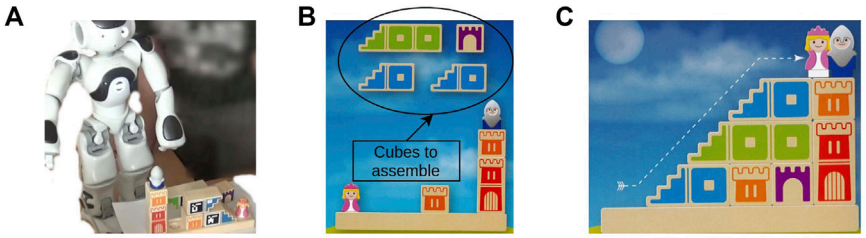


FIGURE 1 | Agents solving the Camelot Jr. game. **(A)** Agents play sequentially: the human starts to play, and then it is the robot's turn. **(B)** This puzzle starts with four cubes to assemble. **(C)** The cubes are correctly assembled, and the puzzle is solved (i.e., a path composed by cubes is created between both figurines).

theory are considered among the most widespread decision-making methods.

Probabilistic methods are the first and most widely used in decision-making processes. Markov's decision-making processes (e.g., Markov chains) are the most used ones. There are also studies based on other methods such as Gaussian processes (e.g., Gaussian mixtures), Bayesian processes (e.g., likelihood functions), and graph theory. In Roveda et al. (2021), a Hidden Markov Model (HMM) is used to teach the robot how to achieve the task based on human demonstrations, and an algorithm based on Bayesian Optimization (BO) is used to maximize task performance (i.e., avoid task failures while reducing the interaction force), and to enable the robot to compensate for task uncertainties.

The interest in using deep learning in decision-making methods began very early due to unsatisfactory results of probabilistic methods in managing uncertainties in complex tasks. In Oliff et al. (2020), Deep Q-Networks (DQN) are used to adapt robot behavior to human behavior changes during industrial tasks. The drawbacks of deep learning methods are the computation cost and the slowness of learning.

Game theory methods in decision-making processes have only recently been exploited. They can model most of the tasks performed by a group of agents (players) in collaboration or competition, whether the choice of actions is simultaneous [normal form also called matrix form modeling (Conitzer and Sandholm, 2006)] or sequential [extensive form also known as tree form modeling (Leyton-Brown and Shoham, 2008)]. The game theory methods have been used in different HRC applications, for instance, in analyzing and detecting the human behavior to adapt the robot's one to it for reaching a better collaboration performance (Jarrassé et al., 2012; Li et al., 2019). Game theory has been also utilized in HRC in mutual adaptation to achieve industrial assembly scenarios (Gabler et al., 2017). We choose the game theory as a decision-making method due to its simplicity and effectiveness in modeling most interactions between a group of participants and their reactions to each other's decisions. We specifically use the extensive form due to its sequential nature, which is suitable for HRC applications.

2.2 Decision-Making Strategies

The decision-making strategy is the policy of choosing actions based on the value of their reward calculated by the utility function (i.e., the reward function). We present the most used strategies for multi-criteria decision-making in HRC as well as some of their application areas. The following strategies are used intensively in deep learning and/or in Game theory (Conitzer and Sandholm, 2006; Leyton-Brown and Shoham, 2008):

- Dominance: All the actions whose rewards are dominated by others are eliminated. Researchers used it to assess the human's confidence in a robot in Reinhardt et al. (2017).
- Pareto optimality: An action profile is Pareto optimal if we cannot change it without penalizing at least one agent. It is used, for example, in disassembly and remanufacturing tasks (Xu et al., 2020).
- Nash Equilibrium (NE): Each agent responds to the others in the best possible way. The best response is the best actions an agent can choose whatever others have done. This is the main strategy used in Game theory. In Bansal et al. (2020), a NE strategy is used to ensure human safety in a nearby environment during a pick-and-place task.
- Stackelberg duopoly model: The agents make their decision sequentially: one agent (the leader) makes their decision first, and all other agents (followers) decide after. The optimal action of the leader will be the one that maximizes its own reward and minimizes the follower's rewards. This means that the leader has always the biggest reward. This strategy is used, for example, in a collaborative scenario between a human and a car to predict the driver's behavior in this specific scenario (Li et al., 2017) such as the driver's steering behavior in response to a collision avoidance control (Na and Cole, 2014).

2.3 Performance Metrics

After the decision-making process is settled and used to perform a task by a human-robot collaborative team, other works tend to evaluate the performance of the collaboration using some performance metrics. On the one hand, some works focused on evaluating one specific metric, as done in Hoffman (2019), where the author is evaluating several human-robot collaborative teams, performing different tasks, using the fluency metric. On the other hand, other works create a global framework to evaluate, in general, the HRC based on several metrics. In Gervasi et al. (2020), the authors developed a global framework to evaluate the HRC based on more than twenty performance metrics, among which the cognitive load and the physical ergonomics. **Table 1**

TABLE 1 | Some metrics considered for the evaluation of HRC classified based on the task types (Steinfeld et al., 2006; Bütepage and Kragic, 2017; Nelles et al., 2018).

Task	Navigation	Perception	Management	Manipulation	Social	Common metrics that can be used for all task types
Performance metrics	Failure rate, accuracy, ergonomy or posture, time to completion, and rapidity	Velocity, accuracy, time to completion, effectiveness, and number of errors	Time delivery, time request, number of human and robot errors, trust, cognitive load	Positional accuracy and repeatability, velocity, dexterity, time to completion, and effort or force	Persuasiveness, engagement in social characteristics, trust, and compliance	Time to completion, number of human and robot errors, autonomy, cognitive load, and effectiveness

presents the main metrics considered, in the state-of-the-art, to evaluate the optimality of the collaboration. We present in the **Supplementary Material** a more detailed table that introduces more performance metrics and defines each metric according to its usage in different task types.

2.4 Utility Functions

The utility is a reward calculated by the utility function to express the value of an action. Thanks to these utilities, the decision-making strategy can choose the right actions. Some previous works in the literature only considered task accomplishment (and no performance metrics) in their utility functions because their focus was on complex task accomplishment. For example, in Nikolaidis et al. (2017a), a human-robot collaborative team was carrying a table to move it from one room to another. The goal was to ensure mutual adaptation between the agents by having the human also adapt to the robot. In this type of work, none of the performance metrics in **Table 1** is considered.

More recent works include performance metrics (see **Table 1**). However, they considered that they are not changeable without significant changes in their framework. A relevant example is Liu et al. (2018) where, by changing the task allocation, the authors make the robot respect the real-time duration of the assembly process while following the necessary order to assemble the parts. In this case, they considered one metric (the time to completion) since respecting the part's assembly order is a constraint to accomplish the task. However, this time metric cannot be replaced by another (e.g., effort or velocity) using this framework.

2.5 Contributions

Unlike the utility functions used in the state-of-the-art works, we take into account a changeable unrestricted number of performance metrics (from **Table 1**) that are usually optimized no matter how the human is behaving. To summarize our contributions, we propose a framework that allows us to:

- easily change the performance metrics from one scenario to another without changing anything in our formalization except the part in the utility function related to the metrics, and
- improve the collaboration performance without having to change the robot's abilities.

In the following section, we define the problem formalization and present the utility function which optimizes the performance metrics and aims to accomplish the task as a constraint.

3 FORMALIZATION

A HRC[2] consists of a global environment $\{\mathbf{E}\}$ and a task T. The environment state E^k at each iteration k (with $k \in [1, k_f]$, where k_f is the final iteration of the task) comprises a group of n agents (humans and robots), each of them can carry out a finite set of actions (continuous or discrete). E^k changes according to the actions chosen by the agents. The global environment $\{\mathbf{E}\}$ is the set of changes in the environment state at each iteration.

$$\{\mathbf{E}\} = \{E^1, E^2, \ldots, E^k, \ldots, E^{k_f}\} \tag{1}$$

Since the possible actions may change at each iteration, we define $\{\mathbf{A}\}$ as the global set of feasible actions for each iteration k: $\{\mathbf{A}\}^k$.

$$\{\mathbf{A}\} = \{\{\mathbf{A}\}^1, \{\mathbf{A}\}^2, \ldots, \{\mathbf{A}\}^k, \ldots, \{\mathbf{A}\}^{k_f}\} \tag{2}$$

The set $\{\mathbf{A}\}^k$ contains a set of feasible actions for each agent i (with $i \in [1, n]$) at iteration k denoted by $\{\mathbf{A}_i\}^k$.

$$\{\mathbf{A}\}^k = \{\{\mathbf{A}_1\}^k, \ldots, \{\mathbf{A}_i\}^k, \ldots, \{\mathbf{A}_n\}^k\} \tag{3}$$

$A^k_{i,a}$ is the a^{th} feasible action of agent i where $a \in [1, l]$ and l is the number of feasible actions of the agent i at time k.

$$\{\mathbf{A}_i\}^k = \{A^k_{i,1}, \ldots, A^k_{i,a}, \ldots, A^k_{i,l}\} \tag{4}$$

At each iteration, an action profile $(\vec{\mathbf{A}})^k$ groups the actions chosen by each agent i denoted by $A^k_i \subset \{\mathbf{A}_i\}^k$.

$$(\vec{\mathbf{A}})^k = (A^k_1, \ldots, A^k_i, \ldots, A^k_n) \tag{5}$$

The optimal action profile $(\vec{\mathbf{A}})^k_{opt}$ at iteration k is computed through the decision-making function $\mathbf{d}_{M,S}$ as presented in **Eq. 6**. $\mathbf{d}_{M,S}$ relies on the decision-making method M, the decision-making strategy S, and the utility profile $(\vec{\mathbf{U}})^k_A$ that contains all the utilities for all possible actions $\{\mathbf{A}\}^k$ at iteration k. The

[2]We denote functions by lower case letters in bold, sets and subsets between braces with upper case letters in bold, indexes by lower case letters, parameters by upper case letters, and vectors (i.e., profiles) by letters in bold topped by an arrow between parenthesis.

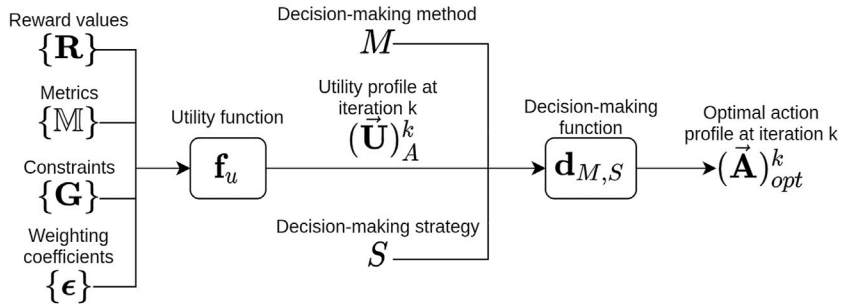


FIGURE 2 | Block diagram of our formalization of the decision-making process used to calculate the optimal action profile $(\vec{\mathbf{A}})^k_{opt}$ at iteration k.

decision-making method M takes into account the constraints related to the task such as the order in which the agents act, i.e., sequentially or simultaneously. The decision-making strategy S defines the way the agents must choose the actions according to their utilities contained in $(\vec{\mathbf{U}})^k_A$, as presented in **Figure 2**.

$$(\vec{\mathbf{A}})^k_{opt} = \mathbf{d}_{M,S}\left((\vec{\mathbf{U}})^k_A\right) \tag{6}$$

The utility profile $(\vec{\mathbf{U}})^k_A$ is computed by the utility function $\mathbf{f}_u$ based on different sets including: 1) the set of performance metrics $\{\mathbb{M}\}$ (cf. **Table 1**), 2) the set of constraints $\{\mathbf{G}\}$ to be respected in order to make the task T progress for accomplishing it, 3) $\{\mathbf{R}\}$ the reward of each action in the profile action which is calculated according to the task and the metrics, and 4) $\{\boldsymbol{\epsilon}\}$ a set of weighting coefficients (between 0 and 1) used to determine the importance of each metric (e.g., favoring one metric over the others, especially when it is in opposition to others). We get:

$$(\vec{\mathbf{U}})^k_A = \mathbf{f}_u(\{\mathbb{M}\},\{\mathbf{G}\},\{\mathbf{R}\},\{\boldsymbol{\epsilon}\}) = (U^k_{A_1},\ldots,U^k_{A_i},\ldots,U^k_{A_n}) \tag{7}$$

Let us discuss how one can make changes to the different elements involved in **Eq. 7**. To only change the scenario of the collaboration by changing the performance metrics $\{\mathbb{M}\}$ in $\mathbf{f}_u$, we first need to change the value of the metrics $\{\mathbb{M}\}$, and the value of the reward $\{\mathbf{R}\}$ of each action, and afterwards recalculate the utilities $(\vec{\mathbf{U}})^k_A$. To only modify the agent's actions, the utilities $(\vec{\mathbf{U}})^k_A$ should be recalculated for the new actions. To change the task, we will need to modify the constraints $\{\mathbf{G}\}$ which define the task by setting the conditions that allow the agent to only choose among the actions which permit to make the task progress. It will then be necessary to recalculate the utilities $(\vec{\mathbf{U}})^k_A$. We can, of course, combine several modifications (e.g., changing the performance metrics and the task) by making the appropriate adaptations (e.g., first modifying the metrics $\{\mathbb{M}\}$, the rewards $\{\mathbf{R}\}$, and the task constraints $\{\mathbf{G}\}$, and afterwards recalculating the utilities $(\vec{\mathbf{U}})^k_A$).

