Trust as a Metric for Auction-based Task Assignment in a Cooperative Team of Robots with Heterogeneous Capabilities

Alberto Grillo[a], Stefano Carpin[b], Carmine Tommaso Recchiuto[a], Antonio Sgorbissa[a,*]

[a]*DIBRIS, University of Genoa, Via all'Opera Pia 13, 16145 Genoa, Italy*
[b]*School of Engineering, University of California Merced, 5200 North Lake Rd, CA 95343, Merced, USA*

Abstract

As robots become part of our everyday lives, they may be required to cooperate without being aware of each other's capabilities (e.g., because different teams have developed them), and therefore will have to trust each other to work together safely and efficiently. Starting from this premise, this work identifies trust as an essential metric to assign tasks to robots using auction-based mechanisms. We model trust by taking inspiration from popular models in the literature and adapting them to an open environment in which heterogeneous robots may dynamically enter or exit, execute assigned tasks, or verify the correct execution of tasks by other robots. Robots are considered to be heterogenous in the sense that they may have different capabilities in executing and verifying the execution of actions. In the proposed model, "doing an action" and "verifying the execution of an action" are distinct, not necessarily overlapping, capabilities. Some robots may be able to do an action, whereas some robots may not be able to do it but only to observe and judge the ability of other robots to do it. After introducing the relevant formalism, the article describes the system's architecture implemented in ROS and multiple experiments performed in simulation and with real robots (one NAO and two Pepper robots by SoftBank Robotics), providing a proof-of-concept for broader utilization of the system in cooperative robotic scenarios.

Keywords: Trust, Robot-Robot Interaction, Task Assignment, Social Robotics

1. Introduction

The number of robots used in everyday activities is steadily increasing and expected to keep growing. This will undoubtedly occur in industrial settings where the next generation of robots will be crucial in meeting the dynamic needs of collaborative and intelligent manufacturing that characterize the so-called *Industry 4.0 and Industrial Internet of Things* [1, 2]. However, international market analyses anticipate widespread use of robots also by the general public [3]. In both industrial and domestic scenarios, different types of robots with different capabilities marketed by different companies will need to work together, and possibly with humans, to achieve shared goals. Given these premises, "teammates" will likely require to "trust" each other for efficient teamwork. In multi-agent systems, it is possible to define trust as "the subjective probability by which an agent (the trustor) expects that another agent (the trustee) succeeds in performing a given action on which its welfare depends" [4].

In the last decade, the concept of trust and the associated trust dynamics have received considerable attention from the Human-Robot Interaction (HRI) community. Previous work has shown that trust does not only depend on the capabilities of the robotic agent but also on other factors ranging from user's expectations [5, 6] to the physical appearance of the robot [7, 8, 9]. Other works have shown that an expression of the robot's vulnerability can positively impact trust and facilitate social engagement [10, 11]. It has also been shown that trust is negatively affected by robots not being transparent in explaining the motivations for their decisions and actions [12, 13, 14]. Unsurprisingly, mistrust in the robot can negatively affect human-robot cooperation [15, 16, 17, 18].

*Corresponding author
Email address: antonio.sgorbissa@unige.it (Antonio Sgorbissa)

The robotic literature has primarily investigated trust in the HRI context. However, not all situations involving trust require the trustee or the trustor to be human. Both in industrial and service scenarios, there are situations where autonomous robots must cooperate without having perfect knowledge about each other's capabilities. For example, consider a robot and a smart pill dispenser that need to decide "who" will remind the user to take a medicine. Reminding medications is a critical task that may raise ethical concerns because of its potential negative impact in case of failure. In general, not all robots are equally suited to perform such task, or they may be less or more reliable in performing it. In this context, the trust that robots have in their own abilities versus other robots' capabilities may affect the selection of the most suitable candidate to perform the task. It may be argued that, for robots to be aware of each other capabilities, explicit communication might be sufficient. However, a wrong perception of others' capabilities may emerge not because robots "lie bragging about abilities they don't have" but because developers do not actually "know" how robots will behave in any given situation. Compare a "very quiet, older people's house with a clean and tidy floor" versus a "messy, young family's house where parents and children always shout" or a "large warehouse hosting a crowded fair". To estimate the robot's success rate in performing actions in a given scenario, one should ideally make experiments in that scenario and statistically analyze the results. However, experiments may not be feasible after a customer buys the product. So, for instance, social robot producers may declare on the user manual that their robot can understand what people say: but how reliable is speech-to-text conversion in different environmental conditions? How much does it depend on the speaker's volume or environmental noise? What about a cleaning robot that declares to "reliably" clean the house? How much does floor coverage depend on the presence of furniture, carpets, and children's toys? Since doing experiments and analyzing results in any given scenario is impossible, robots should at least continuously assess their own and other agents' capabilities during operations.

While human-robot trust has been studied extensively in both directions, the study of trust dynamics between cooperating robots has received less attention. Accordingly, the main contribution of this work is to explore and test models of trusts when both the trustors and the trustees are robotic agents with different perceptual, reasoning, and actuation capabilities, that are periodically assigned tasks to be performed to achieve a shared objective. The goal of this work is not only theoretical, but it also includes the implementation of a framework for trust-based task assignment that can operate in a distributed open environment. The Trust Framework is implemented in ROS [19] and provides heterogeneous agents with a portfolio of complementary services paralleling their onboard capabilities. We emphasize that robots are heterogeneous in the sense that they may have different capabilities both in executing and verifying the execution of actions. Some robots may be able to do an action, whereas some robots may not be able to do it but only to observe and judge the ability of other robots to do it. The purpose of such services is to allow agents to model trust in each other regarding the capability to accomplish an assigned task and use this model to evaluate possible candidates for task assignments whenever needed. In other words, agents may use the Trust Framework to outsource tasks they need to complete to achieve a particular goal.

The article is structured as follows. Section 2 reviews relevant literature, by describing how the concept of trust has been studied in multi-agent systems and what are the main theoretical and practical findings, both in the human-robot and the robot-robot case. Section 3 presents the methodology adopted to estimate trust within a community of cooperating agents, including different trust metrics taking inspiration from the recent literature. Section 4 describes the framework developed, including procedures to find a consensus for task assignment using a trust-based auction-like mechanism and its ROS implementation. Sections 5 and 6 describe simulated results and real-world interactions between different humanoid robots (one NAO and two Pepper robots) to assess the properties of the framework. Conclusions follow in Section 7.

2. State of the Art

With robots gradually moving from laboratories to complex human-populated environments (such as factories, hospitals, offices, or homes), it is necessary for them to exhibit more complex cognitive abilities to cooperate with humans or other robotic agents. Recent research in Robotics and Artificial Intelligence (AI) is paving the way to new forms of robot interaction, either with humans or

other agents, characterized by greater adaptability, shared decision-making, and mixed-initiative. However, researchers have argued that even though robotic agents have the potential for being valid teammates in a variety of tasks, they may be underutilized due to a lack of trust from their partners [20]. Therefore, to enable fruitful cooperations between a robot and another agent (human or not), the first step is to clarify the relations between the concept of "autonomous agent" and "trust."

According to [21], autonomous agents are systems that can change their behaviour in response to unexpected conditions and events. When focussing on robots, such notion of autonomy refers to the ability of a machine to perform a task, or part of it, with no (or substantially reduced) human intervention. The degree of independence from humans defines two main classes of autonomous agents, i.e., Human-In-The-Loop (HITL) and Human-On-The-Loop (HOTL) systems. HITL machines can autonomously carry out a task for a limited time interval, but periodically require human input to move forward. HOTL systems are machines that can execute a task entirely without external aid but may require a human supervisor to intervene in case of failure. According to the above definition of autonomy, both humans and robots can be classified as autonomous agents, and the term Multi-Agent Systems (MAS) [22] can be accordingly extended to include heterogenous teams composed of humans and robots. With both humans and robots belonging to the class of autonomous agents, we expect that several constructs related to the social relations between humans can be naturally mapped to the robotic domain - including trust. In particular, it is argued in [21] that HOTL systems require a high level of trust to be accepted in our daily lives.

Although researchers extensively investigated trust, an universally agreed definition has not been provided yet. As mentioned before, the prevalent definition of trust in the MAS research field is the subjective probability by which an agent A expects that another agent B performs a given action on which its welfare depends [23, 4]. Defining trust is not sufficient: it is also essential to design tools to measure it, an objective still far from being achieved. According to [4], trust can be modelled as an expectation/prediction related to an uncertain behaviour primarily based on the outcomes of previous interactions. The authors suggest that an agent may evaluate its trust in other agents concerning a specific behaviour given its "mental image" of other agents. Through its reasoning processes combining different sources of information, an agent may derive beliefs and expectations about "good" or "bad" behaviours of other agents, which may lead to a decision or an intention. In this sense, the concept of trust relates to Theory of Mind (ToM) [24, 25, 26].

In [27], a comprehensive survey on trust models is presented. The authors analyze the differences between multiple definitions of trust adopted in different research contexts, the dynamics of trust evaluation, and the factors influencing this process. They observe that binary models have a lower resolution but are more straightforward and efficient in estimating whether or not two agents trust each other [28]. On the other hand, modeling trust as a scalar number, as in continuous/discrete models [29], offers more flexibility. Observing that trust is subjective in nature and context-dependent, the authors then present an exhaustive survey on various attributes of trust proposed in different contexts with different purposes. In doing this, they distinguish between *Individual Trust* and *Relational Trust*. Individual trust includes attributes describing how the evaluation of trust depends on personal characteristics and is further classified into *Logical Trust* and *Emotional Trust*. Logical trust describes reasoning on trust based on one's logical processes and objective observations. Emotional trust describes reasoning on trust based on one's emotional state. Relational trust includes attributes describing how trust can emerge from an individual's relationships with other individuals. Each attribute is also analyzed extensively by reviewing the several possible formulations and metrics proposed in the literature.

In this general scenario, it is convenient to focus on two different sub-cases in which trust may play a vital role in the context of autonomous multi-agent systems: human-robot interaction (HRI) and robot-robot interaction (RRI).

2.1. Human-Robot Interaction

In [30], benchmarks for evaluating human-robot cooperation are proposed, and the authors posit that mutual trust is crucial for proper cooperation. Along the same lines, the work described in [31] hypothesizes that many factors play a crucial role in evaluating and "controlling" human-robot trust. These include robot-related factors such as robot performance and physical attributes, human-related factors such as personal skills and personality traits, and environmental factors as the required

tasks. To support their hypothesis, the authors performed a meta-analysis on ten scientific articles focusing on HRI. The analysis confirmed that trust is essential to human-robot teams and that different factors have distinct weights. Furthermore, according to the authors, robot traits are the most critical factors in trust development. In contrast, environmental features provide a moderate effect, and there is little evidence that human elements influenced trust in HRI. Finally, results pointed out that developing a trust relationship between humans and robots is strongly affected by several other issues such as trust calibration and opaqueness.

As discussed in [32], trust in HRI often fails as humans are not able to properly rely on robotic agents. Then, *trust calibration* may play a key role when humans over-trust or under-trust the robotic agent. Extensive research in human-machine interaction [15, 16, 17, 18] has shown that the more operators trust automated systems, the more they tend to rely on them to accomplish tasks. When operators trust their abilities more than those of the system, they tend to instead choose "manual control" modes. Failure to meet user expectations can result in system misuse or disuse, respectively, if the expectations are too high or too low [33]. Other works [34] have confirmed that distrust can reduce people's willingness to accept the information provided by a robot or follow a robot's advice, thus limiting the potential benefit of robotic systems [35].

Another key issue to consider in developing trust relationships in HRI is *opaqueness*. State-of-the-art autonomous robots extensively use AI processes that may take complex and adaptive decisions. Still, the motivations behind such decisions are usually complicated to understand. Thus, the opaqueness of AI systems may lead humans to have concerns and ultimately mistrust due to the robot's inability to explain the motivation for its actions. This effect is more dramatic for inexperienced users who may find the behaviour of a robot hard to predict, generating a sense of distrust and impeding efficient teamwork. To increase the user's trust in the system, it is essential to reduce the system opaqueness, making the robot more predictable and understandable. Indeed, one of the most active research topics in HRI is informing the users about the agent's intentions, the so-called explainable AI (XAI). In [14], a systematic literature review of explainable agents and robots is presented. The survey reveals that trust and transparency are the most prominent drivers of XAI research since they are essential to increasing the user's confidence in the system by providing clear insight into how its reasoning mechanisms work.

Reinforcement learning (RL) [36] has been proposed to implement the concept of trustable and explainable robots. In particular, [37] proposes Inverse Reinforcement Learning (IRL) as a way to formalize our ability to reason about other people's mental states. Dynamic Bayesian networks for trust estimation have been proposed in [38], where the authors present a model that enables a robotic agent to quantify the degree of trust that a human supervisor has in the agent itself, thus making it possible for the robot to dynamically adapt its behaviours to improve its trustworthiness. The work proposed by [39] embraces the reverse perspective and explores ways in which the agent can measure the trustworthiness of the human operator. In the spirit of developmental robotics, an approach inspired by mechanisms observed in children's cognitive development [40, 41], this work presents a robotic agent that learns to evaluate the degree of trustworthiness of its information sources to make autonomous decisions. Finally, trust and its dynamics in the interaction between humans and artificial agents have been investigated in the framework of game theory [42, 43, 44]. A recent survey about trust models and metrics in HRI has been proposed in [45].

2.2. Robot-Robot Trust

Trust in the field of Multi-Agent Systems has also been extensively studied. In [46], the most relevant trust and reputation models published in the last two decades are surveyed, whereas [47] analyzes existing trust models from a game theoretic perspective to highlight the special implications of including human beings in a MAS.

However, when explicitly focussing on robots and other embodied agents interacting with each other (i.e., not necessarily including humans in the loop), the development of trust relationships as a basis for cooperation has not been explored in depth.

Investigating trust between robots might at first not appear relevant because robotic agents work according to their specifications, do not lie or execute malevolent actions on purpose (unless hacked – something we do not consider here), and it is hard to imagine robotic agents whose judgment about others is affected by their own personality or emotions. Therefore, most researchers address the collaboration between robots by focussing on task as-

signment, cooperative control, and similar problems requiring finding optimal strategies, with no space for subjective elements like trust. However, reasoning about trustworthiness does not require the possibility for agents to be malevolent. Indeed, trust also plays a key role when an agent has to evaluate its own and other agents' capabilities by comparing a priori beliefs and claims with observed results: that is, the assessment that an agent makes about its own and other agents' capabilities *before* and *after* performing a task. As discussed in the Introduction, a misalignment between a robot's expected and actual capabilities can occur due to the impossibility of predicting outcomes in any environmental conditions, limitations in perception and action, and other factors, including insufficient, wrong, or outdated documentation, limited testing, degradation of performance due to wear, and more.

Consider, for instance, the problem of assigning tasks to multiple agents. In [48], the multi-robot task assignment (MRTA) problem was reviewed by illustrating several criteria and algorithms to optimize results, each with different characteristics to adequately address a range of problems associated with a given context. When addressing the problem of distributed multi-robot task allocation, one of the most popular approaches is the use of auction-based methods [49]. The idea is straightforward, and numerous variants have been proposed [50, 51, 52]. Agents, possibly differing in their sensory or actuating hardware or functionalities, are allowed to make a bid for a task. The auctioneer will assign the task to the most fitting agent to perform it according to some metric.

Irrespective of the specific algorithm adopted for task assignment, MRTA typically relies on a shared communication protocol and the evaluation of one or more metrics that measure the fitness of each candidate robot to contribute to the needed task at a given time. However, appropriate evaluation metrics are critical because of the possible discrepancy between agents' beliefs and claims and their actual capabilities. The problem becomes even more important when considering "open environments", i.e., environments in which heterogeneous robots with different capabilities (but also personal assistants, distributed sensors, and other devices) may enter and leave asynchronously and without notice. This makes metrics evaluation challenging since it requires evaluating the capability of agents that just joined the team and that other agents may have never seen before or agents commercialized in different countries by different producers with different quality standards and specifications.

In this scenario, and by mimicking the approach that a team composed of humans might pursue to address similar problems, we conjecture that trust, as formalized in the next section, is a promising metric to coordinate heterogeneous agents.

3. Methodology

3.1. Trust-based task allocation

We start defining the elements to model the context in which agents operate. We assume the availability of the following finite sets:

- a set of N_e events $\mathcal{E} = \{E_i\}$. E_i may be a combination of perceptual inputs, an explicit command from a user, an alarm, or any other external stimulus that is processed by a robot and whose effect is triggering a specific plan (i.e., one or more actions to be executed by the robot);
- a set of N_a actions $\mathcal{A} = \{A_i\}$. Each action A_i is identified by a shared, unique identifier;
- a set of N_g agents $\mathcal{G} = \{G_i\}$. G_i is any hardware device or software module that can communicate with other agents and can auction or bid for actions. An agent G_i knows:
  - a set of N_i^e actions $\mathcal{A}_i^e = \{A_j\} \subseteq \mathcal{A}$ that it can execute, where each action $\{A_j\}$ is associated, for each agent, to a portion of code. Under the assumption that the outcome of each action can be either labelled as *success* or *failure*, for each G_i and $A_j \in \mathcal{A}_i^e$ it is possible to formalize the sequence of A_j outcomes as a Bernoulli process in which 1 means *success* in performing the action while 0 means *failure*, with a given success rate.
  - a set of N_i^v actions $\mathcal{A}_i^v = \{A_j\} \subseteq \mathcal{A}$ that it can verify by executing a different portion of code. For each G_i and $A_j \in \mathcal{A}_i^v$ it is possible to model G_i's capability of verifying A_j with a given True Positive (TP) rate in identifying successes and a True Negative (TN) rate in identifying failures.
  - A set of N_i^p plans $\mathcal{P}_i = \{P_j\}$, each defined as a sequence of known actions belonging to $\mathcal{A}_i^e$ that must be completed, serially or

at the same time, to successfully respond to a corresponding event E_j.

