

Helpfulness as a Key Metric of Human-Robot Collaboration

Richard G. Freedman

SIFT, LLC

319 N. 1st Ave., Suite 400, Minneapolis, MN 55401
rfreedman@sift.net

Brian C. Williams

MIT CSAIL

32 Vassar St., Cambridge, MA 02139
williams@csail.mit.edu

Steven J. Levine

MIT CSAIL

32 Vassar St., Cambridge, MA 02139
sjlevine@alum.mit.edu

Shlomo Zilberstein

University of Massachusetts Amherst
140 Governors Dr., Amherst, MA 01003
shlomo@cs.umass.edu

Abstract

As robotic teammates become more common in society, people will assess the robots' roles in their interactions along many dimensions. One such dimension is effectiveness: people will ask whether their robotic partners are trustworthy and effective collaborators. This begs a crucial question: how can we quantitatively measure the helpfulness of a robotic partner for a given task at hand? This paper seeks to answer this question with regards to the interactive robot's decision making. We describe a clear, concise, and task-oriented metric applicable to many different planning and execution paradigms. The proposed helpfulness metric is fundamental to assessing the benefit that a partner has on a team for a given task. In this paper, we define helpfulness, illustrate it on concrete examples from a variety of domains, discuss its properties and ramifications for planning interactions with humans, and present preliminary results.

1 Introduction

Interactions between decentralized, autonomous agents can yield a range of positive to negative experiences. Positive interactive experiences can establish or reinforce relationships of trust, but negative ones may require explanations to justify why trust should be maintained. A common explanation is, "I was trying to *help*." The notion of *helpfulness* is that an agent is, with honest intentions, trying to play a positive role with the task at hand. However, this term is vague when considered as a justification of behavior—what motivated the choice as a helpful one, and did it actually help?

Furthermore, as robots become more tightly involved in interactions with humans, people will prefer robots that are good interaction partners. Trust and helpfulness are two criteria that contribute to evaluating the quality of interaction partners, and they are coupled. A robot that successfully helps someone can increase their trust in the robot. Likewise, a person who trusts a robot is more likely to accept that robot's attempts to help rather than avoid it.

Helping does not typically have an explicit definition like most artificial intelligence planning problems. In particular, we cannot formulate goal criteria directly unless other agents' goals are known. It is thus *dynamic as other agents' goals change*. Without a clear definition for how intelligent interactive robots can help, it is important to identify im-



Figure 1: A robot's plan is *helpful* if it decreases the effort for the team to complete a task, compared to the human performing the task alone. If the task is to prepare a blueberry pie, an action in a helpful plan may be to fetch the fork.

portant factors that contribute to helpfulness and use those factors to create a general-purpose metric.

This paper provides an initial proposal towards such a metric for quantitatively measuring a robot's helpfulness with respect to artificial intelligence decision making. Section 2 introduces the intuition behind helpfulness, followed by related work in Section 3. We then present more formal definitions of helpfulness in Section 4 and their use within planning and execution in Sections 5, 6, and 7. Preliminary experimental results are described in Section 8. We conclude with a discussion of open questions and challenges that our proposed definition introduces in Section 9.

2 Helpfulness in a Nutshell

This section serves to illustrate the key intuitions of helpfulness via a grounded example. Consider a household robot assisting a human with a cooking task in the kitchen as shown in Figure 1. Suppose the human takes some blueberries out of the refrigerator. Based on previous mornings, the robot responsively infers that their goal is to make a blueberry pie. Given that goal, what actions should the robot take to assist with it? Preheating the oven or fetching other supplies (such as the pie plate) intuitively seem helpful because

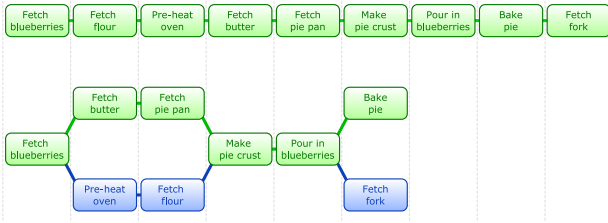


Figure 2: Top: plan for a human to make a blueberry pie alone, which takes 9 steps. Bottom: plan for a human-robot team to make a blueberry pie, taking 6 steps. Human actions are shown in green, and robot actions are shown in blue.

they bring the team closer to completing the goal.

Figure 2 illustrates two example plans that accomplish the goal. On top, we see a plan for a single agent (the human) to prepare the blueberry pie. This plan has nine steps, and only the human performs each action. On bottom, we see a plan for a human-robot team preparing the same blueberry pie. During certain steps, the robot can act in parallel with the human, thus saving time. For instance, the robot can preheat the oven at the same time that the human fetches the butter. As a result, the human-robot team takes only six steps to make the blueberry pie.