To illustrate how **Eqs 6**, **7** can be settled, let us consider the example of a collaborative team composed of a human and a robot, each holding an edge of a gutter on which there is a ball (Ghadirzadeh et al., 2016). Their goal is to position the ball, for instance, in the center of the gutter. Our solution using our formalization for such task can be as follows:

- Agents: Agent 1 is the human, and agent 2 is the robot. Both agents are making decisions simultaneously.
- Human actions $\{\mathbf{A}_1\}^k$: They are the angles of inclination of the gutter. The actions are continuous. The set of human actions remains the same for all the iterations ($\{\mathbf{A}_1\}^k = \{\mathbf{A}_1\}$).
- Robot actions $\{\mathbf{A}_2\}^k$: They are angles of inclination of the gutter by the robot end-effector. The decision-making method will provide the correspondent joint values needed to reach the desired position by the end-effector. The actions are continuous (since it is a continuous control task). The set of robot actions remains the same for all iterations ($\{\mathbf{A}_2\}^k = \{\mathbf{A}_2\}$).
- Constraints $\{\mathbf{G}\}$: The angles of inclination should be between $[-30^\circ, 30^\circ]$, other values will be penalized.
- Performance metrics $\{\mathbb{M}\}$: Time to completion and human posture. Human posture is measured by ISO standards that define some uncomfortable work postures (Delleman and Dul, 2007). These uncomfortable postures (or positions) will lead, for example, to find that when the human inclines the gutter with an angle out of the interval $[-20^\circ, 20^\circ]$, it is getting painful for them.
- Rewards $\{\mathbf{R}\}$: They will be calculated by the following equation: $-\|C_b - C_g\| * \lambda$. Where C_b is the position of the center of the ball, C_g is the position of the center of the gutter (the desired position), and λ is a fixed gain for a case scenario. λ allows to privileged an action according to the performance metrics ($\{\mathbb{M}\}$) and the constraints ($\{\mathbf{G}\}$).
- Weighting coefficients $\{\boldsymbol{\epsilon}\}$: It is equal to 1 for both performance metrics.
- Decision-making method M: It is the reinforcement learning process that is based on trial and error learning. The agent 2 (the robot which learns) in state s makes an action $A_{2,a}$ which changes the state to s'. The observation the agent got from the environment describes the changes that happened by moving from state s to s'. The reward ($R(s, A_{2,a})$) evaluates the taken action $A_{2,a}$ (which leads to the new state s') with respect to the desired learning goal. The state s is made up of C_b, C_g, and the position of the robot's end-effector. The learning procedure of all reinforcement learning algorithms consists of learning the value that is attributed to the state $V(s)$ defined below.

- Decision-making strategy S: It is the dominance strategy. Once the $V(s)$ are learned for all possible states, the optimal actions can be chosen. Most of the reinforcement learning algorithms are based on the Bellman equation for choosing the optimal actions (Nachum et al., 2017):

$$V(s) = \max_{A_{2,a}} (R(s, A_{2,a}) + \gamma V(s')) \tag{8}$$

γ is the discount factor that determines how much agent 2 cares about rewards in the distant future relative to those in the immediate future. $\max_{A_{2,a}}$ is the strategy S for choosing the action (i.e., dominance strategy).

The decision-making method manages the way agents act (simultaneously or sequentially) as well as the different types of actions (continuous or discrete). It is also necessary to ensure that the decision-making strategy can handle the nature of the actions (discrete or continuous) and how they are chosen (sequentially or simultaneously). As our framework allows us to easily change the decision-making method and strategy, we just have to select them according to the nature of the actions and how they are chosen. **Figure 2** summarizes our formalization of the decision-making process using a block diagram. In **Section 4**, we explain the selected decision-making method and strategy in our experiments as well as the performance metrics that can be taken into consideration.

4 APPROACH

To illustrate our contributions, we define a constant decision-making method M and strategy S. We assume as decision-making method the Perfect-Information Extensive Form (PIEF) of the game theory (environment and actions are known) in which the full flow of the game is displayed in the form of a tree. Using Nash Equilibrium as the strategy of the decision-making process ensures optimality regarding the choice of the actions, which is what we seek to guarantee.

4.1 Perfect-Information Extensive Form

As decision-making method M in **Eq. 6** we used the Perfect-Information Extensive Form (PIEF). Using this method, the agent has all the information about the actions and decisions of other agents and the environment. A game (or task or application) in PIEF in game theory is represented mathematically by the tuple $T = (\{\mathbf{N}\}, \{\mathbf{A}\}, \{\mathbf{H}\}, \{\mathbf{Z}\}, \boldsymbol{\chi}, \boldsymbol{\rho}, \boldsymbol{\sigma}, \{\vec{\mathbf{U}}\})$ (Leyton-Brown and Shoham, 2008), with:

- T represents the game (i.e., the task) as a tree (graph) structure.
- $\{\mathbf{N}\}$ is a set of n agents.
- $\{\mathbf{A}\}$ is a set of actions of all agents for all iterations.
- $\{\mathbf{H}\}$ is a set of non-terminal choice nodes. A non-terminal choice node represents an agent that chooses the actions to perform.
- $\{\mathbf{Z}\}$ is a set of terminal choice nodes; disjoint from $\{\mathbf{H}\}$. A terminal choice node represents the utility values attributed to the actions A_i^k each agent i chose in an alternative (i.e., a branch of the tree).
- $\boldsymbol{\chi}$: $\{\mathbf{H}\} \mapsto \{\mathbf{A}\}_{@H}$ is the action function, which assigns to each choice node H a set of possible actions $\{\mathbf{A}\}_{@H}$.
- $\boldsymbol{\rho}$: $\{\mathbf{H}\} \mapsto \{\mathbf{N}\}$ is the agent function, which assigns to each non-terminal choice node an agent $i \in \{\mathbf{N}\}$ who chooses an action in that node.
- $\boldsymbol{\sigma}$: $\{\mathbf{H}\} \times \{\mathbf{A}\} \mapsto \{\mathbf{H}\} \cup \{\mathbf{Z}\}$ is the successor function, which maps a choice node and an action to a new choice node or terminal node.
- $\{\vec{\mathbf{U}}\} = \{(\vec{\mathbf{U}})_A^1, \ldots, (\vec{\mathbf{U}})_A^k, \ldots, (\vec{\mathbf{U}})_A^{k_f}\}$ is the global utility profile for all iterations.

We apply this structure to represent the task in the following sections. In our case, since the number of nodes is small, $\boldsymbol{\chi}$, $\boldsymbol{\rho}$, and $\boldsymbol{\sigma}$ are straightforward functions (cf. **Figure 4**).

From a high-level perspective, a perfect-information game in extensive form is simply a tree (e.g., **Figure 4**) which consists of:

- Non-terminal nodes (squares): each square represents an agent that will choose actions.
- Arrows: each one represents a possible action (there are as many arrows as available actions $\{\mathbf{A}_i\}^k$ for agent i at iteration k).
- Terminal nodes (ellipses): each ellipse represents the utilities calculated for each action chosen by each agent in an alternative (i.e., a branch of the tree).

Note that this kind of tree is made for all the possible alternatives (considering all the actions an agent might choose) even if some of them will never happen (the agent will never choose some of the available actions). In this way, the tree represents all possible reactions of each agent to any alternative chosen by the others, even if, in the end, only one of these alternatives will really happen.

4.2 Subgame Perfect Nash Equilibrium

As decision-making method S in **Eq. 6** we used Nash Equilibrium (NE). The game T can be divided into subgames T^k at each iteration. In game theory (Leyton-Brown and Shoham, 2008), we consider a subgame of T (in PIEF game) rooted at node H as the restriction of T to the descendants of H. A Subgame Perfect Nash Equilibrium (SPNE) of T is all action profiles $(\vec{\mathbf{A}})^k$ such that for any subgame T^k of T, the restriction of $(\vec{\mathbf{A}})^k$ to T^k is a Nash Equilibrium of T^k.

Nash Equilibrium in pure strategy (game theory) at iteration k is reached when each agent i best responds to the others (denoted by $-i$). The Best Response (BR) at k is defined mathematically as:

$$A_i^{*k} \in BR(A_{-i}^k) \text{ iff } \forall A_i^k \in \{\mathbf{A}_i^k\}, U_{A_i^*|A_{-i}}^k \geq U_{A_i|A_{-i}}^k \tag{9}$$

Hence, NE will ultimately be expressed as follows: $(\vec{\mathbf{A}})_{opt}^k = (A_1^k, \ldots, A_i^k, \ldots, A_n^k)$ is an optimal profile of actions following Nash's equilibrium in pure strategy iff $\forall i, A_i^k \in BR(A_{-i}^k)$.

From a high-level perspective, to ensure that the actions chosen by one agent are following the NE strategy, it is

enough to verify that each agent chooses the actions that have the maximum possible utilities.

4.3 Performance Metrics

As long as a metric can be formulated mathematically or at least can be measured during the execution of the task and expressed as a condition to calculate the task reward, it can be considered in choosing the actions through the performance metrics $\{\mathbb{M}\}$ (some examples are given in **Table 1**). In the next section, we present the tests conducted, and we mention the chosen performance metrics for each scenario.

5 EXPERIMENTS

We conduct real and simulated tests to prove the effectiveness of our formalization. We test three different utility function case scenarios in which the reward values change according to the chosen performance metrics. In the state-of-the-art case scenario, no metric is optimized. In the real experimental tests, the time to completion metric is optimized. In the simulated tests, we optimize the time to completion by considering the probability of human errors and the time each agent takes to make an action.

5.1 The Task

We chose to solve Camelot Jr. as a task. To successfully complete this task, all the cubes must be positioned correctly to build a path between the two figures. We have divided the task completion process into iterations during which each agent chooses an action sequentially.

5.1.1 Experiments Context

We make the collaborative team ($\{\mathbf{N}\}$), composed of a human (h) and the humanoid robot Nao (r), do a task (T) that consists of building puzzles (cf. **Figure 1**). Nao is much slower than the human ($t_{A_r} > t_{A_h}$) in doing physical tasks (e.g., pick-and-place tasks), and we want to minimize the total task time ($\{\mathbb{M}\}$). This slowness depends on the nature of the robot itself (its motor capacity combined with the use of its camera) and the complexity of the puzzle. For the robot, the puzzle is more complex as the number of cubes to assemble increases. It is quite different for the human the complexity depends on their "intelligence" which means that the puzzle is easier as the human is "intelligent". By "intelligent", we mean that the human can discover rapidly and without making mistakes where the correct position of each cube is.