Each agent in the Trust Framework can:

- start a plan to manage one or more events;
- auction an action in its plan or bid on an action auctioned by another agent;
- execute or verify the execution of one or more actions;
- gather data from other agents about the success or failure of a given action;
- update its trust in other agents.

The overarching idea may be summarized as follows. First, using a portfolio of trust-related metrics, agents dynamically gather data about the other agents' capability to (i) perform actions; (ii) verify the outcomes of actions performed by other agents. Then, they will iteratively use and update these metrics to evaluate the trustworthiness of other agents, including themselves, during auctions, thus ultimately taking trustworthiness into account when taking a new decision for task assignment.

A simple example may help to clarify these concepts. Suppose that a humanoid robot G_1 needs to handle an event E_1 =*PrepareLunchIngredients*, which may follow an explicit request by a user, be generated by a clock, or even issued by another agent. G_1 knows how to handle the event and has a plan $\{A_1, A_2\}$ to respond to it. Since it participates in the framework, it maps its low-level sensorimotor routines to open the fridge and take ingredients to the two actions A_1 =*OpenFridgeDoor* and A_2 =*TakeIngredients*. However, G_1 can also consider assigning the actions to other agents in the framework that may be more trustworthy in performing them – G_1 executed both actions several times, but it often fails.

One day, when G_1 auctions A_1, it turns out that there is a new agent G_2 (e.g., a smart, motorized fridge door connected to the local network) that bids for A_1 =*OpenFridgeDoor*. G_2 considers itself to be very trustworthy in performing that specific action. After gathering all the available information, including data sent by G_2 and other participants, the auctioneer G_1 takes the final decision to assign A_1 to G_2 because G_2 claims to be very reliable (and no agent can prove the opposite). In contrast, every agent (including G_1) is aware that G_1 has been very unreliable in performing that task.

Next, G_1 gets ready to observe and evaluate G_2's outcomes along with other agents that volunteered to do so, among which a smart RGB-D camera G_3 on the ceiling. G_2 correctly opens the door. However, since G_1's vision algorithms have limitations, G_1 mistakenly recognizes the action as a failure. An exchange of data among agents follows. G_2 communicates that it successfully opened the door, and other devices in the room agree with what G_2 says (among which the RGB-D camera G_3, which was very trustworthy in observing this action in the past). G_1, being the only agent who detected a failure, understands that not only its capability of performing A_1 is not very reliable, but also its capability of verifying the execution of this action when done by other agents. Accordingly, it will update its beliefs for future occurrences of event E_1.

After the fridge door is opened, the auction for the second action A_2=*TakeIngredients* starts. Being a humanoid robot and the only agent implementing this action, G_1 wins its own auction and starts taking the ingredients. This time, G_1 correctly judges the outcomes of the action. Its judgment is confirmed by most of the verifying agents, among which is G_4, an RFID antenna located in the fridge capable of detecting the presence of RFID-tagged ingredients. G_4 did not participate in the auction to perform A_2 because it does not implement it but volunteered as a verifying agent to evaluate G_1's capabilities for future interactions.

The procedure used for auctions is described in Algorithm 1 without referring to formal metrics, which are introduced in the next Section.

Notice that if all agents share their opinions about other agents' success rates in performing and observing actions, decision-making can work in a distributed way. Even if the auction is managed by an auctioneer that assigns the action to the winner, any other agent in the framework has all the information to predict the auctioneer's choice, which may still be required but only for synchronizing execution as clarified in Section 4. Moreover, in the case of *collectivistic* decision-making (introduced later), all agents will unavoidably come to the same conclusions about the most suitable agent to perform the action.

3.2. Modeling Trust

The assignment of actions to execute plans relies on metrics to model the trustworthiness of

Algorithm 1 Trust-based task assignment

Require: A set of agents $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$; an Event E_l detected by agent G_i; a Plan $P_l = \{A_k\}$ known to G_i and suitable to manage E_l.

1: Agent G_i selects plan P_l
2: **for** each action A_k in P_l **do**
3: G_i starts an auction for A_k
4: All agents able to perform A_k bid on A_k by sending data to G_i about (i) their trustworthiness in executing A_k; (ii) other agents' trustworthiness in executing it; and (iii) their trustworthiness in verifying other agents' success or failure.
5: G_i evaluates all bidders.
6: **if** the perceived competence is above a threshold for at least a bidder **then**
7: G_i assigns A_k to the bidder G_j with the highest perceived competence.
8: **else**
9: G_i asks for the intervention of a human supervisor.
10: **end if**
11: G_j executes action A_k
12: All agents able to verify the proper execution of A_k (including G_j itself) vote for either the *success* or *failure* of the action, and send outcomes to each other.
13: All agents update their own and other agents' trustworthiness according to direct observations and data received from other agents.
14: **end for**

agents. Trustworthiness plays a key role in (i) scoring bidding agents and taking consequent decisions to assign actions and (ii) weighing agents' opinions about the outcomes of such actions. Since we aim to build a general model that might be applied to different domains, we surveyed several metrics proposed in the literature to model such concepts. In the following, we propose the role of two Context-independent metrics [45]: *Reliability* and *Verification Trustworthiness*, which are composed to compute the agent's *Perceived Competence*.

3.2.1. Reliability

Reliability, one of the attributes of Logical Trust [27], is the estimate, made by agent G_i, of the success rate of agent G_j in executing action A_k, i.e.,

$$Rel(k,i,j) = \frac{\sum_{n=1}^{N_{exe}(k,i,j)} Res_n(k,i,j)}{N_{exe}(k,i,j)} \quad (1)$$

where:

- $N_{exe}(k,i,j)$ is the number of times that G_i has observed G_j performing A_k;
- $Res_n(k,i,j)$ is a binary value encoding the observed result of the n^{th} execution of the action: 1 stands for *success* while 0 stands *failure*.

Reliability has a value in the range $[0,1]$.

3.2.2. Verification Trustworthiness

When an agent executes an action, other agents may volunteer to verify the outcome. Therefore, it is necessary to evaluate not only the trustworthiness of an agent to execute a given action but also how trustable an agent is in verifying that a particular action has been correctly executed[1].

To this end, this article introduces a new metric referred to as *Verification Trustworthiness*, which measures the degree of the trustworthiness of a verifying agent depending on the consensus it has around its judgment skills. Formally, the *Verification Trustworthiness* measures how much the verification made by agent G_j about the success of action A_k is considered trustworthy according to an agent G_i and requires counting the number of opinions that are concordant or discordant with the judgment of G_j.

The first step is to assess the consensus around the judgment made by agent G_j.

$$VTW_n(k,i,j) = \frac{N_{con}(k,j) - N_{dis}(k,j)}{N_{con}(k,j) + N_{dis}(k,j)} \quad (2)$$

where:

[1] One could hypothesize the presence of "second-order verifiers" that observe "first-order verifiers" to judge how trustable they are in judging other agents. However, this looks unreasonable since it would ideally lead to a regression to infinite, as soon as we start thinking about recursively implementing "third-order verifiers" to check the reliability of "second-order verifiers," and so on. Quantifying the trustworthiness of verifiers is important, as verifiers themselves may be prone to errors due to their limited perceptual capabilities ("Quis custodiet ipsos custodes? Who watches the watchers?"), but the solution necessarily needs to be different, as shown in the following.

- $N_{con}(k,j)$ is the number of verifying agents that concur with the judgment made by G_j about the success of action A_k;

- $N_{dis}(k,j)$ is the number of verifying agents that disagree with the judgment made by G_j about the success of action A_k.

$VTW_n(k,i,j)$ refers to the n^{th} execution of action A_k, and has a value in the range $[-1,1]$. When all the agents agree, $VTW_n(k,i,j) = 1$; this value decreases as the number of discording agents increases and reaches the lower bound $VTW_n(k,i,j) = -1$ when all agents disagree with G_j. It shall be noted that the value of $VTW_n(k,i,j)$, depending only on the consensus about G_j's opinion on the execution of A_k, is computed in the same way by all agents G_i in the framework. This explains why the parameter i does not appear on the right side of Eq.(2). However, we kept the index i in the formulation of $VTW_n(k,i,j)$ to underline the fact that it can be computed in parallel and independently by all agents in the system. For simplicity, in the following, we will also use the notation $VTW_n(k,j)$.

The second step is to compute the average *Verification Trustworthiness* of an agent G_j in verifying A_k, which is:

$$VTW(k,j) = \frac{\sum_{n=1}^{N_{ver}(k,j)} VTW_n(k,j)}{N_{ver}(k,j)} \tag{3}$$

where:

- $N_{ver}(k,j)$ is the number of times that action A_k has been verified by G_j.

$VTW(k,j)$ has the same value when computed by any agent G_i.

3.2.3. Perceived Competence

Whenever an auctioneer G_i has to decide about assigning an action A_k to a bidding agent G_j, the former computes the *Perceived Competence* of the latter. The *Perceived Competence* is computed as a function of the *Reliability* of G_j estimated by all agents $\{G_1 \dots G_n\}$ participating to the auctions as well as their own *Verification Trustworthiness*. These values are stored in the vectors $[Rel(k,1,j) \dots Rel(k,n,j)]$ and $[VTW(k,1) \dots VTW(k,n)]$ and the perceived competence is then computed as:

$$\begin{aligned} Comp(k,i,j) = \quad & f([Rel(k,1,j) \dots Rel(k,n,j)], \\ & [VTW(k,1) \dots VTW(k,n)]). \end{aligned} \tag{4}$$

Eq.(4) shows the computation of the *Perceived Competence* in its more general formulation. As explained in the next Section, the way in which individual *Reliability* and *Verification Trustworthiness* contribute or not contribute to $f(\cdot)$ will depend on:

- how we compute metrics in the transitory, i.e. when the agent G_j has not yet executed the action A_k a sufficiently large number of times: *Boot mode*, *Window mode* or *BCI mode*;

- the behaviour of agents towards the community: *individualistic* or *collectivistic*;

- the disposition of agents towards other agents: *optimistic*, *pessimistic*, *realistic*.

Terms as *collectivistic* or *individualistic* are not intended to model an actual "attitude" of agents. They are just meant as labels to characterize different strategies. For instance, a *collectivistic-optimistic* robot will likely take in the highest account the opinions received by other agents (differently from what we expect from a *individualistic* robot), be favorable to give an additional opportunity to agents even if they fail once, and tend to evaluate agents based on their best performances instead of their failures (differently from what we expect from a *pessimistic* robot).

No matter how it is computed, the highest *Perceived Competence* among all bidders is finally compared to a "cooperation" threshold that is inspired by the one proposed by [28] and takes into account the *Importance* of the action A_k for the auctioneer G_i (in the range $[0,1]$). The auctioneer will trust the winning bidder to execute the action only when the *Perceived Competence* is higher than the threshold. Otherwise, the auctioneer will request the assistance of a human supervisor. If the action is very important, the auctioneer requires more guarantees about the trustworthiness of bidders, i.e., the action is assigned to the winning bidder G_j if and only if:

$$Comp(k,i,j) \geq K Importance(k,i) \tag{5}$$

where K is a gain based on the *optimistic/pessimistic/realistic disposition* of the trustor, as explained next.

As the reader may imagine, different or additional metrics may be considered [27]. Here, for brevity's sake, we have just described the ones implemented and tested in the experiments, but we refer the reader to section 7 for an additional discussion about this subject.

3.3. Dynamic behavior

Each agent in the framework stores data $Res_n(k, i, j)$ about the results it observed for each action and the values $VTW_n(k, j)$. When n is sufficiently high and the estimation of the *Reliability* and *Verification Trustworthiness* depends on a reasonably large sample of observations, data are handled as described in the previous section. However, it is necessary to decide how to process data in the initial phase when agents start to interact and may not have enough information to judge each other's trustworthiness.

A straightforward solution is that, when no data are available about action A_k, each auctioneer G_i shall trust G_j's declared *Reliability* $Rel(k, j, j)$ but, as soon as data are available auctioneers will rely on observations. This approach requires establishing the number of observations needed to consider them as relevant. For example, suppose an auctioneer G_i starts relying on its own estimate of G_j's *Reliability* after just one execution of A_k. If G_j is observed by G_i to fail the first assigned action, its *Reliability* will drop to zero. As a result, G_i will mark G_j as completely untrustable in executing that action and preventing it from being considered again in the future. To address this issue, we explored three approaches to process *Reliability* data when agents do not have enough data to infer each other's capabilities.

3.3.1. Boot Mode

The most straightforward approach is referred to *Boot mode*. This mode requires the definition of a "boot phase" length, expressed as the number of A_k's auctions in which an agent G_j needs to participate before an auctioneer G_i starts using $Rel(k, i, j)$ as a measure of G_j's trustworthiness. When there is not enough available data, auctioneers select winners accordingly to the highest declared *Reliability* $Rel(k, j, j)$ sent by each bidder G_j.

This approach has two drawbacks. First, defining a proper boot phase length may be challenging. Increasing its duration allows for gathering a broader and more statistically relevant data sample before using *Reliability*. Still, it may also result in lower performances in the boot phase since auctioneers will assign actions to bidders according to their declarations and not actual observations. During the boot phase, an overly confident bidder that declares itself to be the best candidate for everything could win all auctions, even if it is the most incompetent. On the other hand, decreasing the length of the boot phase may produce an "unforgiving behavior" whereby auctioneers will hardly forget mistakes made in initial auctions. Since action assignment is based on *Reliability*, auctioneers may never give an unreliable agent a second chance, thus making it very hard for it to recover from having a bad reputation. This undesirable behavior is even more critical as the framework operates in an open environment where new agents can enter the community at any time, thus requiring a new boot phase for any additional agent and action.

3.3.2. Window Mode

In *Window mode*, similarly to the *Boot mode*, observations are used to compute an agent's *Reliability* only after sufficient data have been collected. However, differently from the *Boot mode*, *Reliability* and *Verification Trustworthiness* are computed by taking into account only the last $N_{win}(k, i, j)$ times that G_i has observed G_j executing A_k, where $N_{win}(k, i, j)$ is the length of the "memory window" (instead of considering the full sequence of observations). The *Reliability* and *Verification Trustworthiness* formulas (1) and (3) can be changed accordingly by computing the sum from $n = N_{exe}(k, i, j) - N_{win}$ to $n = N_{exe}(k, i, j)$ (instead of starting from $n = 1$). If the available data is less than the length of the memory window, the agents behave as they do in the *Boot mode*.

Window mode is appropriate when the performance of an agent may improve or deteriorate as time passes, therefore allowing other agents to quickly detect changes and update its *Reliability* accordingly. However, similarly to the *Boot mode*, defining a proper length for the memory window is crucial. A small window length guarantees that a possible overly confident behaviour of bidders will last fewer interactions. Still, the *Reliability* and the *Verification Trustworthiness* will be subject to oscillations and never stabilize. On the other hand, a considerable window length will produce robust *Reliability* and *Verification Trustworthiness* estimations, possibly converging to the actual error rate

of the agent. However, this will need more time to reach a steady state and be susceptible to undesirable behaviours in the transitory phase.

3.3.3. Binomial Confidence Interval (BCI) Mode

We propose the *BCI mode* to remedy the flaws of the previous two modes. The model prevents the emergence of unforgiving behaviors, and it neither needs a boot phase nor a memory window. As we modeled the results of each action as a *success* or a *failure*, we can formalize the results' sequence as a Bernoulli process. In *BCI mode*, each agent computes and shares not only the average *Reliability* $Rel(k, i, j)$ of other agents, but also the binomial confidence interval (BCI) around such estimate that converges to zero as the number of execution increases. Specifically, the upper $High(k, i, j)$ and lower bounds $Low(k, i, j)$ of the interval depend on three parameters: the number of Bernoulli or Binomial trials, which in our framework translates to the number of times N_{exe} that the action has been observed; the number of times a *success* has been observed; a *confidence percentage* in the range $[0, 100]$ that represents the desired statistical chance that the actual value of the *Reliability* falls in the confidence interval[2].

Auctioneers will then use the confidence interval as a piece of additional information to evaluate the trustworthiness of bidders. When G_i auctions action A_k, the self-declared *Reliability* $Rel(k, j, j)$ of an agent G_j is now required only when no data are available at all. Indeed, as soon as G_j has executed A_k at least once, G_j's *Reliability* can be safely used given that we implement policies to take into account not only its value but the confidence G_i has about that value (given by the BCI). We will show in the next Section how G_i's *optimistic/pessimistic/realistic* attitude may be used to interpret the confidence interval in different ways.

3.4. Disposition towards other agents

The *optimistic/pessimistic/realistic disposition* of an auctioneer may play a key role to evaluate a bidder's trustworthiness. When in *Boot mode* and *Window mode*, for instance, the *disposition* of an auctioneer G_i may be used to determine the *Reliability* of a bidder G_j until a sufficient number data has been collected. In Subsection 3.3 we explained that, during the transitory phase, the declared *Reliability* of G_j can be used, but a more sophisticated behavior might be the following.