The advantage of collaborating with the robot is that it knows the solution to the construction task. Therefore, the robot is always performing well, even if it is slower than the human. The human agent, however, can make mistakes. The human begins to play, and then, it is the robot's turn. The robot will correct the human's move if this move is wrong. The changes in the robot's decision-making between the three case scenarios, including all the details we will present in the following sections, are shown in **Figure 6**. The implementation procedure and computation times for the conducted experiments are presented in the **Supplementary Material**.

5.1.2 Assumptions

To illustrate the contributions of this paper, we consider the following assumptions:

- The task is always achievable. We solve the task while optimizing the performance metrics through the utility function. The optimization of the metrics does not have an impact on the solvability of the task.
- We limit the number of agents to two: a human (h) and a robot (r). Hence, $\{\mathbf{N}\} = \{h, r\} \Rightarrow n = 2$.
- We limit agents to choose only one discrete action per iteration (i.e., $|A_i^k| = 1$) and to maximize only one metric (time to completion) in the real experiment and two metrics (time to completion by considering the probability of human errors) in the simulated experiments.
- The task is performed sequentially through iterations. An iteration k includes the human making an action, then the robot reacting.
- The agent set of actions and the time the agent takes to make an action are invariable by iteration.

5.1.3 The Actions

The set of human actions **Eq. 10** and the set of robot actions **Eq. 11** are the same at every iteration, and each one of them consists of three actions:

- $A_{h,g} \equiv A_{r,g}$: perform the good action (i.e., grasp a free cube and release it at the right place).
- $A_{h,w} \equiv A_{r,w}$: wait (i.e., the agent does nothing and passes its turn).
- $A_{h,b}$: perform the bad action (i.e., the human makes an error: grasping a free cube and releasing it at the wrong place).
- $A_{r,c}$: correct a bad action (i.e., the robot removes the cube from the wrong place).

$$\{\mathbf{A}_h\}^k = \{\mathbf{A}_h\} = \{A_{h,g}, A_{h,w}, A_{h,b}\} \tag{10}$$

$$\{\mathbf{A}_r\}^k = \{\mathbf{A}_r\} = \{A_{r,g}, A_{r,w}, A_{r,c}\} \tag{11}$$

5.1.4 Utility Calculation

The following equation is the adaptation of **Eq. 7** to the current task. So, the utility of every available action a for each agent i is calculated as follows:

$$U_{A_{i,a}}^k = U_{A_{i,a}} = \left(\frac{1}{t_{A_{i,a}}} \times G_{A_{i,a}} \times R_{A_{i,a}}\right) t \tag{12}$$

with:

- $t_{A_{i,a}}$: the duration of action a of agent i,
- t: the total time for an iteration ($t = \sum_{i=1}^{n} t_{A_{i,a}}$, here having $n = 2$, therefore $t = t_{A_{h,a}} + t_{A_{r,a'}}$),
- $G_{A_{i,a}}$: the constraint that ensures the task progression by penalizing the actions which make the task regress (cf. **Table 2**),
- and $R_{A_{i,a}}$: the reward of action a of agent i.

TABLE 2 | The value of the constraint of the task accomplishment for each action: making the task progress ($G_{A_{i,a}} = 1$), making no progression ($G_{A_{i,a}} = 0$), and making the task regress ($G_{A_{i,a}} = -1$).

	Action	$G_{A_{i,a}}$	Task progress
Human	$A_{h,g}$	1	Progression
	$A_{h,w}$	0	No progression
	$A_{h,b}$	− 1	Regression
Robot	$A_{r,g}$	1 if $A_h \neq A_{h,b}$ or − 1 otherwise	Progression
	$A_{r,w}$	0	No progression
	$A_{r,c}$	1 if $A_h = A_{h,b}$ or − 1 otherwise	Progression

5.1.5 Strategy of Action's Choice

In our formalization, we mentioned that the agents are choosing Nash Equilibrium (NE) as the decision-making strategy. But since the behavior (the decision-making strategy) of each human is different from one to another, we cannot claim that they will follow the NE for choosing their actions. For the robot, however, we restrict it to choose the actions by using the NE strategy. That is why the robot is choosing the action with the highest utility knowing the one chosen by the human. Note that, in our case scenarios, the robot reacts to the human's action since they are doing the task sequentially, and the human starts.

5.2 State-of-the-Art Utility Function

In state-of-the-art techniques, there is no optimization of the task. This is equivalent to always consider: $\forall a, i\ R_{A_{i,a}} = 1$ in our approach (in **Eq. 12**). For each iteration (each agent chooses an action with a utility), we can represent the task with the tree structure of **Figure 4A**. We will refer in the rest of the article to this case scenario by using C_1.

In this case, using NE, the robot's reaction to the human action will be as follows: $A_{r,g}$ if the human chose $A_{h,g}$ or $A_{h,w}$, and $A_{r,c}$ if the human chose $A_{h,b}$.

5.3 Real Experiments with Time Metric

We conducted tests[3] with a group of 20 volunteers. The objectives were to prove that the framework is applicable for a real task and to check human adaptation to the robot.

5.3.1 Experiment Procedure

After explaining the game rules to the participants, we asked them to complete two puzzles to make sure they understood the gameplay. Afterward, we asked each participant to complete three puzzles, chosen randomly among five, by collaborating with the Nao Robot.

The participant began the game. Then, it was Nao's turn. It continued until the puzzle was done. At each time, the participant had 20 s (t_{A_h}) to make an action or to decide to skip their turn. Nao takes on average 60 s (t_{A_r}) to do an action. It was skipping its turn when humans were well-doing and correcting them when they made an error. Nao did not move the cubes on its own (for human safety), but it was showing and telling the human which cube should be moved and where by pointing it. **Figure 3** illustrates, as an example, the steps of solving puzzle two by a participant and Nao.

5.3.2 Utility Function for Optimizing the Time

The reward values **Eq. 13** in the utility function **Eq. 12** ensure to maximize the time metric by penalising the action taken by the robot (the slower agent, i.e., $R_{A_{i,a}} = -1$) if the human (the faster agent denoted by i') chooses the correct action (denoted by a'). This penalization will prevent the robot from interfering with the human actions if the human makes the right decision:

$$R_{A_{i,a}} = \begin{cases} -1 & \text{if } G_{A_{i,a}} > 0 \text{ and } G_{A_{i',a'}} = 1 \text{ and } t_{A_{i',a'}} < t_{A_{i,a}} \\ 1 & \text{otherwise} \end{cases} \quad (13)$$

Thus, for each iteration, we can represent the task with the tree structure of **Figure 4B**. We will refer in the rest of the article to this case scenario using C_2. In this case, using NE, the robot's reaction to the human action will be as follows: $A_{r,w}$ if the human chose $A_{h,g}$, $A_{r,g}$ if the human chose $A_{h,w}$, and $A_{r,c}$ if the human chose $A_{h,b}$.

5.3.3 Results

Experiments with humans (presented in **Section 5.3.1**) were those where the robot used the utility function optimizing the time metric (case 2 (C_2)). It was very difficult to have enough participants to also test the case where the robot does not optimize any metric (the state-of-the-art case (C_1)). The only change in the procedure of the experiments using C_1 will be that even if the human is well-doing, the robot will not pass its turn ($A_{r,w}$) but will perform the good action ($A_{r,g}$). Hence, to compare the achieved results of our technique and the state-of-the-art techniques, we assumed that human actions remain the same in the case C_1 as in the case C_2, and we merely changed the robot reactions.

We chose to keep human actions unchanged between the two cases to ensure that only the switching of the utility function (C_2 to C_1) affects the robot reaction and not the influence of human behavior. **Table 3** provides an example of a scenario for solving puzzle two with C_2 and C_1 (**Figure 3**). We also calculated in **Figure 5** the average time and the standard deviation of the measured times among the experiments (C_2) and the deducted times (C_1).

In C_2, we assumed that if the human does the good action once, they will continue to do it each time. We notice from **Figure 5** that C_1 works better when the human is not "intelligent", i.e., they make lots of errors. That is why, the standard deviation values using C_2 are bigger than using C_1. This is the case for the last three puzzles where the average time using C_2 is bigger than using C_1. For the first puzzle, however, the average time using C_2 is smaller than using C_1, but the standard deviation values using C_2 are bigger than using C_1. The standard deviation values of this puzzle (using C_1 and C_2) are the biggest ones among all puzzles presented in **Figure 5**. Having big standard deviation values means that this puzzle was harder to solve for some participants and easier for others. That is why the average

[3]The experiment protocol was approved by the ethics committee of the Clermont-Auvergne University under the number: IRB00011540-2020-48.

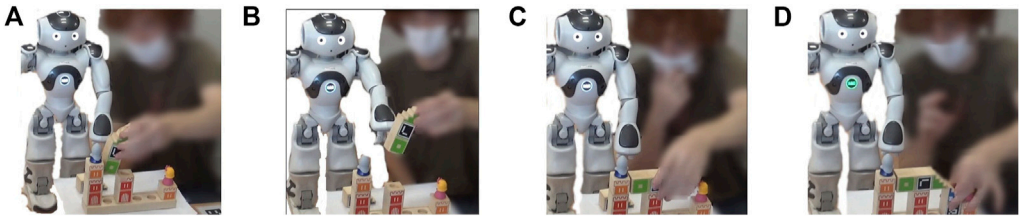


FIGURE 3 | Example of the solving steps of the puzzle two by a participant and Nao. **(A)** The human puts a cube in a wrong position. **(B)** Nao asks him to remove that cube. **(C)** The human puts a cube in a correct position, then the robot does nothing. **(D)** The human puts another cube in a correct position and the puzzle is solved.

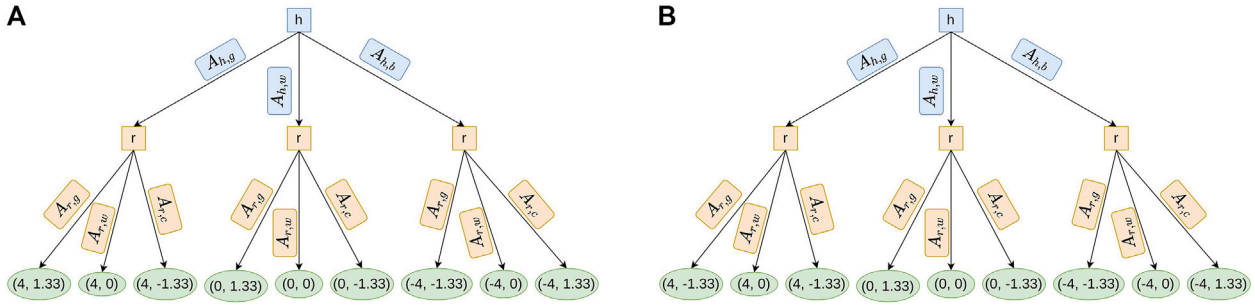


FIGURE 4 | Tree representation of the task based on the utility function in C_1 and C_2. Notice that the difference between both figures is the utility value of the action $A_{r,g}$, of the robot (1.33 and −1.33). It is because C_1 (on the contrary of C_2) does not minimize the time, so the robot continues to make an action even if the robot is slower than the well-performing human. **(A)** This tree is obtained by simulating an iteration of the task without optimization (C_1). The utilities (first for human agent and second for robot in green ellipses) are calculated for $t_{A_h} = 20\ s, t_{A_r} = 60\ s$ and $t = 80$ s. **(B)** This tree is obtained by simulating an iteration of the task optimized by the time metric (C_2). The utilities (first for human agent and second for robot in green ellipses) are calculated for $t_{A_h} = 20\ s, t_{A_r} = 60\ s$ and $t = 80$ s.

TABLE 3 | The adaptation of time calculation from C_2 to C_1 for the resolution of one scenario of puzzle two.

		Iteration 1	Iteration 2	Iteration 3	Total time (s)
C_1	Human actions	$A_{h,b}$ (20s)	$A_{h,g}$ (20s)		160
	Robot reactions	$A_{r,c}$ (60s)	$A_{r,g}$ (60s)		
	Iteration time	80 s	80 s		
C_2	Human actions	$A_{h,b}$ (20s)	$A_{h,g}$ (20s)	$A_{h,g}$ (20s)	140
	Robot reactions	$A_{r,c}$ (60s)	$A_{r,w}$ (20s)		
	Iteration time	80s	40s	20s	

time and the standard deviation values using C_2 and C_1 do not have the same trend.

On the contrary, C_2 performs better when the human is "intelligent". Therefore, the time taken to accomplish the task depends on human "intelligence" that is related to the probability of human errors and the ratio between the time each agent takes to do an action. Without taking into account these two additional metrics, we cannot optimally ensure to minimize the time to completion when the human makes many mistakes.

In the next case (C_3), we present a third utility function that takes into account the time taken by the agents to make an action and optimizes the time to completion by encouraging the human agent to reduce the number of errors. Each metric has the same weight $\epsilon = 1$ (**Eq. 7**) since all these metrics are compatible. It means that optimizing one metric depends on optimizing the others.

5.4 Simulated Experiments with Time and Number of Human Errors Metrics

We use case (C_3) to prove that our framework can handle the changes in the performance metrics from one case scenario to another. In this case (C_3), we select between C_1 and C_2 the case that minimizes the total time by considering the probability of human errors and the ratio between the time each agent takes to make an action. The difference between C_1 and C_2 lies in the robot reaction when the human agent makes the good action ($A_{h,g}$). With C_1, the robot makes the good action ($A_{r,g}$), while with C_2, the robot decides to wait ($A_{r,w}$), to not slow down the human. **Figure 6** presents an algorithmic block diagram showing which case the robot will choose to make an action.

5.4.1 Assumptions on Humans

We did not have enough participants to do real tests so we chose to do simulated tests. For this, we simulated the human decision process as a probability distribution among the set of feasible actions such that: $P(A_{h,g}) = I_1$, $P(A_{h,w}) = I_2$, and $P(A_{h,b}) = I_3 = 1 - (I_1 + I_2)$. I_1, I_2, and I_3 are variable from one participant to another and $0 < I_1 + I_2 \leq 1$.

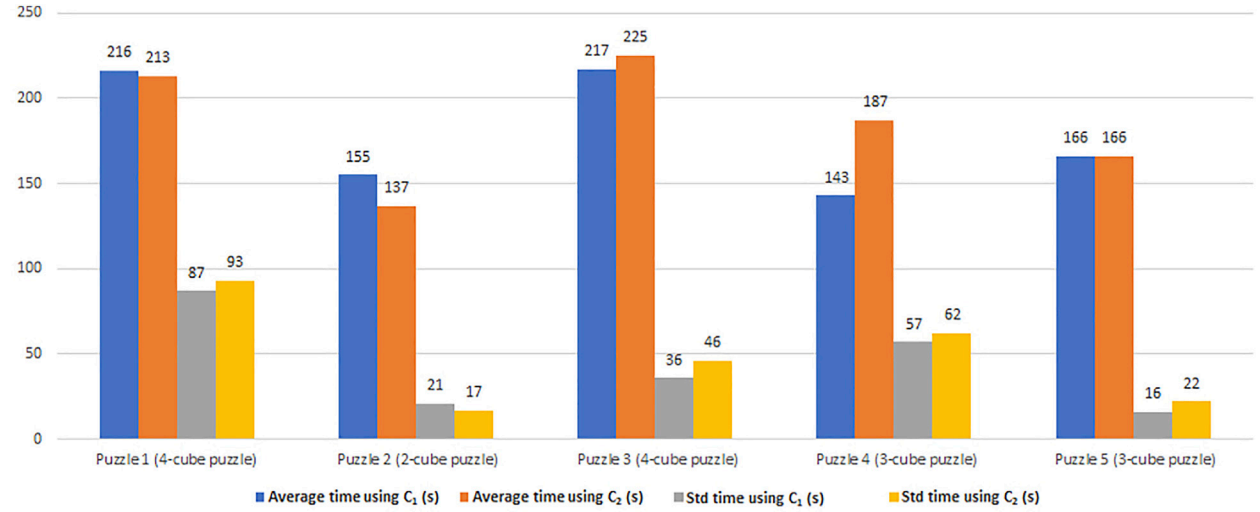


FIGURE 5 | The average time and the standard deviation in seconds of the time taken to do the task with the state-of-the-art utility function (C_1) and the utility function used to optimize the time (C_2), which is our contribution.

5.4.2 Utility Function for Optimizing the Time to Completion While Considering the Probability of Human Errors

Compares to **Eq. 12**, only the reward values ($R_{A_{i,a}}$) change. The reward values of the utility function for C_3 are calculated by the following function:

$$R_{A_{i,a}} = \begin{cases} -1 & \text{if } G_{A_{i,a}} > 0 \text{ and } G_{A_{i',a'}} = 1 \text{ and } t_{C_2} < t_{C_1} \\ 1 & \text{otherwise} \end{cases} \quad (14)$$

Where $t_{C_2} < t_{C_1}$ decides which case (1 or 2) is the best to optimize the total time (cf. **Figure 6**) and thus reduce the number of human errors. So, if $t_{C_2} < t_{C_1}$ is true, C_2 will be faster than C_1, and vice versa. t_C is the generic equation for calculating time payoffs t_{C_1} and t_{C_2} **Eq. 15**. It considers the probability that the human agent will perform each feasible action ($P(A_{i'})$ = probability distribution of human actions) which we assume as known and the time that the agents will take to make an action. $t_{A_{i',a'}}$ is the time required for the other agent i' (i.e., human) to make the chosen action a' and $t_{A_{i,a}}$ is the time taken by the agent i (i.e., robot) to react by making the action a. N_c is the number of cubes correctly placed by taking actions a and a'.

C_2 did not work well because it was assuming that if the human does the good action once, they will continue to do it each time. That is why in **Eq. 13** the comparison of the times ($t_{A_{i',a'}} < t_{A_{i,a}}$) was not including the probability of the human actions (including probability of making errors). In case the human often performs the bad action (e.g., $I_3 \geq 0.6$), the robot is encouraged not to wait but to perform the good action (C_1), despite its slowness. This is done to reduce the number of iterations and thus reduce the number of times the human will make a mistake, as they will have fewer turns to play (i.e., reducing the number of human errors). That is why in C_3, we consider the probability distribution of human actions, including that of doing the bad action (committing errors) while calculating t_c (cf. **Eq. 14**). The robot chooses C_1 if the human will make many errors and C_2 in the opposite case.

$$t_C = \frac{\sum_{a'=1}^{l} P(A_{i',a'})(t_{A_{i',a'}} + t_{A_{i,a}})}{\sum_{a'=1}^{l} P(A_{i',a'})N_c} \quad (15)$$

5.4.3 Simulation Conditions

A simulated test depends on:

- The values of I_1 and I_2 (we tested for $I_1 = (0 : 0.1 : 1)$ and $I_2 = (0 : 0.1 : 1)$ except for $I_1 = I_2 = 0$).
- The ratio between t_{A_h} and t_{A_r} (we tested for 1/1, 1/1.5, 1/2, 1/3, 1/4, 1/5).
- The number of cubes required to solve the puzzle (we tested for 2, 3, 4, and 5).
- The number of simulations (10000) we conducted to calculate the average time and the standard deviation.

5.4.4 Simulation Results

We illustrate the efficiency of our utility function C_3 by showing the improvement in time to completion and the reduction of the number of human errors obtained while solving the puzzles.

5.4.4.1 Time Improvement

We validate the efficiency of our utility function C_3 by comparing the resulted average total times with similar cases using C_1 over 10000 simulations[4]. Like real experiments, we assumed that human actions are constant, and we change merely the robot actions.

[4]All the results are presented on https://github.com/MelodieDANIEL/Optimizing_Human_Robot_Collaboration_Frontiers.

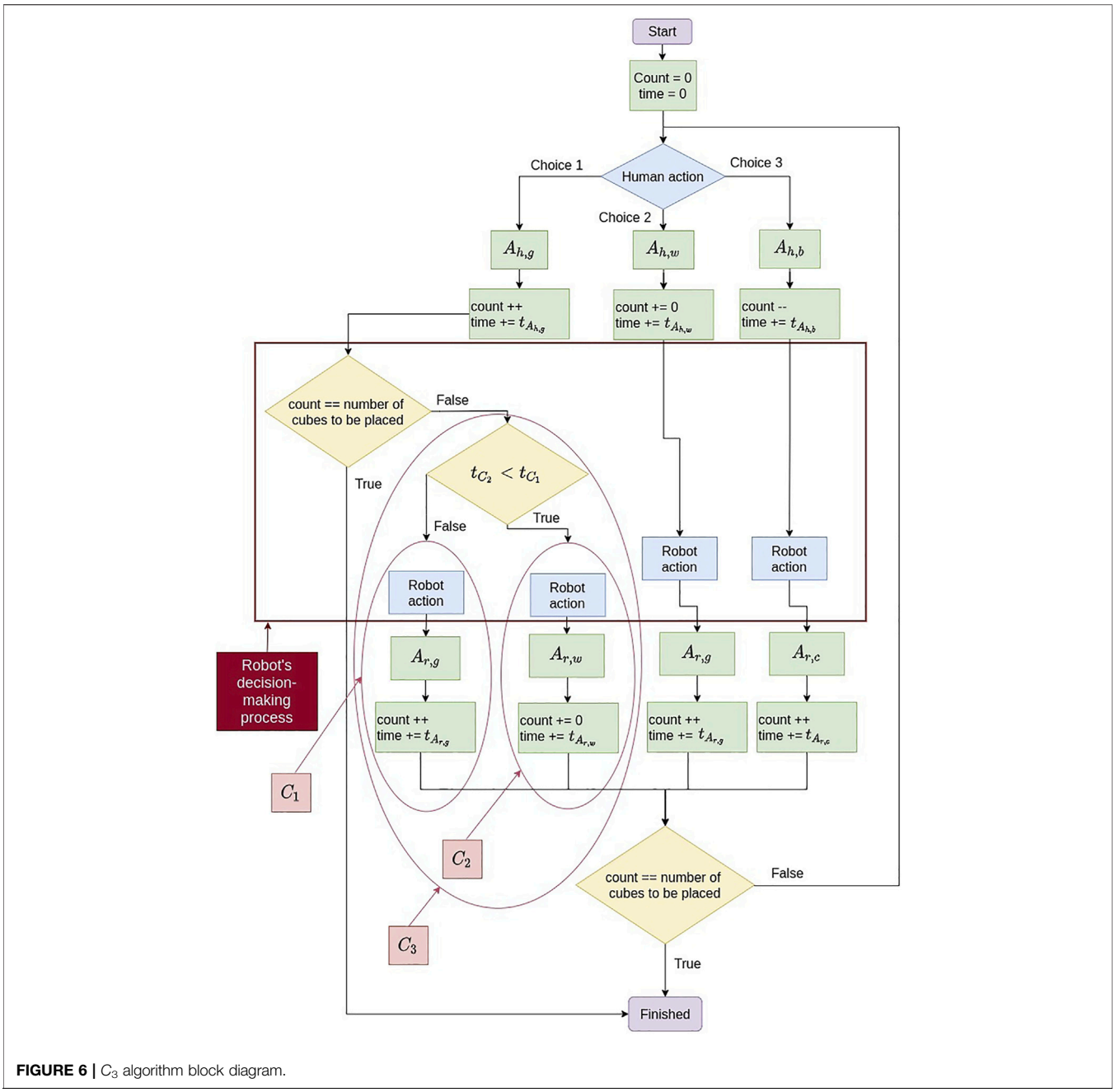


FIGURE 6 | C_3 algorithm block diagram.