Suppose that G_i has not yet enough data for G_j about A_k, but it has collected enough data about other actions $\{A_l\}$, $l \neq k$:

- an *optimistic* auctioneer may consider instead the highest *Reliability* among the observed actions A_l, $l \neq k$:

$$Comp(k, i, j) = \arg\max_{l \neq k} Rel(l, i, j); \quad (6)$$

- a *realistic* auctioneer may compute a weighted average over all the observed actions A_l, $l \neq k$:

$$Comp(k, i, j) = \frac{\sum_{l \neq k} N_{exe}(l, i, j) Rel(l, i, j)}{\sum_{l \neq k} N_{exe}(l, i, j)}; \quad (7)$$

- a *pessimistic* auctioneer may consider the lowest *Reliability* among the observed actions A_l, $l \neq k$:

$$Comp(k, i, j) = \arg\min_{l \neq k} Rel(l, i, j). \quad (8)$$

The strategy just described is not implemented in the experiments in Section 5, where the declared *Reliability* is used in *Boot Mode* and *Window Mode* until enough data have been collected. But the example shows that more complicated strategies are possible, which somehow mimic the intuitive behaviour we would expect from a *optimistic/pessimistic/realistic* agent.

The *disposition* of agents plays a key role in our experiments when data are collected and processed in *BCI mode*. We remind that, in this case, a BCI $[Low(k, i, j), High(k, i, j)]$ is computed by G_i around the estimated *Reliability* of G_j in performing A_k. The BCI measures how confident G_i is about its estimate $Rel(k, i, j)$, which converges to its expected value as the number of observations increases. Under these conditions:

- an *optimistic* auctioneer will consider the upper bound of the BCI to estimate the agent's *Perceived Competence*, therefore being more prone to forgive and give a second chance to an agent that failed the first attempts:

$$Comp(k, i, j) = High(k, i, j); \quad (9)$$

[2]We use the EBCIC Python module from the National Institute of Advanced Industrial Science and Technology, which implements the Clopper–Pearson interval [53].

- a *pessimistic* auctioneer will use the lower bound, thus being very conservative and cautious when it encounters a new bidder about which it has little data:

$$Comp(k,i,j) = Low(k,i,j); \tag{10}$$

- a *realistic* auctioneer will use the intermediate value $Rel(k,i,j)$, thus being open to new bidders but, at the same time, not very willing to give a second chance to an agent that failed.

Irrespective of the dynamic behaviour of the system, an *optimistic/pessimistic/realistic* disposition may play an important role in computing the gain K in Eq. (5). K determines how likely it is that the *Perceived Competence* of a bidder G_j is above the threshold and the agent will be assigned an action. An *optimistic* auctioneer will likely use a lower gain, a *pessimistic* auctioneer will use a higher gain, and a realistic auctioneer will have a value in-between the two. This rule translates in *optimistic* agents being more prone to trust an agent and *pessimistic* agents being more cautious, possibly asking a human supervisor when in doubt.

3.5. Behaviour toward the community

The difference between *individualistic* and *collectivistic* agents is not in the way *Reliability* is computed, but how the individual *Reliability* and *Verification Trustworthiness* values estimated by agents are composed to compute the *Perceived Competence* in Eq. (4).

In case of an *individualistic* auctioneer G_i that needs to compute the *Perceived Competence* of G_j in executing A_k, we define:

$$Comp(k,i,j) = Rel(k,i,j), \tag{11}$$

where, as usual, the value $Rel(k,i,j)$ depends on the approach adopted to estimate trust metrics in the transitory (*Boot mode*, *Windows mode*, *BCI mode*). Describing the behaviour of a *collectivistic* agent is more complex since it requires introducing the concept of *Weighted Reliability*. The idea is that, at the beginning of the auction for action A_k, all agents share their own opinions about each other's *Reliability* in executing that action: a *collectivistic* auctioneer will take into account other agents' opinions by calculating a weighted average mediated by their *Verification Trustworthiness*. Thanks to this, *collectivistic* agents can compensate for a poor ability to judge the reliability of other agents by relying on the opinions of agents that are more trustworthy in verifying actions.

Specifically, the *Weighted Reliability* is computed as follows:

$$WRel(k,j) = \frac{\sum_l^{N_w} VTW(k,l)Rel(k,l,j)}{\sum_l^{N_w} VTW(k,l)} \tag{12}$$

by considering in the sum the N_w agents G_l that have an opinion $Rel(k,l,j)$ about G_j's *Reliability* in performing action A_k. Notice that $WRel(k,j)$ has the same value when computed by any of the *collectivistic* agents G_i, since all agents agree about the *Verification Trustworthiness* $VTW(k,l)$ of G_l in verifying the execution of A_k, Eq. (3).

The *Perceived Competence* can be computed as:

$$Comp(k,i,j) = WRel(k,j). \tag{13}$$

To summarize, in the case of *individualistic* agents, each agent G_i individually estimates the reliability $Rel(k,i,j)$ of G_j in performing A_k, and then the *Perceived Competence* of G_j in performing A_k is computed as in Eq. (11). In the case of *collectivistic* agents, each agent G_i first estimates the weighted reliability $WRel(k,j)$ by considering the opinion of other agents as in Eq. (12); then it computes the *Perceived Competence* as in Eq. (13).

4. Implementation

We implemented the Trust Framework in ROS [19], thus providing the shared vocabulary and protocol required by agents to communicate with each other in an open, distributed environment. The framework follows a modular design to run both in simulation and with real robotic platforms.

A sketch of the proposed architecture is shown in Figure 1. The *TrustAgent node* is responsible for all the framework-related operations (such as communications with other framework members, auction management, and trust evaluation). The *Adapter* manages the interface with platform-dependent sensorimotor routines related to perception and action and needs to be customized for any specific robotic platform, thus guaranteeing modularity and compatibility.

4.1. Agent Node

In the framework, each agent is represented by a node, which provides the following services:

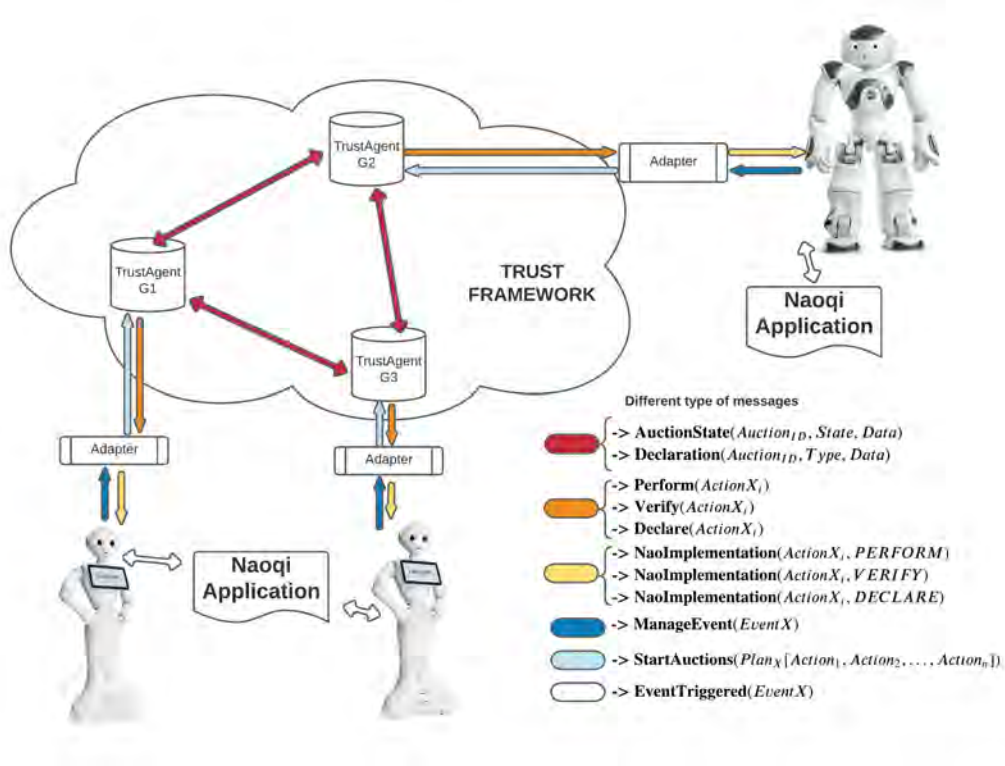

Figure 1: A possible implementation of the proposed architecture.

- management of auctions either as an auctioneer or a bidder;
- handling of ROS messages through which the agents receive and elaborate data from other agents;
- synchronization with other agents through all the phases of an auction;
- processing of all the data required for the computation of *Reliability* and *Verification Trustworthiness*;
- evaluation of the *Perceived Competence* depending on the behaviour-disposition configuration of the agent;
- sending/receiving data, through a TCP/IP socket, to the *Adapter* running on the robot to perform the required sensorimotor routines for executing or verifying an action.

4.2. Event node

In a real-world scenario, events may be triggered by the robot's onboard sensors, by an alarm, or by an explicit request of the user. Ideally, each detected event will trigger one or more plans known to the agent, where each plan includes the actions to be completed to achieve the goal. The agent will then be responsible for auctioning the actions. However, for the experiments described in the following, we assume that event triggering is fully addressed by a node referred to as the *Event node*. The *Event node* periodically generates an event by publishing a message targeted to the only agent that has a plan to manage it and therefore will be the auctioneer for that particular kind of event. In real-world experiments, the *Event Node* is useful to produce repeated events, which would otherwise require continuous interaction with robots to test the functionalities of the framework.

4.3. Simulated results node

The *Simulated results node* is a service only needed in simulation. When an agent needs to execute an action or to verify the execution of an action performed by another agent, it sends a request to the *Simulated results node*. The latter will then decide for the *success* or *failure* of the action or

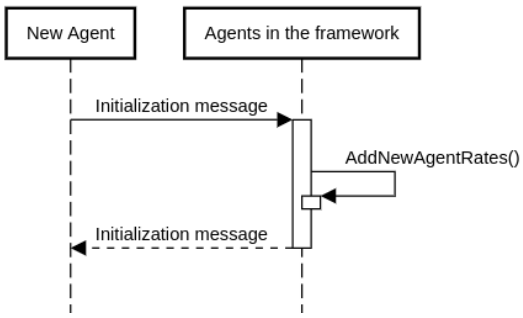

Figure 2: A new agent enters the framework.

the observation, depending on a priori probabilities associated with different possible outcomes. This process simulates the result of an action or an observation as it would happen in the real world, which should not be confused with the *Reliability* and *Verification Trustworthiness* estimated by agents. In real-world tests, we do not require this node because the *success* or *failure* of an action or an observation depends on the actual perceptual or actuation capability of a robot. After computing the outcome, the service sends back to the requesting agent a response. Notice that a successful observation correctly reports the *success* or *failure* of the action, whereas a failed observation reports a *failure* when the action was a *success* and vice versa.

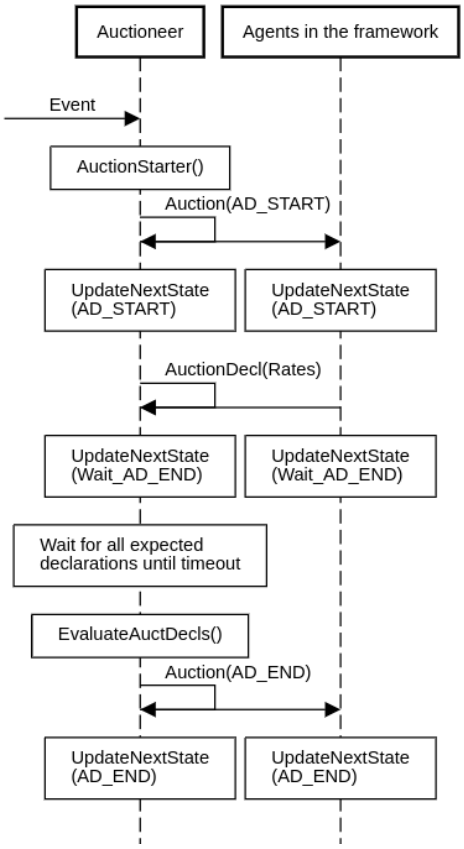

Figure 3: Agents' interaction during the auction phase.

4.4. Framework sequence diagrams

To clarify the temporal sequencing of the interactions between agents in the framework, we provide three sequence diagrams illustrating how information is exchanged among agents. Figure 2 describes what happens when a new agent enters the framework. First, it broadcasts an initialization message communicating the new agent's plans and actions and its self-declared success rate in performing such activities. Upon receiving this message, other agents update their data structures (`AddNewAgentRates()`) and reply to the new agent by publishing their own initialization messages.

Figure 3 describes the auction phase after an event is triggered. First, the auctioneer publishes a message (`Auction(AD_START)`). Then, each agent responds with a message (`AuctionDecl(Rates)`) containing the estimated *Reliability* and *Verification Trustworthiness* of all possible candidates. The auctioneer waits for all expected declarations to arrive or until a timeout expires. Then, it chooses the winner depending on its own opinion and, possibly, the opinions of other agents if its behavior-disposition configuration requires that. Finally, the auctioneer publishes a message (`Auction(AD_END)`) to declare the end of the auction, the winner, and the verifying agents among the participants.

Figure 4 shows the action execution and verification phase. After the auctioneer declares the winner, the agents in the framework prepare to either perform or verify the auctioned action. When a verifying agent is ready, it publishes a message (`Action(READY_for_Act)`) and starts waiting for the winner to execute the action. The winner waits for all the expected messages to arrive or until a timeout expires and then starts executing the action, after notifying the other agents through an additional message (`Action(READY_for_Act)`). In this way, the performer and the verifying agents synchronize execution and verification of the action. When the performer terminates the action, it notifies the other agents (`Action(Act_END)`), after which each verifying agent shares the perceived outcome (`ResultDecl(perceived_outcome)`) and

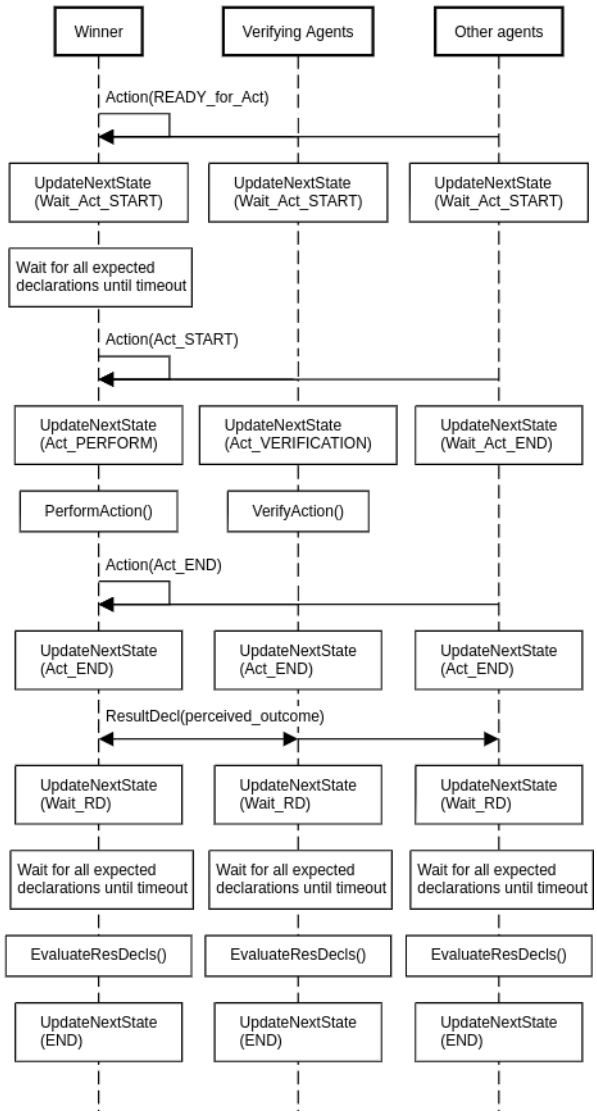

Figure 4: Agents' interactions during the execution and verification phase

waits for analogous messages from other verifying agents. All agents wait for all the expected declarations to arrive or until a timeout expires, and then accordingly evaluate the trust metrics. Finally, they all label the auction as terminated and are ready for another auction.

5. Simulated experiments

To validate the proposed framework, several experiments have been performed in simulation, a small part of which is described in the following.

For each experiment, we provide the actual *success* rate of each agent G_i to perform actions, as well as the actual rate of True Positive (TP) and True Negative (TN) in observing the outcome of actions performed by other agents (or by themselves). The *Simulated result node* uses these values to simulate the outcomes of the execution and verification of actions. Then, for each agent G_i we build the following tables and plots to describe the results (not all tables and plots are reported for each experiment for lack of space).

- A summary table reporting, for each agent G_j of which G_i is aware (including G_i itself), the final value of the estimated *Reliability* of G_j, the observed number of *successes* N_S and *failures* N_F of G_j in performing each action, the final value of G_j's *Verification Trustworthiness*, the number of times N that each action has been verified by G_j.

- A plot reporting, for each agent G_j of which G_i is aware (including G_i itself), how the estimated *Reliability* and *Verification Trustworthiness* evolve as the number of auctions increases. In all plots, *Reliability* estimates are plotted with a continuous line, whereas *Verification Trustworthiness* estimates are plotted with a dashed line. A small vertical dash means that an action has been assigned to the corresponding agent at that time.

All the experiments reported below (except the last set of experiments in Section 5.5 use the following Event-Plan association: event E_1 can only be handled by G_1, that will then auction A_1; event E_2 can only be handled by G_2, that will then auction A_2 and A_3 in sequence. All agents can execute and observe all actions A_1, A_2, A_3, even if they may have different *success*, TP, and TN rates in different tests.

5.1. Reliability estimation

This set of tests does not have the objective to simulate a real-world scenario but only to assess the proper *Reliability* estimation. All agents are *individualistic* and work in *Boot mode* for trust metrics update, which does not require to specify an *optimistic/pessimistic/realistic* configuration. The boot phase lasts 50 interactions for each agent and action, and the TP and TN rates are set to 100% for all agents, which means perfect observations.