We calculate the time improvement **Eq. 16** by comparing the average total times ($\bar{D}_{C_3}$) calculated using the utility function of C_3 to the average total times ($\bar{D}_{C_1}$) calculated using the state-of-the-art utility function (i.e., C_1). This is illustrated in **Figure 7** for a 4-cube puzzle with a ratio $t_{A_h}/t_{A_r} = 1/5$. As it can be observed, the experiment times are improved up to 66.7%. Another example is given in **Supplementary Material** for a 3-cube puzzle with a ratio $t_{A_h}/t_{A_r} = 1/3$. The percentage of the time improvement depends on how much the human participant is "intelligent".

$$\text{Percentage of time improvement} = \frac{\bar{D}_{C_1} - \bar{D}_{C_3}}{\bar{D}_{C_1}} * 100 \qquad (16)$$

Theoretically, however, this percentage can reach a value close to 100% for a very small time taken by the human (which lead to a very small $\bar{D}_{C_3}$) and a very big time taken by the robot (which lead to a very big $\bar{D}_{C_1}$). We can note that having the time improvement percentage equal to 0 signifies that we are using C_1; while utilizing C_2 increases the value of the time improvement percentage. It means that in the worst-case scenario, the efficiency of our method is as the state-of-the-art peers.

5.4.4.2 Reduction of the Number of Human Errors

For reducing the time to completion, we consider the probability of human errors in **Eq. 15**. So, we choose between C_1 and C_2 the

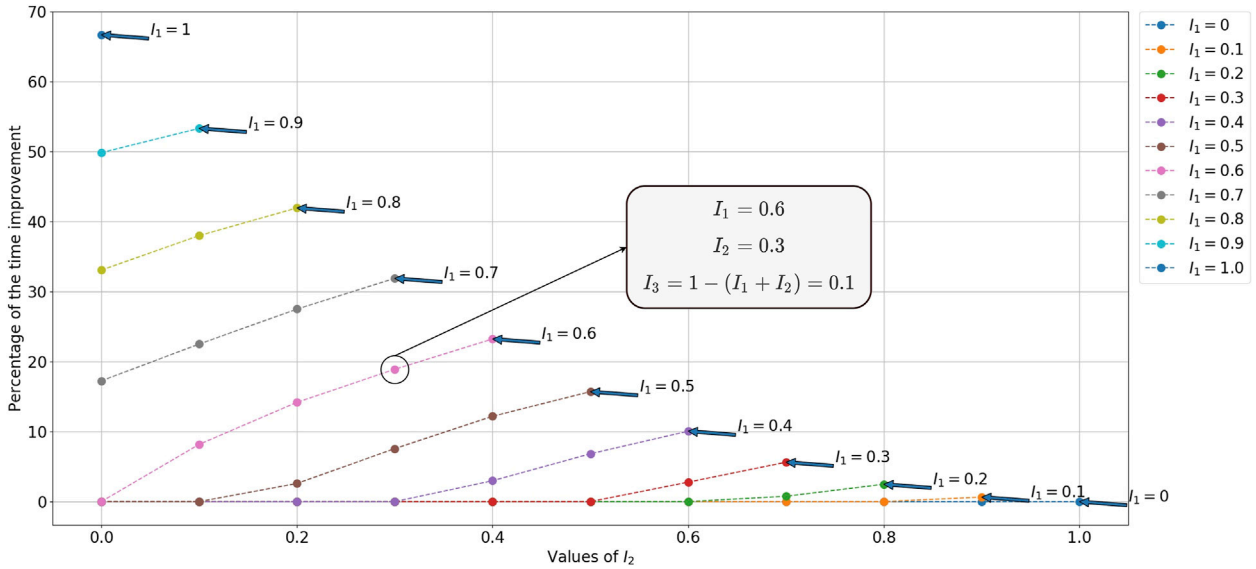


FIGURE 7 | Percentage of time improvement between C_3 and C_1 for a 4-cube puzzle. $t_{A_h} = \{15, 0, 15\}$ and $t_{A_r} = \{75, 0, 75\}$, so the ratio $t_{A_h}/t_{A_r} = 1/5$. $P(A_{h,g}) = I_1$, $P(A_{h,w}) = I_2$, and $P(A_{h,b}) = I_3 = 1 - (I_1 + I_2)$. In this figure, each dotted line is equivalent to a specific I_1 value. Each dot corresponds to a I_2 value (read on the x-axis). For each dot knowing I_1 and I_2, we can deduce its I_3 value using $I_3 = 1 - (I_1 + I_2)$. For illustrating, we give I_1, I_2, and I_3 values of the dot marked in the figure.

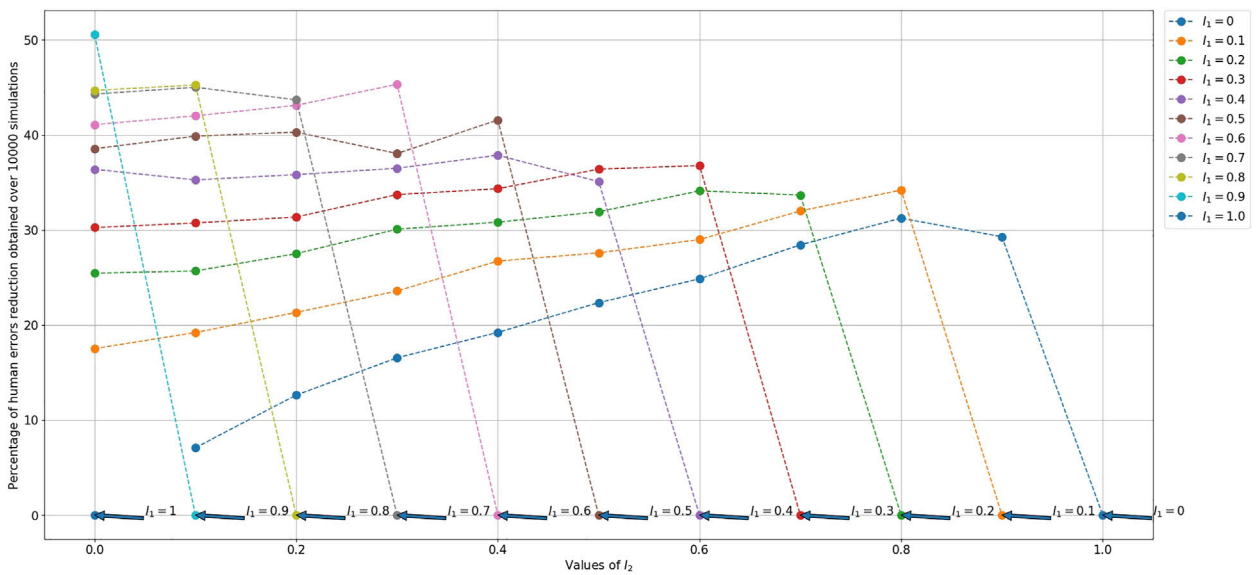


FIGURE 8 | Percentage of human errors reduction between the predicted probability of human errors and the measured one for a 2-cube puzzle. $t_{A_h} = \{15, 0, 15\}$ and $t_{A_r} = \{15, 0, 15\}$, so the ratio $t_{A_h}/t_{A_r} = 1/1$. $P(A_{h,g}) = I_1$, $P(A_{h,w}) = I_2$, and $P(A_{h,b}) = I_3 = 1 - (I_1 + I_2)$. In this figure, each dotted line is equivalent to a specific I_1 value. Each dot corresponds to a I_2 value (read on the x-axis). For each dot knowing I_1 and I_2, we can deduce its I_3 value using $I_3 = 1 - (I_1 + I_2)$.

case which minimizes the time by reducing the number of iterations needed for solving the puzzle. This means choosing the case which reduces the number of human errors as explained in **Section 5.4.2**. We calculate, in **Eq. 17**, the percentage of human errors reduction (PHER) using the difference between the predicted probability of human errors I_3 and the average (over the 10000 simulations) measured probability of human errors $\frac{N_{h_e}}{N_{h_a}}$.

$$\text{Percentage of human errors reduction} = \begin{cases} \left(\frac{I_3 - \left(\frac{\overline{N_{h_e}}}{N_{h_a}} \right)}{I_3} \right) * 100 & \text{if } I_3 > 0 \\ 0 & \text{if } I_3 = 0 \end{cases} \quad (17)$$

Where I_3 is the predicted probability that the human makes a wrong move (makes an error), N_{h_e} the measured number of human errors, and N_{h_a} the measured total number of human actions. So, $\frac{N_{h_e}}{N_{h_a}}$ will be the measured probability that the human makes an error after one simulation. The reduction of the number of human errors is as big as $\frac{\overline{N_{h_e}}}{N_{h_a}}$ is small.

The reduction percentage of the number of human errors increases with the reduction of t_{A_r} (the time the robot takes to make an action) and the reduction of the number of cubes that should be assembled to solve a puzzle. In other words, the human will have fewer turns to play and so fewer chances to make mistakes. The best result we got is presented in **Figure 8** (for a 2-cube puzzle with $t_{A_h} = t_{A_r}$): the reduction percentage of the number of human errors is up to 50.6%. The result can be better in case that the robot is faster than the human in performing an action ($t_{A_r} < t_{A_h}$). Note that, when the I_3 is equal to 0%, the percentage of human errors reduction is also equal to 0%. It means that the human never makes errors so, there is nothing that needs to be improved. Another example is presented in **Supplementary Material** for a 3-cube puzzle with a ratio $t_{A_h}/t_{A_r} = 1/3$.

6 CONCLUSION AND FUTURE WORK

We propose a new formalization of the decision-making process to perform the task and accomplish it more efficiently. We assess through the experiments that our formalization can be applied to feasible tasks and optimize the human-robot collaboration in terms of all defined metrics. We also prove through the experiments that we can change the three studied case scenarios by changing the performance metrics in the utility function (i.e., reward function) without changing the entire framework.

Validating this, experiments are carried out by simulating the task of solving the construction puzzle. It shows that using our proposed utility function instead of the state-of-the-art utility function improves the experiment time up to 66.7%, hence improves the human-robot collaboration without extending the robot's abilities. Theoretically, this improvement can reach a value close to 100%. We also got a percentage of human errors reduction up to 50.6% by considering the predicted probability that the human makes errors for optimizing the time to completion.

We note that there are still some points to improve in future work. First, we want to add to the formalization a predictive function to estimate human behavior through a realistic database that can be used in a reinforcement learning procedure. Secondly, we set in this paper the decision-making method and the strategy. We want to develop another formalization in which they will be variable and dynamically adaptable to the task.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/MelodieDANIEL/Optimizing_Human_Robot_Collaboration_Frontiers.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by The ethics committee of the Clermont-Auvergne University under the number: IRB00011540-2020-48. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

MH designed and implemented the core algorithm presented in the paper and carried out the experiments on the Nao humanoid robot. SL, JC, and YM contributed to the presented ideas and to the review of the final manuscript.

FUNDING

This work has received funding from the Auvergne-Rhône-Alpes Region through the ATTRIHUM project and from the European Union's Horizon 2020 research and innovation programme under grant agreement No 869855 (Project "SoftManBot").

ACKNOWLEDGMENTS

We would like to thank the European Union's Horizon 2020 research and innovation programme under grant agreement No

869855 (Project "SoftManBot") for funding this work. We would like to thank Sayed Mohammadreza Shetab Bushehri and Miguel Aranda for providing feedback and English editing on the previous versions of this manuscript and for giving us valuable advice.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2021.736644/full#supplementary-material

REFERENCES

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., and Khatib, O. (2018). Progress and Prospects of the Human-Robot Collaboration. *Auton. Robot* 42, 957–975. doi:10.1007/s10514-017-9677-2

Bansal, S., Xu, J., Howard, A., and Isbell, C. (2020). *A Bayesian Framework for Nash Equilibrium Inference in Human-Robot Parallel Play*. Corvalis, OR: arXiv. preprint arXiv:2006.05729.

Bütepage, J., and Kragic, D. (2017). *Human-robot Collaboration: From Psychology to Social Robotics*. arXiv. preprint arXiv:1705.10146.

Chen, M., Nikolaidis, S., Soh, H., Hsu, D., and Srinivasa, S. (2020). Trust-Aware Decision Making for Human-Robot Collaboration. *J. Hum. Robot Interact.* 9, 1–23. doi:10.1145/3359616

Clabaugh, C., Mahajan, K., Jain, S., Pakkar, R., Becerra, D., Shi, Z., et al. (2019). Long-term Personalization of an in-home Socially Assistive Robot for Children with Autism Spectrum Disorders. *Front. Robot. AI.* 6, 110. doi:10.3389/frobt.2019.00110

Conitzer, V., and Sandholm, T. (2006). "Computing the Optimal Strategy to Commit to," in Proceedings of the 7th ACM conference on Electronic commerce, Ann Arbor, MI, June 11–15, 82–90. doi:10.1145/1134707.1134717

Delleman, N. J., and Dul, J. (2007). International Standards on Working Postures and Movements ISO 11226 and EN 1005-4. *Ergonomics* 50, 1809–1819. doi:10.1080/00140130701674430

DelPreto, J., and Rus, D. (2019). "Sharing the Load: Human-Robot Team Lifting Using Muscle Activity," in 2019 International Conference on Robotics and Automation, Montreal, QC, May 20–24 (ICRA), 7906–7912. doi:10.1109/ICRA.2019.8794414

Durantin, G., Heath, S., and Wiles, J. (2017). Social Moments: A Perspective on Interaction for Social Robotics. *Front. Robot. AI.* 4, 24. doi:10.3389/frobt.2017.00024

Fishman, A., Paxton, C., Yang, W., Ratliff, N., Fox, D., and Boots, B. (2019). "Collaborative Interaction Models for Optimized Human-Robot Teamwork, "in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, October 24–January 24, 11221–11228. *preprint arXiv: 1910.04339*.

Flad, M., Otten, J., Schwab, S., and Hohmann, S. (2014). "Steering Driver Assistance System: A Systematic Cooperative Shared Control Design Approach," in 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (IEEE), San Diego, CA, October 5–8, 3585–3592. doi:10.1109/smc.2014.6974486

Fülöp, J. (2005). "Introduction to Decision Making Methods," in *BDEI-3 Workshop*. Washington, 1–15.

Gabler, V., Stahl, T., Huber, G., Oguz, O., and Wollherr, D. (2017). "A Game-Theoretic Approach for Adaptive Action Selection in Close Proximity Human-Robot-Collaboration," in In 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, May 29–June 3 (IEEE), 2897–2903. doi:10.1109/icra.2017.7989336

Gervasi, R., Mastrogiacomo, L., and Franceschini, F. (2020). A Conceptual Framework to Evaluate Human-Robot Collaboration. *Int. J. Adv. Manuf Technol.* 108, 841–865. doi:10.1007/s00170-020-05363-1

Ghadirzadeh, A., Bütepage, J., Maki, A., Kragic, D., and Björkman, M. (2016). "A Sensorimotor Reinforcement Learning Framework for Physical Human-Robot Interaction," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, October 9–14 (IEEE), 2682–2688. doi:10.1109/iros.2016.7759417

Hoffman, G. (2019). Evaluating Fluency in Human-Robot Collaboration. *IEEE Trans. Human-mach. Syst.* 49, 209–218. doi:10.1109/thms.2019.2904558

Hosseini, S. M. F., Lettinga, D., Vasey, E., Zheng, Z., Jeon, M., Park, C. H., et al. (2017). "Both "look and Feel" Matter: Essential Factors for Robotic Companionship," in 2017 26th IEEE International Symposium on Robot and Human Interactive Communication, Lisbon, Portugal, August 28–September 1 (RO-MAN IEEE), 150–155. doi:10.1109/roman.2017.8172294

Jarrassé, N., Charalambous, T., and Burdet, E. (2012). A Framework to Describe, Analyze and Generate Interactive Motor Behaviors. *PloS one* 7, e49945. doi:10.1371/journal.pone.0049945

Kwon, M., Li, M., Bucquet, A., and Sadigh, D. (2019). "Influencing Leading and Following in Human-Robot Teams," in Proceedings of Robotics: Science and Systems FreiburgimBreisgau, Germany, June 22–26. doi:10.15607/rss.2019.xv.075

Leyton-Brown, K., and Shoham, Y. (2008). Essentials of Game Theory: A Concise Multidisciplinary Introduction. *Synth. Lectures Artif. Intelligence Machine Learn.* 2, 1–88. doi:10.2200/s00108ed1v01y200802aim003

Li, N., Oyler, D. W., Zhang, M., Yildiz, Y., Kolmanovsky, I., and Girard, A. R. (2017). Game Theoretic Modeling of Driver and Vehicle Interactions for Verification and Validation of Autonomous Vehicle Control Systems. *IEEE Trans. Control Syst. Technol.* 26, 1782–1797.

Li, Y., Carboni, G., Gonzalez, F., Campolo, D., and Burdet, E. (2019). Differential Game Theory for Versatile Physical Human-Robot Interaction. *Nat. Mach Intell.* 1, 36–43. doi:10.1038/s42256-018-0010-3

Liu, Z., Liu, Q., Xu, W., Zhou, Z., and Pham, D. T. (2018). Human-robot Collaborative Manufacturing Using Cooperative Game: Framework and Implementation. *Proced. CIRP.* 72, 87–92. doi:10.1016/j.procir.2018.03.172

Malik, A. A., and Bilberg, A. (2019). Complexity-based Task Allocation in Human-Robot Collaborative Assembly. *Ind. Robot: Int. J. robotics Res. Appl.* 46, 471–180. doi:10.1108/ir-11-2018-0231

Maurtua, I., Ibarguren, A., Kildal, J., Susperregi, L., and Sierra, B. (2017). Human–robot Collaboration in Industrial Applications: Safety, Interaction and Trust. *Int. J. Adv. Robotic Syst.* 14, 1729881417716010. doi:10.1177/1729881417716010

Na, X., and Cole, D. (2014). Game Theoretic Modelling of a Human Driver's Steering Interaction with Vehicle Active Steering Collision Avoidance System. *IEEE Trans. on Human-Machine Sys.* 45, 25–38.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2017). "Bridging the gap between Value and Policy Based Reinforcement Learning," in 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA.

Negulescu, O.-H. (2014). Using a Decision Making Process Model in Strategic Management. *Rev. Gen. Manage.* 19.

Nelles, J., Kwee-Meier, S. T., and Mertens, A. (2018). "Evaluation Metrics Regarding Human Well-Being and System Performance in Human-Robot Interaction - A Literature Review," in Congress of the International Ergonomics Association, Florence, Italy, August 26–30 (Springer), 124–135. doi:10.1007/978-3-319-96068-5_14

Nikolaidis, S., Hsu, D., and Srinivasa, S. (2017a). Human-robot Mutual Adaptation in Collaborative Tasks: Models and Experiments. *Int. J. Robotics Res.* 36, 618–634. doi:10.1177/0278364917690593

Nikolaidis, S., Nath, S., Procaccia, A. D., and Srinivasa, S. (2017b). "Game-theoretic Modeling of Human Adaptation in Human-Robot Collaboration," in Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17, Vienna, Austria, March 6–9 (ACM Press). doi:10.1145/2909824.3020253

Nocentini, O., Fiorini, L., Acerbi, G., Sorrentino, A., Mancioppi, G., and Cavallo, F. (2019). A Survey of Behavioral Models for Social Robots. *Robotics* 8, 54. doi:10.3390/robotics8030054

Oliff, H., Liu, Y., Kumar, M., Williams, M., and Ryan, M. (2020). Reinforcement Learning for Facilitating Human-Robot-Interaction in Manufacturing. *J. Manufacturing Syst.* 56, 326–340. doi:10.1016/j.jmsy.2020.06.018

Reinhardt, J., Pereira, A., Beckert, D., and Bengler, K. (2017). "Dominance and Movement Cues of Robot Motion: A User Study on Trust and Predictability," in 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, October 5–8 (IEEE). doi:10.1109/smc.2017.8122825

Rosenberg-Kima, R., Koren, Y., Yachini, M., and Gordon, G. (2019). "Human-robot-collaboration (Hrc): Social Robots as Teaching Assistants for Training Activities in Small Groups," in 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Daegu, South Korea, March 11–14 (IEEE), 522–523. doi:10.1109/hri.2019.8673103

Roveda, L., Magni, M., Cantoni, M., Piga, D., and Bucca, G. (2021). Human-robot Collaboration in Sensorless Assembly Task Learning Enhanced by Uncertainties Adaptation via Bayesian Optimization. *Robotics Autonomous Syst.* 136, 103711. doi:10.1016/j.robot.2020.103711

Seel, N. M. (2012). *Encyclopedia of the Sciences of Learning* (Boston, MA:Springer US). doi:10.1007/978-1-4419-1428-6

Sharkawy, A.-N., Papakonstantinou, C., Papakostopoulos, V., Moulianitis, V. C., and Aspragathos, N. (2020). Task Location for High Performance Human-Robot Collaboration. *J. Intell. Robotic Syst.* 100, 1–20. doi:10.1007/s10846-020-01181-5

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., et al. (2006). "Common Metrics for Human-Robot Interaction," in Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, Salt Lake City, UT, March 2–3, 33–40. doi:10.1145/1121241.1121249

Tabrez, A., and Hayes, B. (2019). "Improving Human-Robot Interaction through Explainable Reinforcement Learning," in 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Daegu, South Korea, March 11–14 (IEEE), 751–753. doi:10.1109/hri.2019.8673198

Tanevska, A., Rea, F., Sandini, G., Cañamero, L., and Sciutti, A. (2020). A Socially Adaptable Framework for Human-Robot Interaction. *Front. Robot. AI* 7, 121. doi:10.3389/frobt.2020.00121

Wagner-Hartl, V., Gehring, T., Kopp, J., Link, R., Machill, A., Pottin, D., Zitz, A., and Gunser, V. E. (2020). "Who Would Let a Robot Take Care of Them? - Gender and Age Differences," in International Conference on Human-Computer Interaction, Copenhagen, Denmark, July 19–24, 2020 (Springer), 196–202. doi:10.1007/978-3-030-50726-8_26

Weitschat, R., and Aschemann, H. (2018). Safe and Efficient Human-Robot Collaboration Part II: Optimal Generalized Human-In-The-Loop Real-Time Motion Generation. *IEEE Robot. Autom. Lett.* 3, 3781–3788. doi:10.1109/lra.2018.2856531

Xu, W., Tang, Q., Liu, J., Liu, Z., Zhou, Z., and Pham, D. T. (2020). Disassembly Sequence Planning Using Discrete Bees Algorithm for Human-Robot Collaboration in Remanufacturing. *Robotics. Computer-Integrated Manufacturing* 62, 101860. doi:10.1016/j.rcim.2019.101860

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Hani Daniel Zakaria, Lengagne, Corrales Ramón and Mezouar. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Supplementary Material

1 PERFORMANCE METRICS

We present, in Table S1, the performance metrics used for the evaluation of the Human-Robot Collaboration (HRC) classified based on the task types. We define each metric according to its usage in the different task types. We also introduce some common metrics used for evaluating HRC in general rather than the performance of a specific task.

2 IMPLEMENTATION OF THE CONDUCTED EXPERIMENTS

In figure S1, we present a block diagram of our implementation of the conducted experiments. First, the participant finishes their turn. Then, it is the turn of the robot which will be mainly made up of three parts:

- Perception: Nao's camera captures the markers, then using the Aruco library (Garrido-Jurado et al., 2014), we can estimate the pose of the cubes. This allows us to compute the state of the puzzle and to identify the human action.
- Decision-making process: The robot chooses its action according to the human's one. The utility function calculates a utility for each of the robot's actions. The robot chooses, based on the Nash equilibrium, the action that has the highest utility. This is illustrated in Figure 6 of the main paper.
- Robot's action: If the robot passes its turn ($A_{r,w}$), it tells the human. In this case, the robot does not have to make any movement. On the contrary, if the robot corrects the human's action ($A_{r,c}$) or helps them by indicating where to place a cube ($A_{r,g}$), the robot will have to speak and move its arm to point out the cube to move.

3 ENTIRE EXAMPLE OF A SIMULATED TEST ON THE ASSEMBLY TASK

In the main paper, we presented the best-simulated results for illustrating the percentage of the time improvement and the percentage of the reduction of the number of human errors. In this section, we want to present both results for the same simulated test as an example. We consider a 3-cube puzzle with a ratio of the time taken by each agent (the human h and the robot r) to make an action equals to: $t_{A_h}/t_{A_r} = 1/3$. This ratio is the same as the one we had while doing the real experiment with Nao and a human participant. We define $P(A_{h,g}) = I_1$ as the probability that the human does the good action, $P(A_{h,w}) = I_2$ as the probability that the human is passing their turn and, $I_3 = 1 - (I_1 + I_2)$ as the probability that the human makes an error.