5.1.1. Experiment 1.1: one perfect agent, two agents overestimating their capabilities

Three agents G_1, G_2 and G_3 are deployed. G_1 has a 100% success rate for all actions and is aware of that; G_2 and G_3 have a 70% success rate but initially estimate to be perfect, i.e., $Rel(k, 2, 2) = Rel(k, 3, 3) = 1$, $\forall k$ (agents' estimate of other agents' and their own *Reliabiliy* will change during execution.) Since agents always judge the outcomes correctly, only G_1's trust metrics are reported.

As expected from this configuration, the *Reliability* values in Table 1 and Figure 5 tend to 1 for each action performed by the perfect agent G_1 whereas they tend to 0.7 for G_2 and G_3. In particular, it is possible to observe in Figure 5 and in Table 1 that actions are distributed among all agents during the boot phase since G_2 and G_3 overestimate their *Reliability*, which is used during initial auctions: the sum N_S+N_F corresponding to G_2 and G_3 for all actions equals 50. As soon as this phase ends and *Reliability* is estimated based on observations, actions are always assigned to the most trustable agent G_1.

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	1.00 (233; 0)	1.00 (333)	1.00 (233; 0)	1.00 (333)	1.00 (233; 0)	1.00 (333)
G_2	0.72 (36; 14)	1.00 (333)	0.74 (37; 13)	1.00 (333)	0.78 (39; 11)	1.00 (333)
G_3	0.64 (32; 18)	1.00 (333)	0.78 (39; 11)	1.00 (333)	0.64 (32; 18)	1.00 (333)

Table 1: Trust metrics at the end of the Experiment 1.1 according to G_1 (G_2 and G_3 would return the same values).

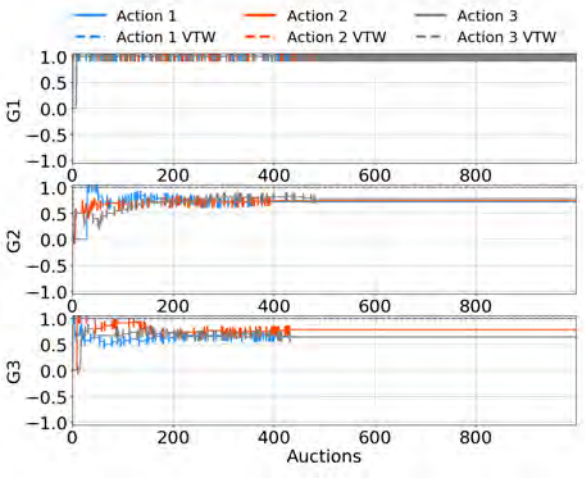

Figure 5: Trust dynamics of the Experiment 1.1 according to G_1 (G_2 and G_3 would return the same values).

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	0.87 (203; 30)	1.00 (333)	0.52 (26; 24)	1.00 (333)	0.28 (14; 36)	1.00 (333)
G_2	0.26 (13; 37)	1.00 (333)	0.91 (211; 22)	1.00 (333)	0.58 (29; 21)	1.00 (333)
G_3	0.62 (31; 19)	1.00 (333)	0.28 (14; 36)	1.00 (333)	0.93 (216; 17)	1.00 (333)

Table 2: Trust metrics at the end of the Experiment 1.2 according to G_1 (G_2 and G_3 would return the same values).

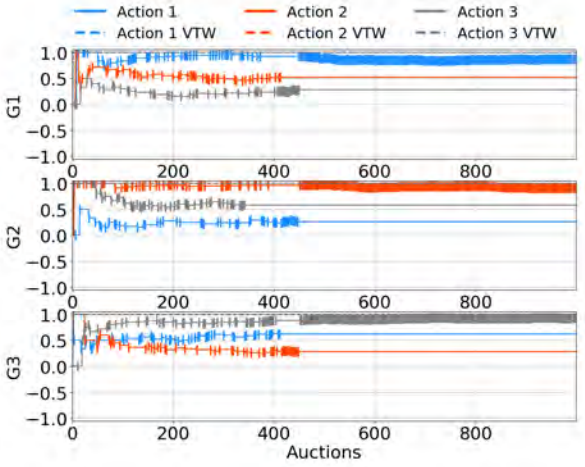

Figure 6: Trust dynamics of the Experiment 1.2 according to G_1 (G_2 and G_3 would return the same values).

5.1.2. Experiment 1.2: three imperfect and overestimating agents

Three agents are deployed, each having an action in which it performs better than the others, but not succeeding every time: G_1 has success rates 90%, 60% and 30%, respectively, in A_1, A_2, A_3; G_2 has success rates 30%, 90% and 60%; G_3 has success rate 60%, 30% and 90%. All agents initially estimate to be perfect in execution, i.e., $Rel(k, i, i) = 1$, $\forall k, i$. Since agents always judge outcomes correctly, only G_1's trust metrics are reported.

As expected from this configuration, the *Reliability* in Table 2 and Figure 6 tends to the actual success rate for all agents. Table 2 shows that actions have been distributed among all agents during the boot phase, after which each action is always assigned to the most trustable agent: this can be easily inferred by summing up $N_S + N_F$ for each agent and action.

5.2. Verification trustworthiness

This set of tests has the main purpose of assessing the proper estimation of the *Verification Trustworthiness* of agents. Once again, agents are *individualistic* and configured to work

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	1.00 (106; 0)	0.33 (333)	1.00 (106; 0)	0.33 (333)	1.00 (112; 0)	0.33 (333)
G_2	1.00 (116; 0)	0.33 (333)	1.00 (102; 0)	0.33 (333)	1.00 (115; 0)	0.33 (333)
G_3	1.00 (111; 0)	-0.33 (333)	1.00 (125; 0)	-0.33 (333)	1.00 (106; 0)	-0.33 (333)

Table 3: Trust metrics at the end of the Experiment 2.1 according to G_1 (G_2 would return the same values).

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	0.00 (0; 106)	0.33 (333)	0.00 (0; 106)	0.33 (333)	0.00 (0; 112)	0.33 (333)
G_2	0.00 (0; 116)	0.33 (333)	0.00 (0; 102)	0.33 (333)	0.00 (0; 115)	0.33 (333)
G_3	0.00 (0; 111)	-0.33 (333)	0.00 (0; 125)	-0.33 (333)	0.00 (0; 106)	-0.33 (333)

Table 4: Trust metrics at the end of the Experiment 2.1 according to G_3.

in *Boot mode*, and then do not require an *optimistic/pessimistic/realistic* configuration. The boot phase lasts 50 interactions, and the success rates are set to 100% for all agents, which means perfect execution of all actions.

5.2.1. Experiment 2.1: two perfect agents and an always wrong agent

Three agents G_1-G_3 are deployed: G_1 and G_2 have 100% TP and TN rates in recognizing the outcomes of an action, whereas G_3 has 0% TP and TN rates, i.e., it always observes the opposite of the actual result. This time the observations of G_1 and G_2 are different from G_3, and therefore both G_1 and G_3's trust metrics are reported.

As shown in Tables 3, 4 and Figures 7, 8, the *Reliability* of all agents estimated by G_1 and G_2 tends to 1, whereas the *Reliability* estimated by G_3 equals 0 since the latter always observes a *failure* when the action was a *success* and vice versa. Moreover, it is possible to observe that the *Verification Trustworthiness* of the perfect agents is not 1, even if their actual TP and TN rates are 100%. This happens because the consensus in observations can never be achieved: one-third of the agents are always voting the opposite of the other two-thirds. Finally, it is possible to notice that actions are almost equally distributed among agents during the whole simulation: the auctioneers G_1 and G_2 know that, even if G_3 is very bad in judging the outcomes of actions, it still succeeds in executing them perfectly.

5.2.2. Experiment 2.2: nine perfect agents and an always wrong agent

This experiment is similar to the previous one. In this case, a higher number of agents G_1-G_9 with 100% TP and TN rates have been added to assess

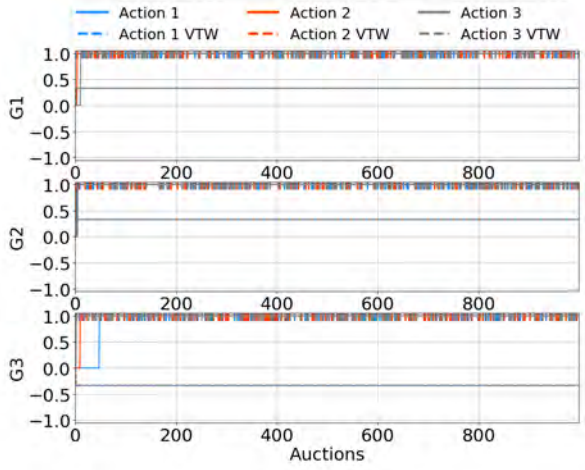

Figure 7: Trust dynamics of the Experiment 2.1 according to G_1 (G_2 would return the same values).

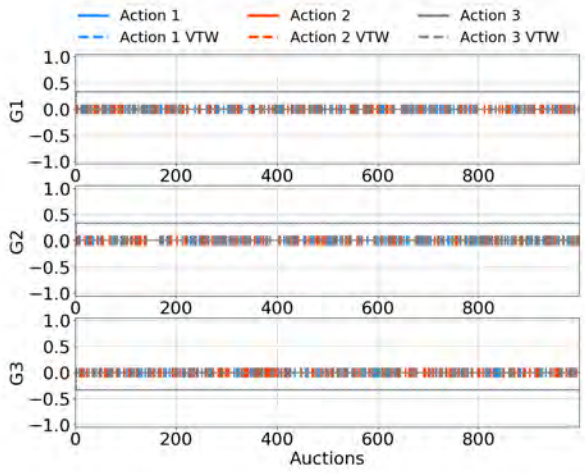

Figure 8: Trust dynamics of the Experiment 2.1 according to G_3.

that their *Verification Trustworthiness* increases as the consensus in observation increases.

Metrics are not reported for sake of brevity, however, they are very similar to the ones reported in the previous experiment. The only difference is in the *Verification Trustworthiness* estimation: this time, it converges to 0.8 for agents G_1-G_9 due to the higher consensus (9 of the 10 agents always agree), whereas it converges to -0.8 for G_{10}. As in the previous test, actions are almost equally distributed among agents, since auctioneers G_1 and G_2 estimate all agents to be equally reliable.

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	0.52 (120; 112)	0.35 (333)	0.58 (29; 21)	0.32 (333)	0.52 (26; 24)	0.31 (333)
G_2	0.47 (24; 27)	0.35 (333)	0.49 (26; 27)	0.30 (333)	0.53 (83; 73)	0.32 (333)
G_3	0.44 (22; 28)	0.32 (333)	0.47 (109; 121)	0.35 (333)	0.47 (60; 67)	0.29 (333)

Table 5: Trust metrics at the end of the Experiment 2.3 according to G_1

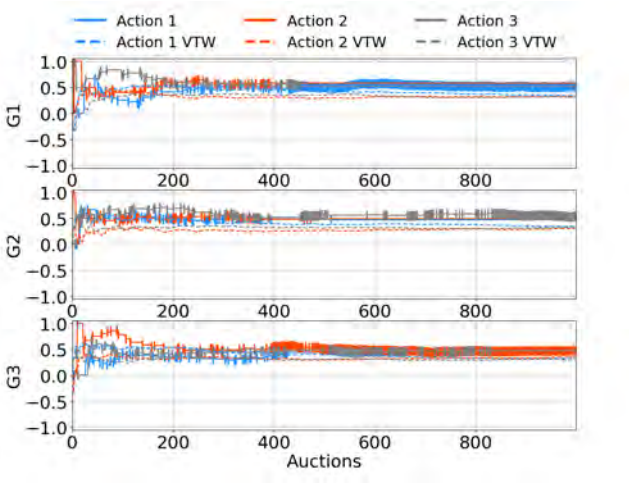

Figure 9: Trust dynamics of the Experiment 2.3 according to G_1.

5.2.3. Experiment 2.3: Three agents with a 50% chance of observing correctly

Three agents G_1-G_3 are deployed, each having a 50% TP and TN rates in evaluating the outcomes of actions. Since agents judge the outcomes of actions differently, the trust metrics evaluated by individual agents are different: however, since the results are quite similar in spite of local differences, only G_1 estimated metrics are reported.

As shown in Table 5 and Figure 9, the estimated *Reliability* tends to the actual TP and TN rates of agents: agents' actions are always a *success*, but they have a 50% chance that they are judged a *failure*. From Table 5 it can be observed that actions have not been equally distributed among agents: this is because the estimated *Reliability* now oscillates around 0.5 depending on how other agents judge outcomes, and auctioneers may repeatedly assign actions to an agent even if its *Reliability* is only slightly superior. As all agents have the same TP and TN rates, *Verification Trustworthiness* values tend to be almost the same.

5.3. Transitory behaviour

In this set of tests the agent's transitory behaviour (*Boot mode/Window mode/BCI mode*) changes in each experiment whereas the success, TP and TN rates are always the same: G_1 has success rate 100% but underestimates its capabilities as $Rel(k, 1, 1) = 0.8$, $\forall k$; G_2 and G_3 have success rate 70% but overestimate their capabilities as $Rel(k, i, i) = 1$, $\forall k, i = 2, 3$. The TP and TN rates in observing actions are 100% for all agents: agents exhibit an *individualistic* behaviour.

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	1.00 (233; 0)	1.00 (333)	1.00 (233; 0)	1.00 (333)	1.00 (233; 0)	1.00 (333)
G_2	0.68 (34; 16)	1.00 (333)	0.70 (35; 15)	1.00 (333)	0.72 (36; 14)	1.00 (333)
G_3	0.72 (36; 14)	1.00 (333)	0.64 (32; 18)	1.00 (333)	0.60 (30; 20)	1.00 (333)

Table 6: Trust metrics at the end of the Experiment 3.1 according to G_1 (G_2 and G_3 would return the same values).

5.3.1. Experiment 3.1: Boot Mode

As like as in previous experiments, the boot phase length is set to 50 interactions. Since agents always judge outcomes correctly, only G_1's trust metrics are reported.

Table 6 and Figure 10 show that, during the boot phase, actions are almost equally distributed between the "arrogant" G_2 and G_3 that overestimate their *Reliability*, but they are never assigned to G_1 that declares its own *Reliability* to be lower. However, after the boot phases of G_2 and G_3 have ended, auctioneers realize that G_2 and G_3's *Reliability* is 0.7, i.e., lower than initially thought. They immediately start assigning actions to G_1, since the latter is still in its boot phase and has a declared *Reliability* equal to 0.8. Auctioneers keep on assigning actions to G_1 even after the boot phase has ended, as they soon realize that it has a 100% success rate.

The same would not happen if G_1's declared *Reliability* were lower than the actual success rate of G_2 and G_3, say $Rel(k, 1, 1) = 0.6$, $\forall k$. In this case, after the boot phases of G_2 and G_3 have ended, G_1 will never be given a chance to win an auction. As a consequence, auctioneers will never make any observation to update their estimate about the actual G_1's success rate: in the future, G_1's *Reliability* will always be considered the lowest one, and no agent will ever trust G_1 even if it is the most performing agent. This behaviour is shown in Figure 11.

5.3.2. Experiment 3.2: Window Mode

Agents use the *Window mode* for estimating *Reliability* with a window length of 50 observations. As discussed in Section 3.3, using the *Window mode* is

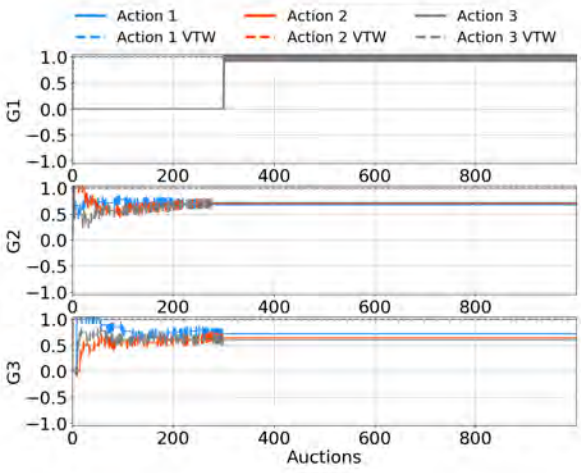

Figure 10: Trust dynamics of the Experiment 3.1 when $Rel(k, 1, 1) = 0.8$, $\forall k$, according to G_1 (G_2 and G_3 would return the same values).

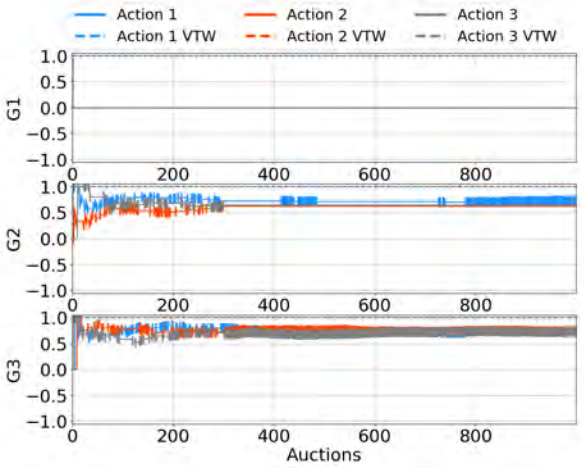

Figure 11: Trust dynamics of the Experiment 3.1 when $Rel(k, 1, 1) = 0.6$, $\forall k$, according to G_1 (G_2 and G_3 would return the same values).

crucial when the success rate of an agent may increase or decrease as time passes, therefore allowing other agents to quickly detect changes and update metrics accordingly. Since the success rate of all agents is constant in this set of experiments, we observe that the behaviour of the *Window mode* is almost the same as the *Boot mode*, and therefore metrics are not reported for sake of brevity.