We note from Figure S2 that the percentage of the time improvement using our utility function (C_3) instead of the state-of-the-art one (C_1) for this puzzle is up to 40%. From Figure S3, the percentage of human errors reduction for the same puzzle is up to 27.9%. In both figures, each dotted line is equivalent to a specific I_1 value. Each dot corresponds to a I_2 value (read on the x-axis). For each dot knowing I_1 and I_2, we can deduce its I_3 value using $I_3 = 1 - (I_1 + I_2)$.

We have performed a lot of simulated tests, the results of which can be found on: `https://github.com/MelodieDANIEL/Optimizing_Human_Robot_Collaboration_Frontiers`.

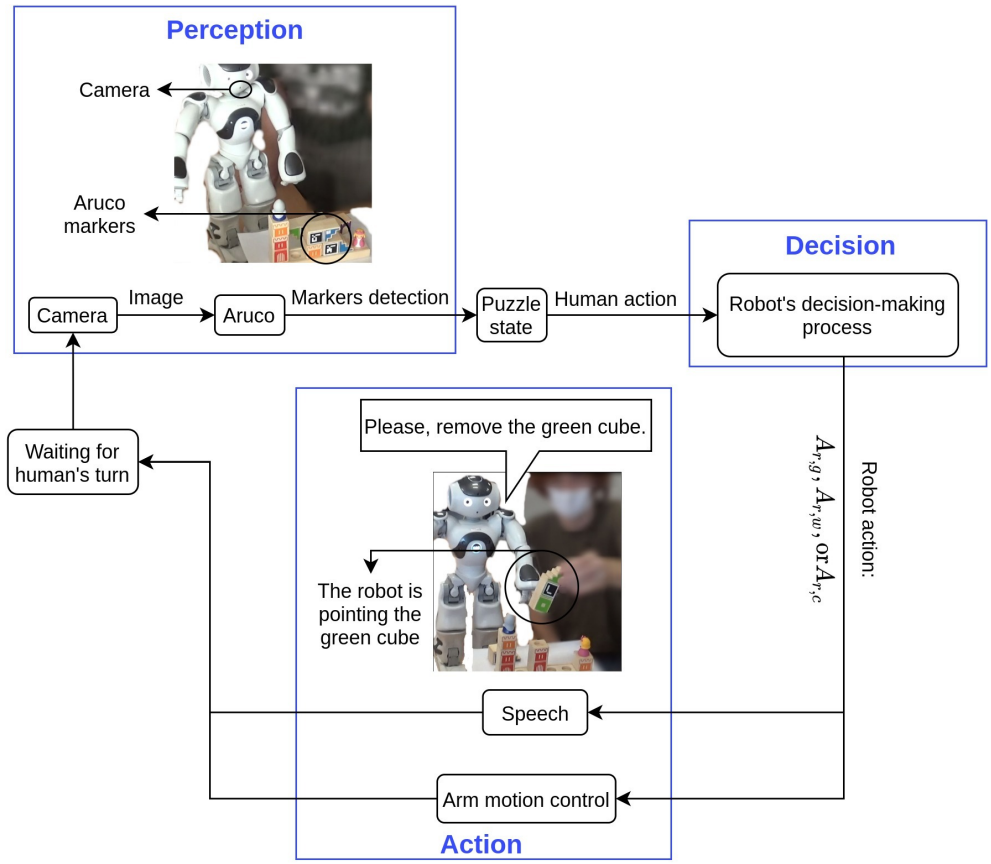


Figure S1. Implementation of the conducted experiments using ROS

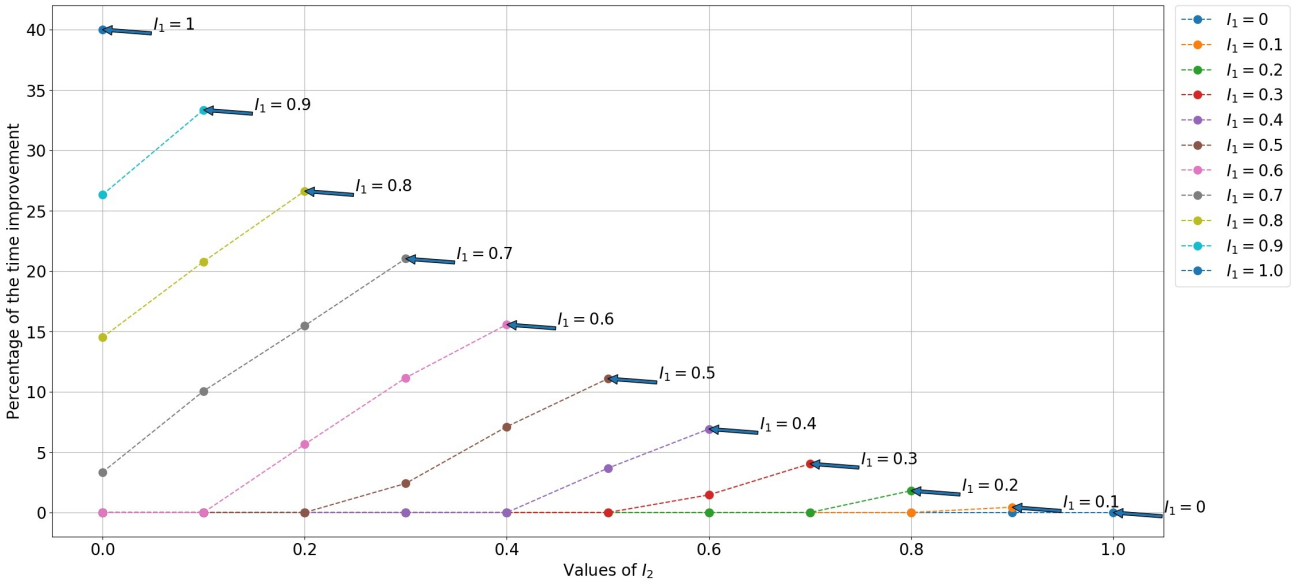


Figure S2. Percentage of time improvement between C_3 and C_1 for a 3-cube puzzle. $t_{A_h} = \{15, 0, 15\}$ and $t_{A_r} = \{45, 0, 45\}$, so the ratio $t_{A_h}/t_{A_r} = 1/3$.

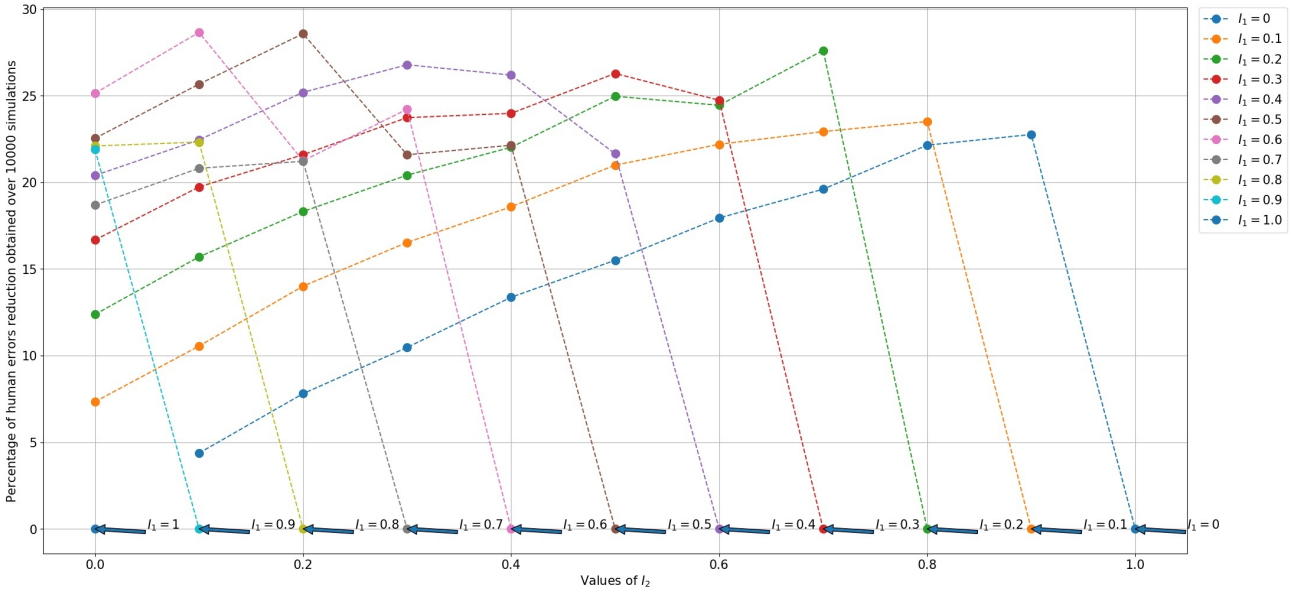


Figure S3. Percentage of human errors reduction between the predicted probability of human errors and the measured one for a 3-cube puzzle. $t_{A_h} = \{15, 0, 15\}$ and $t_{A_r} = \{45, 0, 45\}$, so the ratio $t_{A_h}/t_{A_r} = 1/3$.

4 RESULTED TABLE OF SIMULATION TESTS

In this section, we present the resulted table (Table S2) of the percentage of the time improvement and the reduction of the number of human errors for all the figures presented in the main paper and the supplementary material.

5 COMPUTATION AND EXECUTION TIME OF THE TESTS

Table S3 presents all the computation and execution times of the experiments in real and in simulation. As we can notice, the average computation time of our decision-making framework is 0.5s. This computation time is suitable for the targeted real tasks on which we want to apply this framework.

REFERENCES

Bütepage, J. and Kragic, D. (2017). Human-robot collaboration: from psychology to social robotics. *arXiv preprint arXiv:1705.10146*

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 2280–2292

Nelles, J., Kwee-Meier, S. T., and Mertens, A. (2018). Evaluation metrics regarding human well-being and system performance in human-robot interaction–a literature review. In *Congress of the International Ergonomics Association* (Florence, Italy, August 26-30: Springer), 124–135

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., et al. (2006). Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* (Salt Lake City, Utah, USA, March 2-3), 33–40

Task	Performance metrics	Definition or usability
Navigation	Failure rate	Percentage of navigation tasks completion failure
	Accuracy	The accuracy of the navigation
	Ergonomy or posture	Human ergonomy or posture
	Time to completion	The time needed to complete the task
	Rapidity	The time needed by the robot to adapt itself to the human or vice-versa
Perception	Velocity	The speed of the perception of the robot
	Accuracy	The accuracy of the navigation
	Time to completion	The time needed to complete the task
	Fluency	The fluency of the perception
	Effectiveness	Percentage of the success of the robot's perception
	Number of errors	The number of failures in the robot's perception
Management	Time delivery	The time needed to deliver the request from the robot to the human
	Time request	The time needed by the human (operator) to notice the request
	Number of human errors	The number of times the human cannot identify the situation with awareness
	Number of robot errors	The number of times the robot is misinterpreting human desires
	Trust	Trust of the human in the robot
	Number of actions	The number of actions needed to accomplish the task from the human and the robot
	Cognitive load	The workload required for the human to adapt to the robot
Manipulation	Positional accuracy	The accuracy of the position reached by the robot
	Positional repeatability	The repeatability of the robot to reach the same position
	Velocity	The speed of the robot to do the manipulation
	Time to completion	The time needed to complete the task
	Rapidity	The time needed by the robot to adapt itself to the human or vice-versa
	Cognitive load	The workload required for the human to adapt to the robot
	Ergonomy or posture	Human ergonomy or posture
	Dexterity	The robot's dexterity in doing the manipulation
	Effort or force	The physical effort (or force) that the human must provide to perform the manipulation
	Number of human errors	The number of times the human cannot identify the situation with awareness
	Number of robot errors	The number of times the robot is misinterpreting human desires
	Number of actions	The number of actions needed to accomplish the task from the human and the robot
Social	Persuasiveness	The ability of the robot to persuade the human about something
	Trust	Trust of the human in the robot
	Engagement in social characteristics	Engagement in social characteristics such as emotion, dialogue, personality. The engagement can be measured through the robot's acquisition time for capturing human attention and the duration of holding human interest
	Compliance	The compliance of the robot in appearance, adherence to norms, etc.
Common metrics	Effectiveness	The percentage of the mission that was accomplished with the designed autonomy
	Time to completion	The time needed to complete the task
	Number of human errors	The number of times the human cannot identify the situation with awareness
	Number of robot errors	The number of times the robot is misinterpreting human desires
	Number of actions	The number of actions needed to accomplish the task from the human and the robot
	Cognitive load	The workload required for the human to adapt to the robot
	Self-awareness	The robot knows its accuracy
	Autonomy	The robot autonomy