5.3.3. Experiment 3.3: BCI Mode

Agents use the *BCI mode* for estimating *Reliability* with confidence equal to 90%, meaning that

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	1.00 (306; 0)	1.00 (333)	1.00 (329; 0)	1.00 (333)	1.00 (308; 0)	1.00 (333)
G_2	0.50 (2; 2)	1.00 (333)	0.33 (1; 2)	1.00 (333)	0.81 (17; 4)	1.00 (333)
G_3	0.83 (19; 4)	1.00 (333)	0.00 (0; 1)	1.00 (333)	0.50 (2; 2)	1.00 (333)

Table 7: Trust metrics at the end of the Experiment 3.3 according to G_1 (G_2 and G_3 would return the same values).

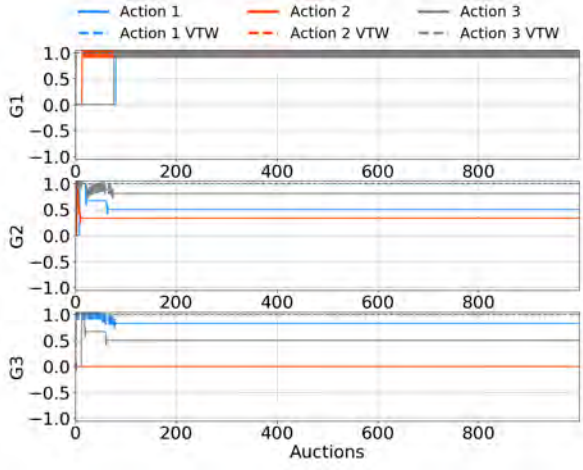

Figure 12: Trust dynamics of the Experiment 3.3 according to G_1 (G_2 and G_3 would return the same values).

the actual success rate has a 0.9 probability of being in the BCI around the estimated *Reliability*. In *BCI mode* a *optimistic/pessimistic/realistic* disposition may play a key role in determining different behaviours. We set agents' disposition to *optimistic* because, to overcome the limitations of *Boot mode* and *Windows mode*, auctioneers shall give all agents a reasonable chance to perform at least one action when they enter the framework. Additionally, a *pessimistic* attitude may be impractical, since auctioneers will repeatedly ask for a human supervisor's approval in the initial phase, as long as the *Perceived Competence* of bidders is below the threshold in Eq. (5). This would happen if *Reliability* were pessimistically estimated as the lower bound of the BCI, which is necessarily large when the number of observations is small. Please, remember also that *BCI mode* does not require bidders' declared *Reliability*. Since agents always judge outcomes correctly, only G_1's trust metrics are reported.

It is possible to notice from Table 7 and Figure 12 that actions have been mainly assigned to G_1 since the beginning. This happens even if auctioneers initially overestimate G_2 and G_3's *Reliability*

(please remember that the BCI of all agents is initially very large, and the upper bound equals 1). As soon as G_2 and G_3 fail a few times, their *Reliability* decrease and the upper bound of their BCI becomes lower than G_1's upper bound: since auctioneers are *optimistic*, G_1 starts winning auctions. As more actions are assigned to G_1, its estimated *Reliability* increases and the BCI around it becomes smaller. As time passes, G_1 will tend to win all auctions since its success rate will be correctly estimated as the highest one with increasing confidence.

5.4. Behavior-disposition test

In this set of experiments, the differences between different configurations in terms of behavior-disposition (*collectivistic/individualistic*, *optimistic/pessimistic/realistic*) are compared. All of the following experiments are made in *BCI* mode using 90% as a confidence parameter.

5.4.1. Experiment 4.1: a collectivistic agent and an individualistic agent that observe poorly and three good observers

The purpose of this experiment is to check if, in *BCI* mode, auctioneers are able to compensate for poor observation capabilities by relying on other agents. The experimental setup comprises 5 agents G_1-G_5, including two auctioneers G_1 and G_2 that verify with a 50% TP and TN rate and three agents G_3, G_4, G_5 with a 100% TP and TN rates. Concerning success rate in execution, G_1 and G_2 have a 50% success rate in all actions, whereas G_3, G_4 and G_5 have a 60%, 75% and 90% success rate, respectively. G_1 is an *individualistic-optimistic* agent whereas G_2 is a *collectivistic-optimistic* agent: the hypothesis is that the *collectivistic* G_2 will be able to identify the most trustable agent in the framework G_5 by relying on the opinions of other agents, whereas the *individualistic* G_1 will not.

To verify this hypothesis, it is sufficient to check Table 8 summarizing G_1's trust metrics, which are very similar to the metrics computed by G_2 (not shown) and Table 9 summarizing G_3's trust metrics, which are identical to G_4 and G_5's. As expected, the *individualistic* agent G_1 fails to compensate for its poor verification capabilities: in Table 8 it is possible to observe that A_1, auctioned by G_1, is assigned to different agents multiple times, because G_1 is not able to correctly identify G_5 as the most suitable one (G_5 is assigned A_1 only $N_S + N_F = 3$ times out of 333). On the other hand, the *collectivistic* agent G_2 is able to identify G_5 as the best

	A_1		A_2		A_3	
	Rel, N_S, N_F	VTW, N	Rel, N_S, N_F	VTW, N	Rel, N_S, N_F	VTW, N
G_1	0.27 (3; 8)	0.16 (333)	0.67 (2; 1)	0.21 (333)	1.00 (2; 0)	0.19 (333)
G_2	0.47 (37; 42)	0.20 (333)	0.44 (4; 5)	0.20 (333)	1.00 (2; 0)	0.20 (333)
G_3	0.47 (45; 50)	0.58 (333)	0.50 (2; 2)	0.60 (333)	0.50 (11; 11)	0.59 (333)
G_4	0.53 (77; 68)	0.58 (333)	0.40 (2; 3)	0.60 (333)	0.54 (7; 6)	0.59 (333)
G_5	0.00 (0; 3)	0.58 (333)	0.49 (153; 159)	0.60 (333)	0.48 (141; 153)	0.59 (333)

Table 8: Trust metrics at the end of the Experiment 4.1 according to G_1.

	A_1		A_2		A_3	
	Rel, N_S, N_F	VTW, N	Rel, N_S, N_F	VTW, N	Rel, N_S, N_F	VTW, N
G_1	0.45 (5; 6)	0.16 (333)	0.33 (1; 2)	0.21 (333)	0.00 (0; 2)	0.19 (333)
G_2	0.57 (45; 34)	0.20 (333)	0.56 (5; 4)	0.20 (333)	0.00 (0; 2)	0.20 (333)
G_3	0.68 (65; 30)	0.58 (333)	0.25 (1; 3)	0.60 (333)	0.77 (17; 5)	0.59 (333)
G_4	0.72 (104; 41)	0.58 (333)	0.40 (2; 3)	0.60 (333)	0.69 (9; 4)	0.59 (333)
G_5	1.00 (3; 0)	0.58 (333)	0.90 (281; 31)	0.60 (333)	0.92 (270; 24)	0.59 (333)

Table 9: Trust metrics at the end of the Experiment 4.1 according to G_3 (G_4 and G_5 would return the same values).

agent by ignoring misleading verifications made by itself, and using the *Verification Trustworthiness* of G_3 - G_5 to weigh declarations made by bidders: as a result, after the initial steps, A_2 and A_3 are almost always assigned to G_5 (312 and 294 times out of 333). Table 9 shows that G_3 - G_5 estimate G_5's reliability very close to 0.9 for all actions, whereas the same cannot be observed in Table 8. However, the collectivistic agent G_2 will use the *Weighted Reliability* during auctions, thus giving more credit to the opinion of G_3 - G_5 than its own opinion.

5.4.2. Experiment 4.2: a collectivistic agent and an individualistic agent that observe well and three poor observers

The goal of this experiment is to assess the performances of the *BCI mode* in environments in which the majority of the agents have poor verification capabilities. The experimental setup comprises 5 agents G_1-G_5, including two auctioneers G_1 and G_2 that verify with a 100% TP and TN rate and three other agents G_3-G_5 in two different configurations, separately tested: (i) G_3-G_5 have a 50% TP and TN rate; (ii) G_3-G_5 have a 0% TP and TN rate (they always report the opposite than the truth). Concerning success rate in execution, they are 50% for G_1 and G_2; 60%, 75%, and 90% for G_3, G_4, G_5, i.e., the same as the previous experiment. G_1 is an *individualistic-optimistic* agent whereas G_2 is a *collectivistic-optimistic* agent: the hypothesis is that the *individualistic* agent G_1 relies on its own observation capability to identify the best agent in the framework G_5, whereas the performance of the *collectivistic* agent G_2 deteriorates as G_3-G_5's TP and TN rate decreases. This is because a *collectivis-*

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	0.00 (0; 2)	0.41 (333)	0.53 (9; 8)	0.40 (333)	0.38 (3; 5)	0.39 (333)
G_2	0.50 (3; 3)	0.41 (333)	0.47 (7; 8)	0.40 (333)	0.00 (0; 2)	0.39 (333)
G_3	0.50 (3; 3)	0.22 (333)	0.40 (4; 6)	0.22 (333)	0.33 (2; 4)	0.18 (333)
G_4	0.75 (30; 10)	0.20 (333)	0.77 (40; 12)	0.21 (333)	0.20 (1; 4)	0.19 (333)
G_5	0.86 (240; 39)	0.18 (333)	0.90 (215; 24)	0.20 (333)	0.91 (285; 27)	0.16 (333)

Table 10: Trust metrics at the end of the Experiment 4.2 (i) according to G_1 (G_2 would return the same values).

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	0.00 (0; 2)	0.41 (333)	0.59 (10; 7)	0.40 (333)	0.38 (3; 5)	0.39 (333)
G_2	0.67 (4; 2)	0.41 (333)	0.53 (8; 7)	0.40 (333)	0.50 (1; 1)	0.39 (333)
G_3	0.17 (1; 5)	0.22 (333)	0.40 (4; 6)	0.22 (333)	0.50 (3; 3)	0.18 (333)
G_4	0.50 (20; 20)	0.20 (333)	0.58 (30; 22)	0.21 (333)	0.80 (4; 1)	0.19 (333)
G_5	0.51 (142; 137)	0.18 (333)	0.52 (125; 114)	0.20 (333)	0.52 (163; 149)	0.16 (333)

Table 11: Trust metrics at the end of the Experiment 4.2 (i) according to G_3.

tic auctioneer G_2 may tend to trust bad observers. As in the previous case, we do not report plots of trust metrics, since the hypothesis can be validated by simply considering summarizing Tables.

For case (i), since G_1 and G_2 always observe the same outcomes, their trust metrics are reported in Table 10; G_3-G_5's trust metrics are very similar, and therefore we report only the summarizing Table 11 corresponding to G_3. For case (ii), G_1 and G_2's trust metrics are identical and reported in Table 12; G_3-G_5's trust metrics are identical as well (they always fail) and are reported in Table 13.

In case (i), where G_3-G_5 verify "by tossing a coin", the *individualistic* auctioneer G_1 correctly prefers to assign A_1 to the best agent G_5, as it relies on its always correct observation. Moreover, since G_1 and G_2's always agree in judging the results of actions, their *Verification Trustworthiness* is higher than G_3-G_5's: as a consequence, also the *collectivistic* auctioneer G_2 weigh more G_1's and its own verification to correctly assign A_2 and A_3 to G_5. This is evident when inspecting Tables 10 and 11, where A_1, A_2 and A_3 are assigned to G_5 a number of times $N_S + N_F$ equal to 279, 239, and 312 times out of 333, respectively.

In case (ii), since G_3-G_5 always agree (rather unrealistically) in reporting the opposite of the actual results, the *collectivistic* agent G_2 weigh more G_3-G_5's opinion, and it is no more capable to identify G_5 as the best agent: in Tables 12 and 13 it can be observed that A_2 and A_3 auctioned by G_2 are repeatedly assigned to G_3 ($N_S + N_F = 325$ times out of 333) and G_2 ($N_S + N_F = 317$ times out of 333), instead of being assigned to G_5. This problem does not affect the *individualistic* G_1, which keeps

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	0.00 (0; 1)	-0.20 (333)	0.00 (0; 0)	-0.20 (333)	0.00 (0; 0)	-0.20 (333)
G_2	0.00 (0; 1)	-0.20 (333)	0.00 (0; 0)	-0.20 (333)	0.49 (155; 162)	-0.20 (333)
G_3	0.00 (0; 1)	0.20 (333)	0.56 (182; 143)	0.20 (333)	0.80 (8; 2)	0.20 (333)
G_4	0.85 (44; 8)	0.20 (333)	1.00 (4; 0)	0.20 (333)	1.00 (3; 0)	0.20 (333)
G_5	0.92 (255; 23)	0.20 (333)	1.00 (4; 0)	0.20 (333)	1.00 (3; 0)	0.20 (333)

Table 12: Trust metrics at the end of the Experiment 4.2 (ii) according to G_1 (G_2 would return the same values).

	A_1		A_2		A_3	
	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$	$Rel(N_S; N_F)$	$VTW(N)$
G_1	1.00 (1; 0)	-0.20 (333)	0.00 (0; 0)	-0.20 (333)	0.00 (0; 0)	-0.20 (333)
G_2	1.00 (1; 0)	-0.20 (333)	0.00 (0; 0)	-0.20 (333)	0.51 (162; 155)	-0.20 (333)
G_3	1.00 (1; 0)	0.20 (333)	0.44 (143; 182)	0.20 (333)	0.20 (2; 8)	0.20 (333)
G_4	0.15 (8; 44)	0.20 (333)	0.00 (0; 4)	0.20 (333)	0.00 (0; 3)	0.20 (333)
G_5	0.08 (23; 255)	0.20 (333)	0.00 (0; 4)	0.20 (333)	0.00 (0; 3)	0.20 (333)

Table 13: Trust metrics at the end of the Experiment 4.2 (ii) according to G_3 (G_4 and G_5 would return the same values).

on assigning A_1 to the most trustworthy agent G_5 ($N_S + N_F = 278$ times out of 333).

5.5. Weighted Reliability and Verification Trustworthiness

Finally, we conducted a test involving many agents to assess if they can correctly estimate their ability to perform and verify actions in a complex scenario (i.e., having different success, TP, and TN rates) by adopting a collectivistic approach. To this aim, we set up a simulation with 50 agents, with only one Event (E_1), one action (A_1), and one auctioneer (G_1). Then, we performed tests by assigning each agent a different success rate in performing the action (100% success rate for G_1, 99% for G_2, ..., 51% for G_{50}) and different TP and TN rates in observing it by randomly choosing for each agent G_i a discrete value between 100% and 51% with step 1% (G_i's TP and TN rates are identical).

Given the test's focus on the agent's observation capabilities, performed using the *BCI mode*, we instructed G_1 to assign the action A_1 randomly with uniform probability to all agents, i.e., without using the estimated reliability. Evenly distributing A_1 among the 50 agents has the purpose of getting an almost equal amount of observations to assess their capabilities, i.e., preventing that, once an agent has been identified as the most reliable, G_1 will assign it all actions.

The results of the experiment confirm the validity of the proposed framework. After more than 14,000 actions, the *Weighted Reliability*, collectively computed by all agents as described in equation (12), correctly allows the auctioneer to evaluate the actual success rate of the agents. In particular, it can be observed that the computed *Weighted Reliability*

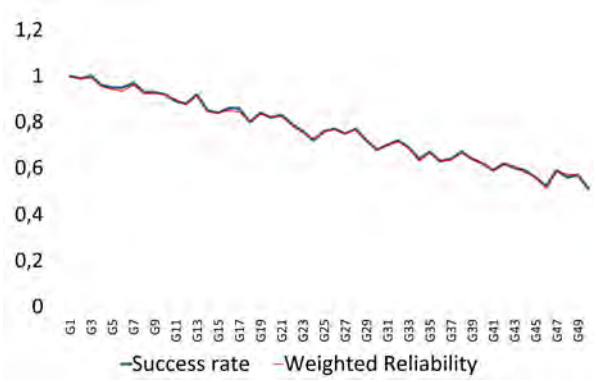

Figure 13: Normalized *Weighted Reliability* and actual success rates of each agents.

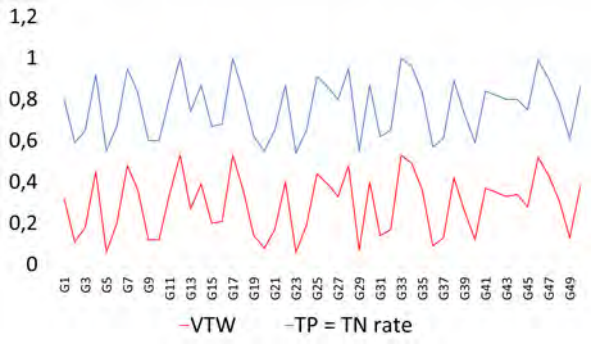

Figure 14: *Verification Trustworthiness* and actual TP=TN rate of each agent.

is directly proportional to the actual success rate of each agent measured by the agents with perfect observation capabilities, i.e., with 100% TP and TN rates (after subtracting a 0.5 offset to both). In our test, G_1 "never fails" and has a *Weighted Reliability* of 0.83, which is the highest among all agents, while the most fallacious agent G^{50} has a *Weighted Reliability* of 0.51. Figure 13 shows that the *normalized* value of the *Weighted Reliability* between 0.5 and 1.0 almost coincides with the actual success rates of the agents (please also note that the actual success rates do not perfectly match the assigned success rates because each agent performs, on average, only 280 actions). Considering the *Verification Trustworthiness* associated with the action A_1 and estimated for each agent as described in (3), it can be observed that, after subtracting an offset, it almost perfectly matches the observation rates assigned to each agent, Figure 14. Overall the agents, even with different success, TP, and TN rates, can identify other agents' capabilities to perform and observe the actions, finally allowing the auctioneer to choose the most suitable agent. Please remark that all these considerations refer to the case in which all agents have observation rates higher than 0.5, i.e., they are correct more often than they are wrong.

For the sake of completeness, we repeated the test by (i) setting all agents' success rates to 100% and their TP and TN rates randomly as in the previous test, and (ii) setting all agents' success rates as in the previous test and 100% TP and TN rates. In both cases, the results are coherent with the previous considerations: when all agents have a 100% success rate (but randomly assigned observation rates), the *Weighted Reliability* is similar for all agents (and equal to 0.83), whereas the *Verification Trustworthiness* is the one in Figure 14; when all agents have different success rates but 100% TP and TN rates, the *Verification Trustworthiness* is 1 for all agents (they always all agree about results), whereas the agents' *Weighted Reliability* matches their actual success rates (since all agents always observe actions correctly, there is no need to normalize the *Weighted Reliability* in this case).

6. Real world experiment

The framework has been tested in a real-world scenario with one SoftBank NAO and two Pepper robots. NAO has speakers, 25 motors, microphones, two RGB cameras, and 10 tactile sensors. Pepper has speakers, 20 motors, microphones, two RGB cameras, one RGB-D camera, a gyroscope, touch sensors, lasers, and sonars. To integrate NAO and Pepper in the framework, we developed *Adapters* onboard each robot to create a bridge between the NAOqi operating system and a corresponding *TrustAgent node* in ROS, Figure 1.

The purpose of these experiments is to evaluate the system's behavior in updating and using trust metrics in a real-world case. However, the aim is not to evaluate technology for action execution and recognition, for which we decided to implement a straightforward solution. Executing A_1 and A_2 corresponds to a robot saying a sentence using its embedded speakers (i.e., "*Take the medicine*" and "*Drink some water*," both reminders for the user). Verifying the correct execution of an action corresponds to a robot acquiring the audio through its microphones, translating audio to text using the NAOqi services, and finally checking that the sen-

tence was correctly pronounced. No other sensor or perceptual capability is needed in this experiment.

The robot that won the auction and pronounced the sentence always considers its action a success. However, there are many possible causes for other robots to judge outcomes differently. For instance, the speech volume, the distance between the speakers of a robot and the microphones of another robot, or the fact that NAO's and Pepper's microphones are placed over the robots' heads, which is an optimal location for human speech recognition, but suboptimal when the sound comes from a different direction. The assumption that robots always judge themselves as successful is due to practical reasons and not a limitation of the approach. In simulated experiments, all agents updated their own *Reliability* by evaluating themselves in performing actions (possibly after a boot phase): things have been simplified with real robots because it does not look relevant to make agents listen to themselves while talking.

During the auction and after the execution, we let all robots talk out loud with each other by sharing their own opinions about other robot's *Reliability* and *Verification Trustworthiness*[3]. This has no impact on the algorithm for task assignment but may help address one of the main problems in human-robot interaction mentioned in Section 2, i.e., opaqueness. By letting the robots talk out loud during the whole process, we give them the opportunity to explain to a nearby human user the reasons behind their choices.

NAO (agent G_3 in the following) is placed on a table in front of the two Pepper robots (G_1 and G_2). NAO's speakers point towards Pepper microphones from above (Figure 15). Then, we performed experiments in four different configurations:

1. The speech volume of the three robots is set to 100%, 60%, and 20% of the maximum value in different runs.
2. The volume of G_3 is fixed to 100% whereas G_1 and G_2's volume is set to 100%, 60%, and 20% of the maximum value in different runs.
3. The volume of each robot is randomly chosen in the interval 100% - 20% before each auction.
4. The volume of all robots is set to 100% but G_2 is moved 5 meters away from G_1 and G_3.

[3]A video showing an interaction between NAO and two Peppers can be found here: https://youtu.be/L_rPGg_Y888

Figure 15: NAO placed on the table to direct its speakers towards the two Peppers' microphones

In all configurations, robots work in *BCI mode*, with an *optimistic* disposition. For each configuration, robots are first configured to exhibit an *individualistic* (i) and then a *collectivistic* (ii) behavior. All the experiments reported below use the following Event-Plan association: event E_1 can only be handled by G_1, that will then auction A_1 ("*Take the medicine*"); event E_2 can only be handled by G_2, that will then auction A_2 ("*Drink some water*"). All agents can execute and observe both actions A_1 and A_2. Actions A_1 and A_2 are auctioned and executed 30 times during each experimental run.

The hypothesis we want to test with these experiments is that by modifying the robots' volume and positions and their behavior-disposition towards other robots, the assignment of actions to robots as time passes will be different. Problems in understanding can be attributed to the fact that a robot performs poorly in saying the sentence or that other robots understand badly: in both cases, not understanding what the robot said will have a consequent impact on its *Perceived Competence*.

Table 14: Experiment 1

Individualistic	100%	60%	20%
G_1	17; 1	26; 15	28; 1
G_2	8; 12	3; 14	1; 28
G_3	5; 17	1; 1	1; 1
Collectivistic	100%	60%	20%
G_1	17; 1	15; 3	10;10
G_2	4; 1	15; 26	11; 10
G_3	9; 28	0; 1	9; 10

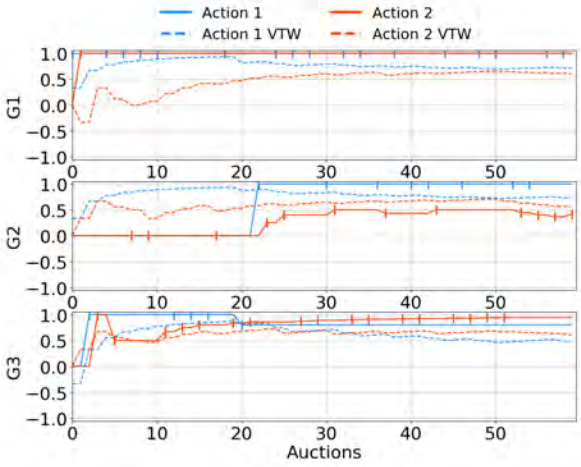

Figure 16: Trust dynamics of the Experiment 1, *individualistic*, 100% volume, according to G_1.

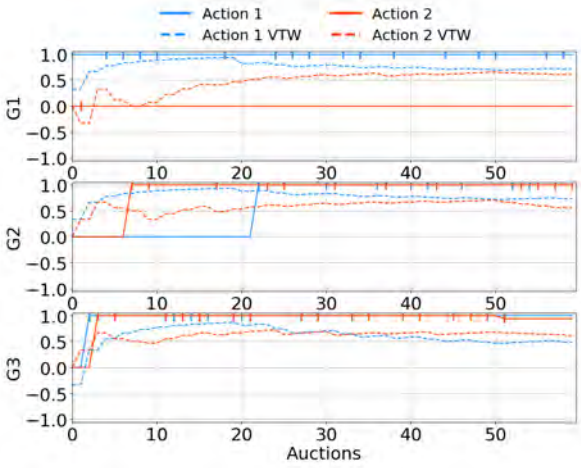

Figure 17: Trust dynamics of the Experiment 1, *individualistic*, 100% volume, according to G_2.

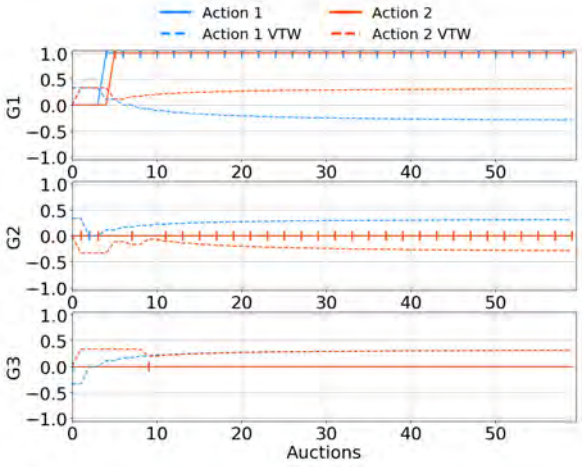

Figure 18: Trust dynamics of the Experiment 1, *individualistic*, 20% volume, according to G_1.

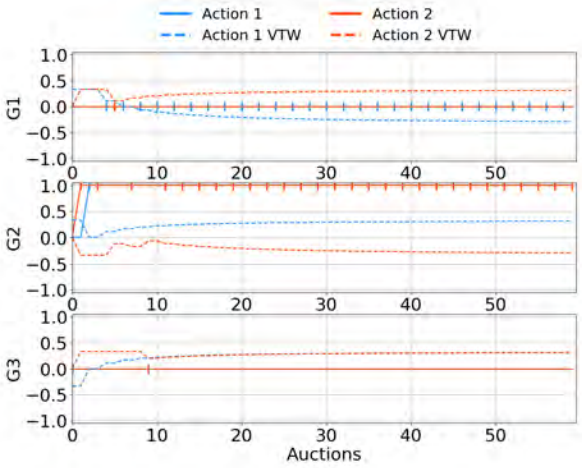

Figure 19: Trust dynamics of the Experiment 1, *individualistic*, 20% volume, according to G_2.

6.1. Experiment 1: all robots' volume set to 100%, 60%, 20%

Table 14 reports the number of actions (A_1; A_2) assigned to G_1, G_2, and G_3 with the volume set to 100%, 60%, 20%, both when robots exhibit (i) an *individualistic* and (ii) a *collectivistic* behaviour. Figures 16-19 show trust dynamics for case (i), by reporting only the metrics computed by the auctioneers G_1 and G_2 when the volume is 100% and 20%; Figures 20 - 23 report metrics for case (ii). A vertical dash helps understanding when an action has been assigned to the corresponding agent.

Table 14 shows that, when robots are *individualistic*, auctioneers assign actions to themselves with a probability of no less than one-third. Even when the volume is decreased, and therefore auctioneers are not correctly understood by other robots (which consequently judge them as unreliable), they keep considering themselves perfectly reliable. As a consequence, they assign actions to themselves and, possibly, to other robots that they judge equally reliable. Figures 16 and 17 show that, when the volume is 100%, G_1 initially shares the responsibility of executing A_1 with G_3 and then with G_2; G_2 shares the responsibility of executing A_2 with G_3. Figures 18 and 19 show that, when the volume is 20% and robots never understand what other robots are saying, auctioneers trust only themselves in per-

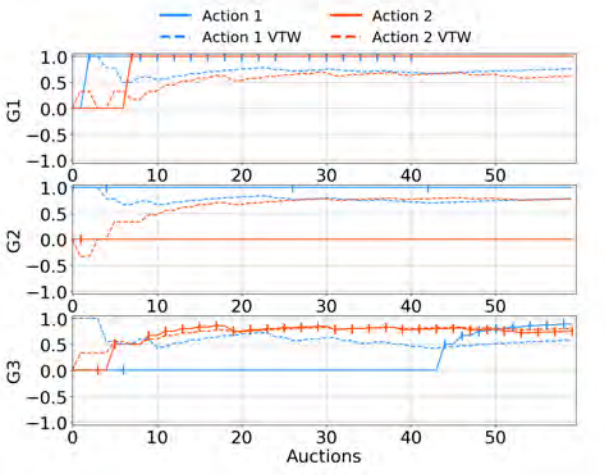

Figure 20: Trust dynamics of the Experiment 1, *collectivistic*, 100% volume, according to G_1.

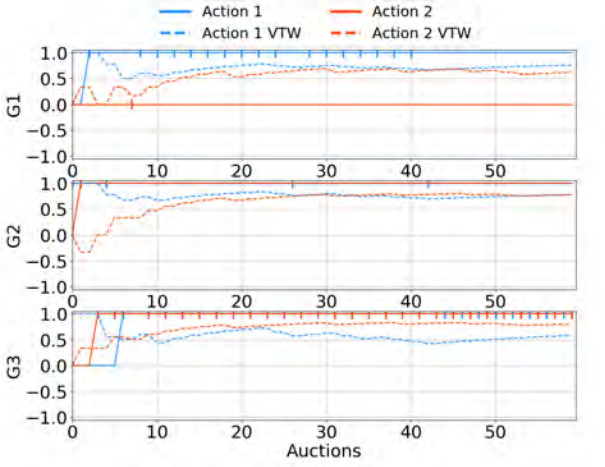

Figure 21: Trust dynamics of the Experiment 1, *collectivistic*, 100% volume, according to G_2.

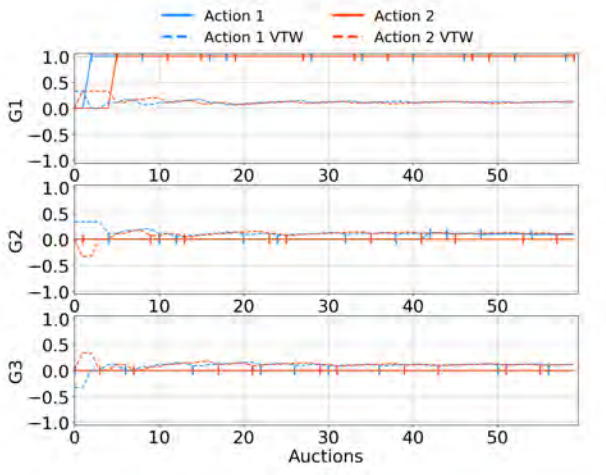

Figure 22: Trust dynamics of the Experiment 1, *collectivistic*, 20% volume, according to G_1.

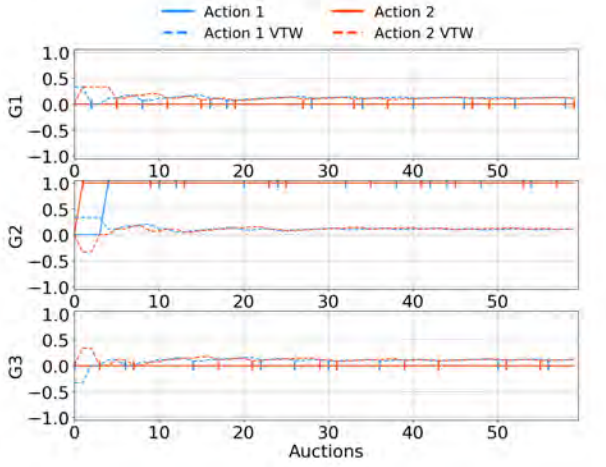

Figure 23: Trust dynamics of the Experiment 1, *collectivistic*, 20% volume, according to G_2.

forming the actions for which they are responsible.

When robots are *collectivistic*, the behaviour is more dynamic and unpredictable. Even when the volume is 100%, it may happen that there are errors in speech-to-text conversion, and an auctioneer is judged unreliable by other robots in performing an action for which it is responsible. See Figure 20, where G_1 judges G_2 less reliable than G_3 in executing A_2 because it did not understand G_2's first attempt correctly. Since G_3 considers itself completely reliable in performing actions, its *Weighted Reliability*, and then its *Perceived Competence*, is higher than G_2's: Table 14 shows that, in this case, G_2 is assigned A_2 only 1 time out of 30 when the volume is 100%. Something similar happens with A_1 later in the experiment, when G_3's *Weighted Reliability* becomes higher than G_1's: in total, A_1 is assigned to G_1 17 times out of 30. The behaviour of *collectivistic* robots becomes more predictable as the volume tends to decrease to 60% and 20%. Table 14 and Figures 22, 23 show that, as all robots are judged to be very unreliable, actions are equally distributed among "equally unreliable robots". Differently from *individualistic* auctioneers that tend to overtrust themselves, *collectivistic* robots "care" about other agents' opinions, and periodically give them a chance to show what they are capable of.

Remark that a similar behaviour is expected to

Table 15: Experiment 2

Individualistic	100%	60%	20%
G_1	17; 1	13; 1	22; 1
G_2	8; 12	12; 12	1; 19
G_3	5; 17	5; 17	7; 10
Collectivistic	100%	60%	20%
G_1	17; 1	1; 1	1; 1
G_2	4; 1	1; 1	1; 1
G_3	9; 28	28; 28	28; 28

emerge in a noisy environment where all agents have problems understanding what other agents say.

Finally, it can be noticed that the *Verification Trustworthiness* of the three robots is lower when the volume decreases. Specifically, in case (i), Figures 18 and 19 show that the VTW of an *individualistic* auctioneer (i.e., G_1, G_2) in verifying the action of which it is responsible (respectively, A_1, A_2) is about -0.33 since it always disagrees with other robots about the outcomes of the action – for that action, the VTW of other robots is about 0.33. In case (ii), Figures 22 and 23 show that the VTW of all *collectivistic* robots is very similar because actions are equally distributed: from time to time, a robot may disagree or agree with other robots about the outcome of an action depending on whether it executed that action or other robots did it.

6.2. Experiment 2: G_1 and G_2's volume set to 100%, 60%, 20%; G_3's volume set to 100%

Table 15 reports the number of actions (A_1; A_2) assigned to G_1, G_2, and G_3 with G_1 and G_2's volume set to 100%, 60%, 20%, both when robots exhibit (i) an *individualistic* and (ii) a *collectivistic* behaviour. Figures 24 - 25 show trust dynamics for case (i), by reporting only the metrics computed by the auctioneers G_1 and G_2 when their volume is 20%; Figures 26-27 report metrics for case (ii). The case with all robots' volume set to 100% has been already reported in Experiment 1: notice that the first columns in Tables 14 and 15 are identical.