Table S1. Some metrics considered for the evaluation of HRC classified based on the task types (Steinfeld et al., 2006; Bütepage and Kragic, 2017; Nelles et al., 2018)

	Percentage of time improvement		Percentage of human errors reduction	
I_1, I_2, and I_3 values	Figure 6 (main paper)	Figure S2	Figure 7 (main paper)	Figure S3
$I_1 = 0$, $I_2 = 0.1$, and $I_3 = 0.9$	0.0	0.0	7.09412331431	4.37067607062
$I_1 = 0$, $I_2 = 0.2$, and $I_3 = 0.8$	0.0	0.0	12.6336858252	7.8020796966
$I_1 = 0$, $I_2 = 0.3$, and $I_3 = 0.7$	0.0	0.0	16.556882222	10.4704561576
$I_1 = 0$, $I_2 = 0.4$, and $I_3 = 0.6$	0.0	0.0	19.2303631915	13.3638679771
$I_1 = 0$, $I_2 = 0.5$, and $I_3 = 0.5$	0.0	0.0	22.3643444888	15.4993421616
$I_1 = 0$, $I_2 = 0.6$, and $I_3 = 0.4$	0.0	0.0	24.8542177267	17.9408050283
$I_1 = 0$, $I_2 = 0.7$, and $I_3 = 0.3$	0.0	0.0	28.4374851075	19.6044166944
$I_1 = 0$, $I_2 = 0.8$, and $I_3 = 0.2$	0.0	0.0	31.2453571429	22.1412554113
$I_1 = 0$, $I_2 = 0.9$, and $I_3 = 0.1$	0.0	0.0	29.2866666667	22.7585714286
$I_1 = 0$, $I_2 = 1$, and $I_3 = 0$	0.0	0.0	0.0	0.0
$I_1 = 0.1$, $I_2 = 0$, and $I_3 = 0.9$	0.0	0.0	17.5302972429	7.33482163197
$I_1 = 0.1$, $I_2 = 0.1$, and $I_3 = 0.8$	0.0	0.0	19.2143318311	10.5479301203
$I_1 = 0.1$, $I_2 = 0.2$, and $I_3 = 0.7$	0.0	0.0	21.3374721068	14.0052043345
$I_1 = 0.1$, $I_2 = 0.3$, and $I_3 = 0.6$	0.0	0.0	23.5786203979	16.5280674286
$I_1 = 0.1$, $I_2 = 0.4$, and $I_3 = 0.5$	0.0	0.0	26.7249188197	18.5858323843
$I_1 = 0.1$, $I_2 = 0.5$, and $I_3 = 0.4$	0.0	0.0	27.5937682456	20.9918095099
$I_1 = 0.1$, $I_2 = 0.6$, and $I_3 = 0.3$	0.0	0.0	28.9937657713	22.1913226588
$I_1 = 0.1$, $I_2 = 0.7$, and $I_3 = 0.2$	0.0	0.0	32.0019444444	22.9278030303
$I_1 = 0.1$, $I_2 = 0.8$, and $I_3 = 0.1$	0.0	0.0	34.205	23.5061904762
$I_1 = 0.1$, $I_2 = 0.9$, and $I_3 = 0$	0.646808142428	0.441773745339	0.0	0.0
$I_1 = 0.2$, $I_2 = 0$, and $I_3 = 0.8$	0.0	0.0	25.4460206798	12.376411145
$I_1 = 0.2$, $I_2 = 0.1$, and $I_3 = 0.7$	0.0	0.0	25.6904326925	15.6962303954
$I_1 = 0.2$, $I_2 = 0.2$, and $I_3 = 0.6$	0.0	0.0	27.5106709889	18.315465336
$I_1 = 0.2$, $I_2 = 0.3$, and $I_3 = 0.5$	0.0	0.0	30.0794412809	20.4125719276
$I_1 = 0.2$, $I_2 = 0.4$, and $I_3 = 0.4$	0.0	0.0	30.8272285354	22.0221180209
$I_1 = 0.2$, $I_2 = 0.5$, and $I_3 = 0.3$	0.0	0.0	31.9211207311	24.9664681615
$I_1 = 0.2$, $I_2 = 0.6$, and $I_3 = 0.2$	0.0	0.0	34.119047619	24.4444642857
$I_1 = 0.2$, $I_2 = 0.7$, and $I_3 = 0.1$	0.765528401311	0.0	0.0	27.6111904762
$I_1 = 0.2$, $I_2 = 0.8$, and $I_3 = 0$	2.4788012545	1.80659141302		0.0
$I_1 = 0.3$, $I_2 = 0$, and $I_3 = 0.7$	0.0	1.91789786313	30.2540826341	16.6733556418
$I_1 = 0.3$, $I_2 = 0.1$, and $I_3 = 0.6$	0.0	0.0	30.7411696561	19.7313028361
$I_1 = 0.3$, $I_2 = 0.2$, and $I_3 = 0.5$	0.0	0.0	31.3673661689	21.5925174216
$I_1 = 0.3$, $I_2 = 0.3$, and $I_3 = 0.4$	0.0	0.0	33.7247655123	23.7318043068
$I_1 = 0.3$, $I_2 = 0.4$, and $I_3 = 0.3$	0.0	0.0	34.3449001924	23.9794936545
$I_1 = 0.3$, $I_2 = 0.5$, and $I_3 = 0.2$	0.0	0.0	36.4016269841	26.2835119048
$I_1 = 0.3$, $I_2 = 0.6$, and $I_3 = 0.1$	2.77628815301	1.46088929863	36.7869047619	24.7346428571
$I_1 = 0.3$, $I_2 = 0.7$, and $I_3 = 0$	5.6292792232	4.03759880367	0.0	0.0
$I_1 = 0.4$, $I_2 = 0$, and $I_3 = 0.6$	0.0	0.0	36.3891268668	20.3970746112
$I_1 = 0.4$, $I_2 = 0.1$, and $I_3 = 0.5$	0.0	0.0	35.2710778111	22.4513601676
$I_1 = 0.4$, $I_2 = 0.2$, and $I_3 = 0.4$	0.0	0.0	35.8305131674	25.1917275086
$I_1 = 0.4$, $I_2 = 0.3$, and $I_3 = 0.3$	0.0	0.0	36.4988095238	26.7846236171
$I_1 = 0.4$, $I_2 = 0.4$, and $I_3 = 0.2$	2.97937356761	0.0	37.8678571429	26.1883928571
$I_1 = 0.4$, $I_2 = 0.5$, and $I_3 = 0.1$	6.8298290148	3.67128494973	35.0916666667	21.646547619
$I_1 = 0.4$, $I_2 = 0.6$, and $I_3 = 0$	10.0581040567	6.91767350379	0.0	0.0
$I_1 = 0.5$, $I_2 = 0$, and $I_3 = 0.5$	0.0	0.0	38.5427888223	22.5377313961
$I_1 = 0.5$, $I_2 = 0.1$, and $I_3 = 0.4$	0.0	0.0	39.8677200577	25.6673357198
$I_1 = 0.5$, $I_2 = 0.2$, and $I_3 = 0.3$	2.5889362939	0.0	40.2805687831	28.5753607504
$I_1 = 0.5$, $I_2 = 0.3$, and $I_3 = 0.2$	7.56507185318	2.4071413430	38.0336309524	21.5879166667
$I_1 = 0.5$, $I_2 = 0.4$, and $I_3 = 0.1$	12.187798206	7.11099379702	41.565	22.1403571429
$I_1 = 0.5$, $I_2 = 0.5$, and $I_3 = 0$	15.7060720797	11.0874137267	0.0	0.0
$I_1 = 0.6$, $I_2 = 0$, and $I_3 = 0.4$	0.0	0.0	41.0730555556	25.1399181374
$I_1 = 0.6$, $I_2 = 0.1$, and $I_3 = 0.3$	8.17205250781	0.0	42.012965368	28.6511640212
$I_1 = 0.6$, $I_2 = 0.2$, and $I_3 = 0.2$	14.1780122742	5.64235894103	43.109047619	21.2425054113
$I_1 = 0.6$, $I_2 = 0.3$, and $I_3 = 0.1$	18.8995795602	11.1593671564	45.3266666667	24.2171428571
$I_1 = 0.6$, $I_2 = 0.4$, and $I_3 = 0$	23.2260934025	15.568227852	0.0	0.0
$I_1 = 0.7$, $I_2 = 0$, and $I_3 = 0.3$	17.2439908187	3.33567251462	44.296547619	18.6905624931
$I_1 = 0.7$, $I_2 = 0.1$, and $I_3 = 0.2$	22.4988118634	10.0605063426	45.0121428571	20.8104816017
$I_1 = 0.7$, $I_2 = 0.2$, and $I_3 = 0.1$	27.4951742932	15.4641925539	43.6866666667	21.2117857143
$I_1 = 0.7$, $I_2 = 0.3$, and $I_3 = 0$	31.8854548846	21.023549533	0.0	0.0
$I_1 = 0.8$, $I_2 = 0$, and $I_3 = 0.2$	33.0590564877	14.5036859249	44.6876190476	22.0960714286
$I_1 = 0.8$, $I_2 = 0.1$, and $I_3 = 0.1$	37.9880475163	20.7778656126	45.2483333333	22.3210714286
$I_1 = 0.8$, $I_2 = 0.2$, and $I_3 = 0$	41.9390153589	26.6215414675	0.0	0.0
$I_1 = 0.9$, $I_2 = 0$, and $I_3 = 0.1$	49.8353835563	26.317921026	50.5583333333	21.8939285714
$I_1 = 0.9$, $I_2 = 0.1$, and $I_3 = 0$	53.3069306931	33.3469782673	0.0	0.0
$I_1 = 1$, $I_2 = 0$, and $I_3 = 0$	66.6666666667	40.0	0.0	0.0

Table S2. Time improvement percentage and human errors reduction percentage obtained for all the figures of the main paper and the supplementary material.

	Step	Time in seconds
Real tests	Computation time of the decision-making (applying our formalization)	The robot takes an average of 0.5s to choose the action to perform after knowing the state of the puzzle through the perception part.
	Time taken by the robot for the perception of the puzzle state	The robot takes between 20s and 30s depending on how well the cubes are placed and how many cubes are left to assemble.
	Time taken by the robot for doing a physical movement	The robot takes on average 15s for doing a physical movement.
	Waiting time for the robot when it gives an indication for the human	The robot waits between 5s and 15s each time it gives an indication to the human, depending on its complexity (for example, to ask the human to remove a cube, the robot waits 5s, and to ask the human to take a certain cube and place it in a certain position, the robot waits 15s).
	Global time taken by the robot to perform an action	It is between 20s and 60s, depending on the complexity of the movement (the number of cubes left to assemble at this iteration) and if the robot gives indications to the human. We considered that it was 60s.
	Global time taken by the human to perform an action	The human takes between 1s and 30s, depending on the complexity of the movement (if they know what to do or not). We considered that it was 20s.
Tests in simulation	Time required for all probability distributions of possible human actions without printing the figures (such as Figures S2 and S3)	The Python code takes between 80s and 100s on a Dell laptop with an Intel Core i7 CPU and 32GB RAM.

Table S3. Computation and execution times of the experiments in real and in simulation