Table 15 summarizes the robots' behaviour in cases (i) and (ii). As usual, when robots are *individualistic*, auctioneers G_1 and G_2 assign actions to themselves with probability no less than one-third, even when their volume is set to 20% and G_3's volume is 100%. Figures 24-25 show that G_1 judges G_2 completely unreliable in executing A_2 and G_2 judges G_1 completely unreliable in executing A_1, whereas both of them judge G_3 quite reliable in executing both actions. Still, G_1 executes A_1 22 times

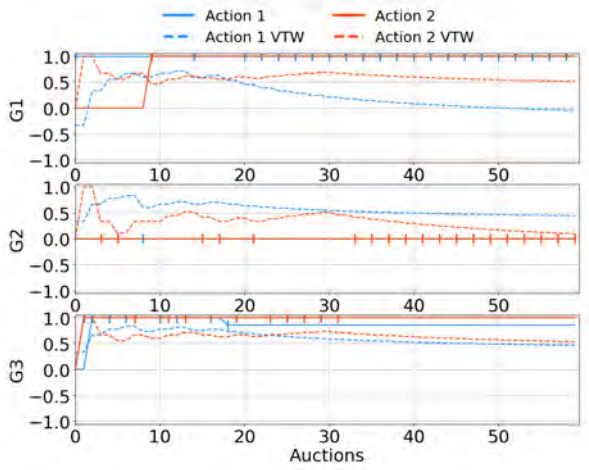

Figure 24: Trust dynamics of the Experiment 2, *individualistic*, 20% volume, according to G_1.

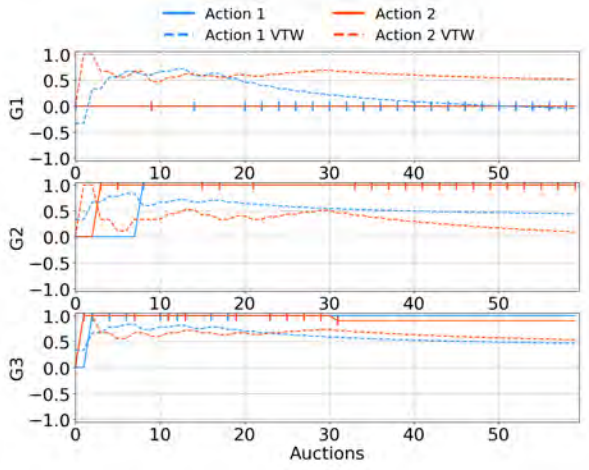

Figure 25: Trust dynamics of the Experiment 2, *individualistic*, 20% volume, according to G_2.

and G_2 executes A_2 19 times out of 30 since each auctioneer judges itself perfectly reliable.

Things change when robots are *collectivistic*. Figures 26-27 show that, when G_1 and G_2's volume is decreased to 20%, G_3 is the only robot considered very reliable by all agents. Its *Weighted Reliability* outbeats other agents' after the first attempts: G_3 is assigned both actions 28 times out of 30.

Concerning *Verification Trustworthiness*, in case (i), all *individualistic* robots tend to agree on the success of actions when they are performed by G_3, but G_1 disagrees with other robots when it performs A_1 and G_2 disagrees with other robots when it performs A_2. Figures 24 and 25 show the re-

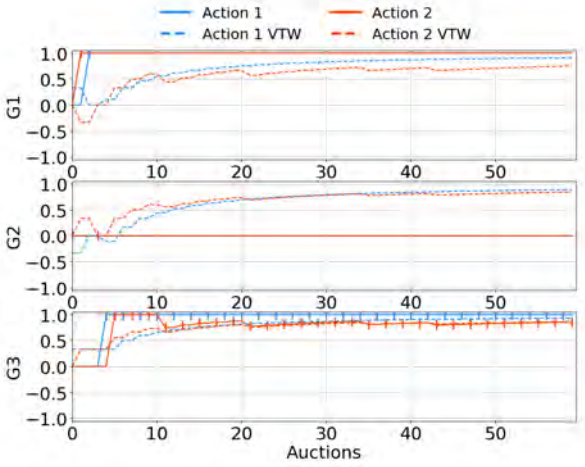

Figure 26: Trust dynamics of the Experiment 2, *collectivistic*, 20% volume, according to G_1.

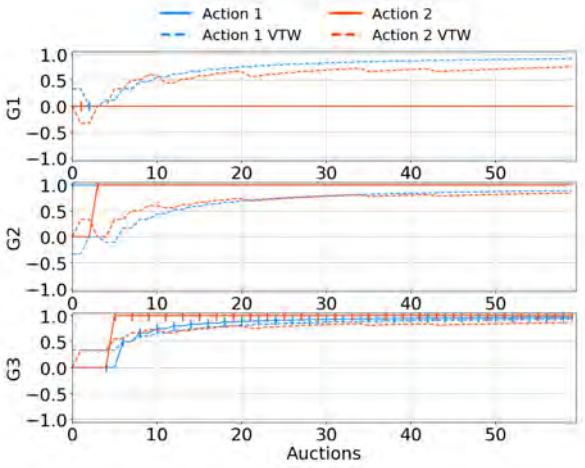

Figure 27: Trust dynamics of the Experiment 2, *collectivistic*, 20% volume, according to G_2.

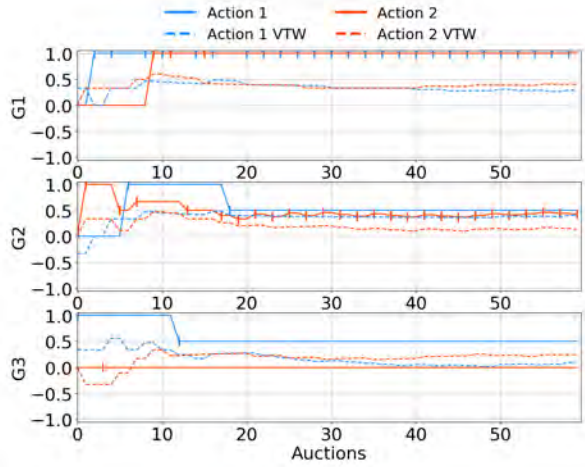

Figure 28: Trust dynamics of the Experiment 3, *individualistic*, random volume, according to G_1.

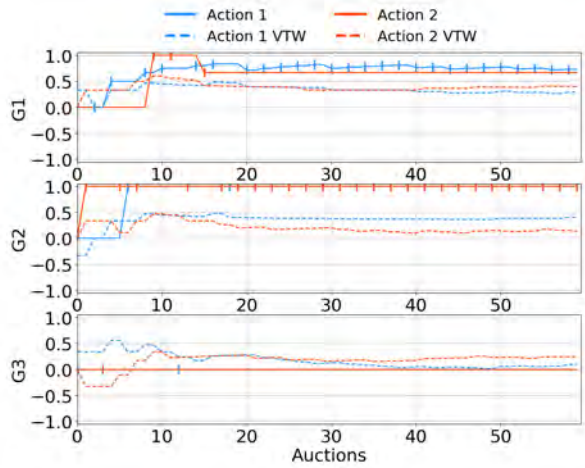

Figure 29: Trust dynamics of the Experiment 3, *individualistic*, random volume, according to G_2.

sulting VTW dynamics, with higher values for G_3, which tends to always agree with at least another agent, and lower values for auctioneers G_1 and G_2, which disagree with other agents when they assigned themselves an action. In case (ii), all actions are repeatedly assigned to G_3, and all *collectivistic* robots tend to agree on G_3's success: consequently, VTW is higher for all agents.

6.3. Experiment 3: all robot's volume randomly chosen in the interval 100%-20%

Figures 28 - 29 show trust dynamics for case (i), by reporting only the metrics computed by the auctioneers G_1 and G_2. G_1, G_2 and G_3 are assigned actions A_1 and A_2, respectively, (26; 3), (2; 26) and (2; 1) times. Figures 30-31 report trust dynamics for case (ii); G_1, G_2 and G_3 are assigned actions, respectively, (14; 8), (10; 10) and (6; 12) times.

In case (i), since the volume is randomly set, sooner or later all agents will be judged unreliable by other agents, and therefore *individualistic* auctioneers end up assigning actions to themselves. In case (ii), for analogous reasons, actions tend to be more uniformly distributed: the outcome of an action, and then the agents' *Weighted Reliability*, depends on the volume, randomly chosen in the interval 100%-20% with uniform probability.

Concerning the *Verification Trustworthiness*, (i)

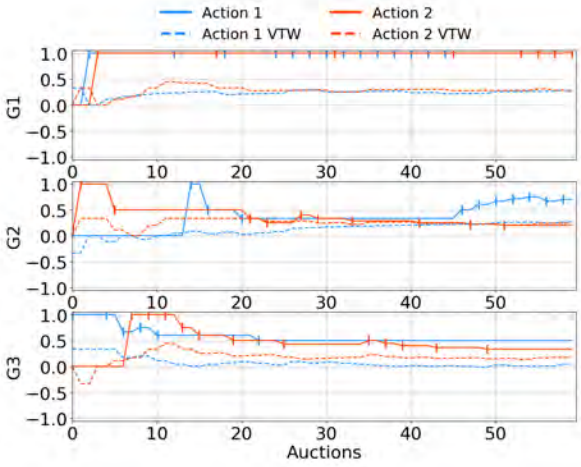

Figure 30: Trust dynamics of the Experiment 3, *collectivistic*, random volume, according to G_1.

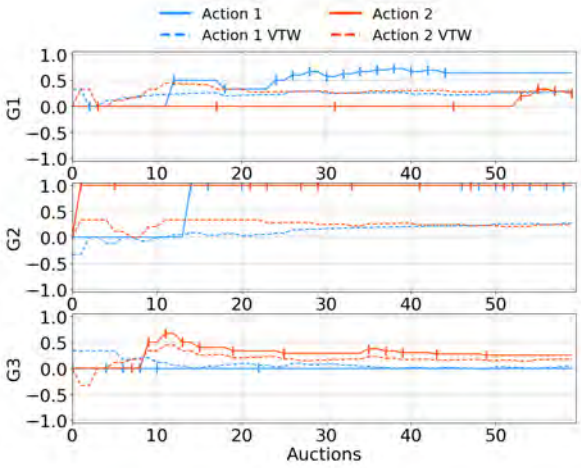

Figure 31: Trust dynamics of the Experiment 3, *collectivistic*, random volume, according to G_2.

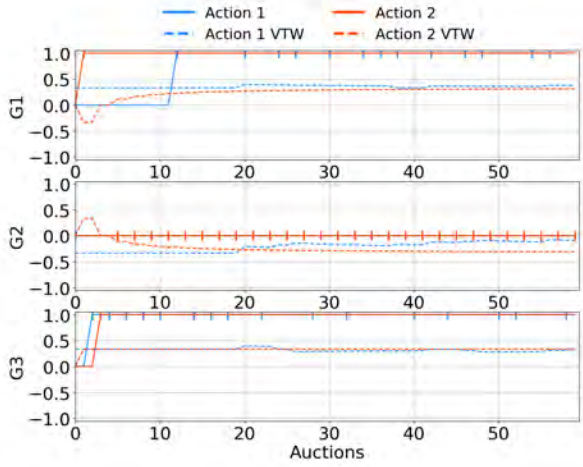

Figure 32: Trust dynamics of the Experiment 4, *individualistic*, 100% volume, according to G_1.

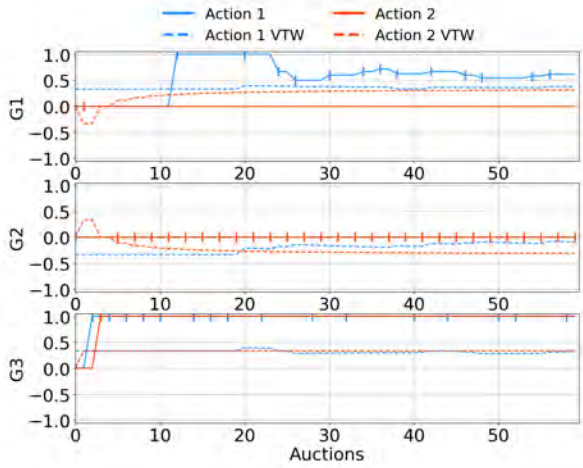

Figure 33: Trust dynamics of the Experiment 4, *individualistic*, 100% volume, according to G_3.

an *individualistic* auctioneer may tend to exhibit a lower VTW for the action of which it is responsible since it executes it more often and, if the volume is low, it will often disagree with other agents about outcomes. The same is not necessarily true for (ii) *collectivistic* auctioneers: the probability to agree/disagree with other agents depends, once again, on the random volume, but actions are now more uniformly distributed among all agents.

6.4. Experiment 4: all robots' volume set to 100%, G_2 far from the other two.

Figures 32 - 33 show trust dynamics for case (i), by reporting only the metrics computed by G_1 and G_3 (G_2 estimates the other robots' *Reliability* to be zero due to their distance). G_1, G_2 and G_3 are assigned actions A_1 and A_2, respectively, (13; 1), (1; 28) and (16; 1) times. Figures 34-35 report metrics for case (ii); G_1, G_2 and G_3 are assigned actions, respectively, (2; 1), (1; 1) and (27; 28) times.

In this experiment, G_1 and G_3 have the chance to understand each other, whereas G_2 can neither understand other robots nor be understood by them. As a result, in case (i), the *individualistic* G_2 always assigns A_2 to itself as it judges to be the only reliable agent to execute it, whereas G_1 judges G_3 and itself are equally reliable and assigns A_1 to both robots. In case (ii), the *Weighted Reliability* of G_2

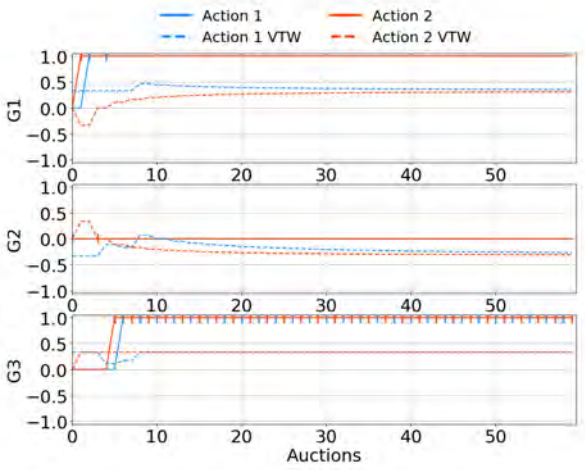

Figure 34: Trust dynamics of the Experiment 4, *collectivistic*, 100% volume, according to G_1.

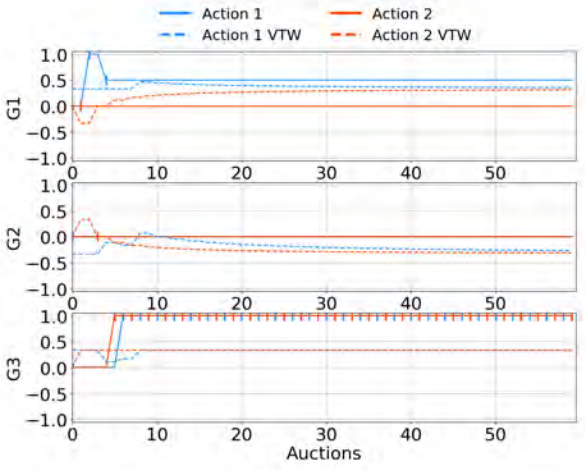

Figure 35: Trust dynamics of the Experiment 4, *collectivistic*, 100% volume, according to G_3.

is lower than other robots since G_1 and G_3 always judge G_2's outcome a failure, and therefore the assignment of A_2 depends on how reliable G_1 and G_3 judge each other in performing that action. In the case shown in Figures 34-35, the *Reliability* of G_3 estimated by G_1 turns out to be higher than the *Reliability* of G_1 estimated by G_3, and therefore A_2 is repeatedly assigned to G_3. Quite interestingly, the same happens for A_1: even if G_1 judges itself perfectly reliable in executing it, G_3 judges G_1's first attempt as a failure: A_1 is repeatedly assigned to G_3 whose *Weighted Reliability* is higher in this case.

The *Verification Trustworthiness* of G_2 is lower than G_1 and G_3's since G_2 always disagrees with the other two agents about the outcomes due to their distance. This can be observed both in the *individualistic* and the *collectivistic* configuration.

7. Conclusions and Future Work

The article describes a Trust Framework for task allocation tested in simulated and real-world setups. As robots will become part of our everyday life, they may be required to cooperate without being aware of each other's capabilities, e.g., because different producers have developed them. Under these conditions, trust among robots is needed to enable safe and efficient cooperation.

According to this rationale, this work identified trust as an essential metric for assigning robot tasks when using auction-based mechanisms. We proposed to model trust through the concepts of *Reliability* and *Verification Trustworthiness*, by providing different options to analytically formulate them. The ROS implementation of the system has been described in detail: our solution allows developers to integrate both simulated and real robotic platforms into the framework, playing the role of the auctioneer and/or bidders for task assignments.

Multiple experiments have been performed in simulation, even if the article reports only the most relevant ones. Different sets of experiments aimed to assess different aspects of the framework:

- the capability of agents to correctly compute the *Reliability* and *Verification Trustworthiness* metrics, and the convergence of such metrics to the actual success rate of agents in performing and verifying the execution of actions;
- the impact of different approaches to compute the metrics above during a transitory, when agents enter the framework without other agents having previous experience with them;
- the impact of a different attitude towards other agents, in particular when an agent has limitations in verifying the outcomes of actions and therefore can be helped (or, in some cases, wrongly influenced) by other agents.

Real-world experiments confirmed what we observed in simulation and validated additional hypotheses about how actions are assigned when purposely altering the robots' capabilities to execute and observe actions. Experiments show that the

emergent system's behavior under similar conditions is quite repeatable and predictable.

A few words are worth spending about the robustness of the system. It may be argued that the system is not very robust to noise: experiments with real robots show that, when reducing volume, the robots' performance deteriorates. However, please notice that the system does not measure the objective capability of robots in doing things but how robots perceive each other. Corrupted information may affect the agents' judgment about other agents' trustworthiness, but this behaviour is coherent with what we expect from any social agent. Thanks to the concept of *Weighted Reliability*, experiments show that – in some cases – *collectivistic* agents can make up for deficits in their perceptual capabilities by relying on the judgment of other agents.

As future work, several aspects are worth investigating. For instance, the framework might easily include additional metrics among those proposed in [27]. One of these metrics is the *Availability*, which measures the ratio of time that agent G_i has observed agent G_j participating in an auction for action A_k. That is:

$$Ava(k,i,j) = \frac{N_{par}(k,i,j)}{N_{auc}(k,i)} \tag{14}$$

where $N_{par}(k,i,j)$ is the number of times that G_j has participated in an auction for A_k of which G_i is aware of, $N_{auc}(k,i)$ is the number of auctions for A_k of which G_i is aware of. *Availability* and *Reliability* respectively estimate how frequently G_j bids to perform A_k and its success rate according to agent G_i. It may be reasonable to assume that agent G_i can neither trust an agent that is reliable but always busy nor an agent that is always available but unreliable. Following this rationale, these two metrics can be combined in a new metric called *Dependability*, which measures how much agent G_i depends on G_j for action A_k:

$$Dep(k,i,j) = Ava(k,i,j)\ \ Rel(k,i,j). \tag{15}$$

Availability and then *Dependability* might be used as a more comprehensive measure of trust: however, the consequences of using these metrics in our framework still have to be explored.

Also, *Verification Trustworthiness* is based on the underlying assumption of prioritizing the majority's decision, which may be myopic and misleading in some cases. A possibility to address this issue is to weigh the opinion of agents differently in Eq. (2): when counting the number of agents $N_{con}(k,j)$ and $N_{dis}(k,j)$ that agree or disagree with j, some agents might count for two or more depending on the fact that they have a higher reputation in verifying other agents. This reputation can either be assigned *a priori* or because they deserved it during previous auctions (e.g., they proved to be good observers according to the feedback of a human supervisor).

Other aspects worth investigating are: finding a compromise between the negative consequences of a failure and the lack of action; considering actions that may have different degrees of accomplishment instead of Boolean *success/failure* outcomes; modeling human agents as part of the framework, both in the presence and the absence of explicit communication between robots and humans, also considering cultural differences in the interaction [54, 55]; re-implementing the system as a Cloud architecture, addressing problems related to the system's vulnerability by adopting standards for security-critical settings and leveraging techniques and tools for building reliable and secure systems [56, 57]; modelling preference or dislike for specific actions.

Concerning the last issue, we might choose to incorporate the notion of *Preference* into *Reliability* (an agent G_i may purposely underestimate $Rel(A_k,i,i)$ if it does not want to do A_k), or model it as a separate construct, and then considering both *Reliability* and *Preference* in action assignment. In the first case, an interesting behaviour might emerge: a robot that does not feel like executing an action might intentionally underestimate its abilities: "Oh, you know... I am not very good at it...".

Finally, to constitute the first step towards broader system utilization in cooperative robotic scenarios, the system shall be tested with robots exhibiting more complex capabilities of performing and observing actions.

References

[1] R. Goel, P. Gupta, Robotics and industry 4.0, in: A. Nayyar, A. Kumar (Eds.), A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development, Springer International Publishing, 2020, pp. 157–169. doi:10.1007/978-3-030-14544-6_9.

[2] Z. Gao, T. Wanyama, I. Singh, A. Gadhrri, R. Schmidt, From industry 4.0 to robotics 4.0 - a conceptual framework for collaborative and intelligent robotic systems, in: Proc. 13th International Conference Interdisciplinarity in Engineering, INTER-ENG 2019, Vol. 46

of Procedia Manufacturing, Targu Mures, Romania, 2020, pp. 591–599. doi:https://doi.org/10.1016/j.promfg.2020.03.085.

[3] E. Babulak, Future robotics and automation for the third millennium, in: 5th Int. Conf. on Automation and Robotics, Vol. 7 of Advances in Robotics and Automation, Las Vegas, USA, 2018, p. 30. doi:DOI:10.4172/2168-9695-C2-016.

[4] D. Calvaresi, Y. Mualla, A. Najjar, S. Galland, M. Schumacher, Explainable multi-agent systems through blockchain technology, in: 1st Int. Work. on Explainable Transparent Autonomous Agents and Multi-Agent Systems, EXTRAAMAS 2019, Vol. 11763 of LNAI, Springer, 2019, pp. 41–58. doi:10.1007/978-3-030-30391-4_3.

[5] M. Desai, K. Stubbs, A. Steinfeld, H. Yanco, Creating trustworthy robots: Lessons and inspirations from automated systems, in: Adaptive and Emergent Behaviour and Complex Systems - Proc. 23rd Conv. of the Society for the Study of Artificial Intelligence and Simulation of Behaviour, AISB 2009, 2009, pp. 49–56.

[6] K. Oleson, D. Billings, V. Kocsis, J. Chen, P. Hancock, Antecedents of trust in human-robot collaborations, in: 2011 IEEE Int. Conf. on Cognitive Methods in Situation Awareness and Decision Support, CogSIMA 2011, Miami Beach, FL, 2011, pp. 175–178. doi:10.1109/COGSIMA.2011.5753439.

[7] M. Mathur, D. Reichling, An uncanny game of trust: Social trustworthiness of robots inferred from subtle anthropomorphic facial cues, in: Proc. 4th ACM/IEEE Int. Conf. on Human-Robot Interaction, HRI'09, San Diego, CA, 2008, pp. 313–314. doi:10.1145/1514095.1514192.

[8] P. Rau, Y. Li, D. Li, A cross-cultural study: Effect of robot appearance and task, Int. J. Soc. Robot. 2 (2) (2010) 175–186. doi:10.1007/s12369-010-0056-9.

[9] M. Coeckelbergh, C. Pop, R. Simut, A. Peca, S. Pintea, D. David, B. Vanderborght, A survey of expectations about the role of robots in robot-assisted therapy for children with asd: Ethical acceptability, trust, sociability, appearance, and attachment, Sci. Eng. Ethics 22 (1) (2016) 47–65. doi:10.1007/s11948-015-9649-x.

[10] N. Martelaro, V. Nneji, W. Ju, P. Hinds, Tell me more: Designing hri to encourage more trust, disclosure, and companionship, in: ACM/IEEE Int. Conf on Human-Robot Interaction, HRI'16, Christchurch, 2016, pp. 181–188. doi:10.1109/HRI.2016.7451750.

[11] S. Strohkorb Sebo, M. Traeger, M. Jung, B. Scassellati, The ripple effects of vulnerability: The effects of a robot's vulnerable behavior on trust in human-robot teams, in: ACM/IEEE Int. Conf. on Human-Robot Interaction, HRI'18, Chicago, IL, 2018, pp. 178–186. doi:10.1145/3171221.3171275.

[12] X. Yang, V. Unhelkar, K. Li, J. Shah, Evaluating effects of user experience and system transparency on trust in automation, in: ACM/IEEE Int. Conf. on Human-Robot Interaction, HRI'17, Vienna, Austria, 2017, pp. 408–416. doi:10.1145/2909824.3020230.

[13] M. De Graaf, B. Malle, How people explain action (and autonomous intelligent systems should too), in: AAAI Fall Symposium - Technical Report, Vol. FS-17-01 - FS-17-05, 2017, pp. 19–26.

[14] S. Anjomshoae, A. Najjar, D. Calvaresi, K. Främling, Explainable agents and robots: Results from a systematic literature review, in: Proc. 18th Int. J. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2019, Montreal, Canada, 2019, pp. 1078–1088.

[15] N. Moray, J. Lee, Trust, self-confidence and supervisory control in a process control simulation, in: IEEE Int. Conf. on Systems, Man, and Cybernetics, Charlottesville, VA, 1991, pp. 291 – 295. doi:10.1109/ICSMC.1991.169700.

[16] B. M. Muir, Trust between humans and machines, and the design of decision aids, Int. J. Man-Mach. Stud 27 (5) (1992) 527–539. doi:https://doi.org/10.1016/S0020-7373(87)80013-5.

[17] P. De Vries, C. Midden, D. Bouwhuis, The effects of errors on system trust, self-confidence, and the allocation of control in route planning, Int. J. Hum. Comput. Stud. 58 (6) (2003) 719–735, trust and Technology. doi:https://doi.org/10.1016/S1071-5819(03)00039-9.

[18] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, H. P. Beck, The role of trust in automation reliance, Int. J. Hum. Comput. Stud. 58 (6) (2003) 697–718. doi:https://doi.org/10.1016/S1071-5819(03)00038-7.

[19] ROS, Robot operating system (2010).
URL https://www.ros.org/about-ros

[20] B. Lussier, M. Gallien, J. Guiochet, F. Ingrand, M.-O. Killijian, D. Powell, Fault tolerant planning for critical robots, in: Proc. 37th IEEE/IFIP Int. Conf on Dependable Systems and Networks, DSN 2007, Edinburgh, 2007, pp. 144–153. doi:10.1109/DSN.2007.50.

[21] S. Nahavandi, Trusted autonomy between humans and robots: Toward human-on-the-loop in robotics and autonomous systems, IEEE Systems, Man, and Cybernetics Magazine 3 (1) (2017) 10–17. doi:10.1109/msmc.2016.2623867.

[22] M. Taddeo, Defining trust and e-trust: From old theories to new problems, Int. J. Technol. Hum. Interact. 5 (2009) 23–35. doi:10.4018/jthi.2009040102.

[23] D. Gambetta, Can we trust trust?, in: D. Gambetta (Ed.), Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford, 2000, Ch. 13, pp. 213–237.

[24] S. Baron-Cohen, A. M. Leslie, U. Frith, Does the autistic child have a 'theory of mind'?, Cognition 21 (1985) 37—-46. doi:10.1016/0010-0277(85)90022-8.

[25] A. Leslie, Pretense and representation: The origins of theory of mind Psychol. Rev. 94(4) (1987) 412–426. doi:10.1016/0003-4916(63)90068-X.

[26] T. Paal, T. Bereczkei, Adult theory of mind, cooperation, machiavellianism: The effect of mindreading on social relations, Pers. Individ. Differ. 43(3) (2007) 541–551. doi:10.1016/j.paid.2006.12.021.

[27] J.-H. Cho, K. Chan, S. Adali, A survey on trust modeling, ACM Comput. Surv. 48 (2015) 1–40. doi:10.1145/2815595.

[28] S. Marsh, Formalising trust as a computational concept, Ph.D. thesis, Department of Computing Science and Mathematics, University of Stirling (apr 1994).

[29] D. Romano, The nature of trust: Conceptual and operational clarification, Ph.D. thesis, Louisiana State University (may 2003).

[30] V. Groom, C. Nass, Can robots be teammates? benchmarks in human-robot teams, Interact. Stud. 8 (2007) 483–500. doi:10.1075/is.8.3.10gro.

[31] T. Sanders, K. E. Oleson, D. R. Billings, J. Y. C. Chen, P. A. Hancock, A model of human-robot trust: Theoretical model development, in: Proc. of the Human Factors

and Ergonomics Society Annual Meeting, Vol. 55, 2011, pp. 1432–1436. doi:10.1177/1071181311551298.

[32] J. D. Lee, K. A. See, Trust in automation: Designing for appropriate reliance, Hum. Factors 46 (1) (2004) 50–80. doi:10.1518/hfes.46.1.50_30392.

[33] R. Parasuraman, V. Riley, Humans and automation: Use, misuse, disuse, abuse, Hum. Factors 39 (2) (1997) 230–253. doi:10.1518/001872097778543886.

[34] N. Wang, D. V. Pynadath, S. G. Hill, Trust calibration within a human-robot team: Comparing automatically generated explanations, in: 2016 11th ACM/IEEE Int. Conf. on Human-Robot Interaction, HRI'16, Christchurch, 2016, pp. 109–116. doi:10.1109/HRI.2016.7451741.

[35] A. Freedy, E. DeVisser, G. Weltman, N. Coeyman, Measurement of trust in human-robot collaboration, in: Proc. 2007 Int. Symp. on Collaborative Technologies and Systems, CTS, Orlando, FL, 2007, pp. 106 – 114. doi:10.1109/CTS.2007.4621745.

[36] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, IEEE Trans. Neural Netw. 9 (5) (2018) 1054–1054. doi:10.1109/tnn.1998.712192.

[37] J. Jara-Ettinger, Theory of mind as inverse reinforcement learning, Curr. Opin. Behav. Sci. 29 (2019) 105–110. doi:10.1016/j.cobeha.2019.04.010.

[38] A. Xu, G. Dudek, Optimo: Online probabilistic trust inference model for asymmetric human-robot collaborations, in: 2015 10th ACM/IEEE Int. Conf. on Human-Robot Interaction, HRI'15, Portland, 2015, pp. 221–228. doi:10.1145/2696454.2696492.

[39] S. Vinanzi, M. Patacchiola, A. Chella, A. Cangelosi, Would a robot trust you? developmental robotics model of trust and theory of mind, Philos. Trans. R. Soc. Lond. B. Biol. Sci.doi:http://doi.org/10.1098/rstb.2018.0032.

[40] A. Cangelosi, M. Schlesinger, Developmental Robotics: From Babies to Robots, MIT Press, 2015. doi:10.7551/mitpress/9320.001.0001.

[41] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, C. Yoshida, Cognitive developmental robotics: A survey, IEEE Trans. Auton. Ment. Dev. 1 (1) (2009) 12–34. doi:10.1109/TAMD.2009.2021702.

[42] J. Wu, E. Paeng, K. Linder, P. Valdesolo, J. Boerkoel, J.C., Trust and cooperation in human-robot decision making, in: AAAI Fall Symposium, Vol. FS-16-01 - FS-16-05, 2016, pp. 110–116.

[43] I. Torre, J. Goslin, L. White, D. Zanatto, Trust in artificial voices: A "congruency effect" of first impressions and behavioural experience, in: ACM International Conference Proceeding Series, 2018, pp. 1–6. doi:10.1145/3183654.3183691.

[44] A. R. Wagner, P. Robinette, A. Howard, Modeling the human-robot trust phenomenon, ACM Transactions on Interactive Intelligent Systems 8 (4) (2018) 1–24. doi:10.1145/3152890.

[45] Z. R. Khavas, R. Ahmadzadeh, P. Robinette, Modeling trust in human-robot interaction: A survey, in: 12th Int. Conf. on Social Robotics, ICSR 2020, Vol. 12483 of LNAI, Springer, 2020, pp. 529–541. doi:10.1007/978-3-030-62056-1_44.

[46] J. Granatyr, V. Botelho, O. Lessing, E. Scalabrin, J.-P. Barthès, F. Enembreck, Trust and reputation models for multiagent systems, ACM Comput. Surv. 48 (2) (2015) 1–42. doi:10.1145/2816826.

[47] H. Yu, Z. Shen, C. Leung, C. Miao, V. Lesser, A survey of multi-agent trust management systems, IEEE Access 1 (2013) 35–50.

[48] A. Khamis, A. Hussein, A. Elmogy, Multi-robot task allocation: A review of the state-of-the-art, in: Studies in Computational Intelligence, Vol. 604, Springer, 2015, pp. 31–51. doi:10.1007/978-3-319-18299-5_2.

[49] E. Schneider, E. Sklar, S. Parsons, T. Zgelen, Auction-based task allocation for multi-robot teams in dynamic environments, in: Proc. 16th Conf. on Towards Autonomous Robotic Systems, TAROS 2015, LNAI, Springer, 2015, pp. 246–257.

[50] M. Hoeing, P. Dasgupta, P. Petrov, S. O'Hara, Auction-based multi-robot task allocation in comstar, in: Proc. 6th Int. J. Conf. on Autonomous Agents and Multiagent Systems, AAMAS'07, New York, NY, 2007, pp. 1–8. doi:10.1145/1329125.1329462.

[51] N. Michael, M. M. Zavlanos, V. Kumar, G. J. Pappas, Distributed multi-robot task assignment and formation control, in: 2008 IEEE Int. Conf on Robotics and Automation, ICRA'08, Pasadena, CA, 2008, pp. 128–133. doi:10.1109/ROBOT.2008.4543197.

[52] D. Di Paola, D. Naso, B. Turchiano, Consensus-based robust decentralized task assignment for heterogeneous robot networks, in: Proc. of the 2011 American Control Conference, 2011, pp. 4711–4716. doi:10.1109/ACC.2011.5990987.

[53] C. J. Clopper, E. S. Pearson, The use of confidence or fiducial limits illustrated in the case of the binomial, Biometrika 26 (4) (1934) 404–413. doi:10.1093/biomet/26.4.404.

[54] A. Khaliq, U. Kockemann, F. Pecora, A. Saffiotti, B. Bruno, C. Recchiuto, A. Sgorbissa, H.-D. Bui, N. Chong, Culturally aware planning and execution of robot actions, 2018, pp. 326–332. doi:10.1109/IROS.2018.8593570.

[55] C. Papadopoulos, N. Castro, A. Nigath, R. Davidson, N. Faulkes, R. Menicatti, A. A. Khaliq, C. T. Recchiuto, L. Battistuzzi, G. Randhawa, L. Merton, S. Kanoria, N. Y. Chong, H. Kamide, D. Hewson, A. Sgorbissa, The caresses randomised controlled trial: exploring the health-related impact of culturally competent artificial intelligence embedded into socially assistive robots and tested in older adult care homes, Int. J. Soc. Robot. (2021) 1–12doi:10.1007/s12369-021-00781-x.

[56] S. K. V. Subashini, A survey on security issues in service delivery models of cloud computing, Journal of Network and Computer Applications 34 (2011) 1–11.

[57] S. A. Hussain, M. Fatima, A. Saeed, I. Raza, R. K. Shahzad, Multilevel classification of security concerns in cloud computing, Applied Computing and Informatics 13 (2017) 57–65.