Intuitively, the robot is very helpful in this example. It selects intelligent actions that align well with the task at hand. Quantitatively, this is reflected by the comparison of the overall *cost* of executing the plan for the human-robot team versus the agent alone:

$$\begin{aligned} \text{cost}(\text{human alone}) &= 9 \\ \text{cost}(\text{human-robot team}) &= 6 \end{aligned}$$

In this example, the cost is the total number of steps until achieving the goal of being ready to eat the blueberry pie. Freedman and Zilberstein (2017) referred to the difference between these two costs—human-alone versus human-robot team—as *helpfulness* H , which quantifies the benefit gained from adding this particular robot to the team.

We further introduce *relative helpfulness*, which intuitively moves from absolute costs to a ratio of costs:

$$H_R = \frac{\text{cost}(\text{human alone}) - \text{cost}(\text{human-robot team})}{\text{cost}(\text{human alone})}$$

By dividing by the cost of the human alone, we can measure the benefit of adding the robot as a percentage. Taking the blueberry pie example, the relative helpfulness is $3/9$, indicating that adding this robot to the team decreased the cost by 33.3%. In this sense, relative helpfulness can be seen as quantifying the improvement in team efficiency.

Thus far, we have only discussed the case where cost is the number of steps to reach the goal. However, it is important to note that we can use any definition of cost, depending on the domain. Section 5 describes other possible $\text{cost}(\cdot)$ functions.

Regardless of cost metric, there is a wide range of possible helpfulness values. In the limit, imagine we have an extremely helpful robot that is capable of achieving the entire

task with no effort. For example, suppose the robot already had a blueberry pie prepared from earlier. Hence adding the robot to the team means that no actions are needed to achieve the goal. In this extreme case, $\text{cost}(\text{human-robot team}) = 0$ while $\text{cost}(\text{human alone}) = k > 0$. The relative helpfulness is therefore 1, indicating a perfectly helpful robot. In contrast, suppose we have a robot that is incapable of assisting or takes “no-operator” (no-op) actions (choosing to do nothing) at every step, or selects actions that are irrelevant to the task at hand (such as fetching spaghetti and a pot to boil water). In this case, the cost of the human performing the task alone will be equal to the cost of the team, resulting in a helpfulness value of 0. Thus, for typical collaborative interactions between the human and the robot, relative helpfulness values lie in the range of $[0, 1]$.

Finally, we wish to highlight one key property of helpfulness: it is a task-oriented metric. Depending on the team’s goals, actions may or may not contribute to a helpful execution. In the case of making a blueberry pie, a robot that begins boiling water would not be very helpful. However, if the goal is to make spaghetti, it would be a very helpful action. Thus, it is important to keep in mind that helpfulness is a function of the team’s goals.

Special Cases of Helpfulness

It is possible for helpfulness and each of its variations to take on atypical values. First, helpfulness can be negative. This usually occurs when a robot has an adversarial role in the interaction, or it can occur when a well-meaning robot happens to be accident-prone. In the case of the kitchen scenario, a clumsy robot that consistently drops items—flour, glass, pie plates, etc.—may actually cause the team to incur a higher cost compared to the human performing the task alone. Then $\text{cost}(\text{human-alone}) < \text{cost}(\text{human-robot team})$, which evaluates to $H < 0$.

Second, helpfulness can reach the extremes of infinity. In automated planning, a cost of ∞ implies that there is no solution—the effort to solve the problem is never-ending. When $\text{cost}(\text{human-alone})$ is a finite number and $\text{cost}(\text{human-robot team}) = \infty$, then the interacting robot makes reaching the goal impossible (e.g., an adversarial kitchen robot that cuts the power cord to the oven) and the helpfulness is $H = -\infty$. When $\text{cost}(\text{human-alone}) = \infty$ and $\text{cost}(\text{human-robot team})$ is a finite number, then the interacting robot is the only reason that the human can achieve the goal (e.g., a human who is less adept at baking might always burn the blueberry pie while the assistive kitchen robot never burns it) and the helpfulness is $H = \infty$. Figure 3 illustrates cases for each extreme.

3 Related Work

The general notion of evaluating interactions between humans and robots has been an ongoing research topic. The majority of the metrics for task-oriented interactions focus on various performance aspects with respect to the human, robot, or entire team (Steinfeld et al. 2006). For measuring robot performance within an environment where they may interact with humans, there is typically a greater focus on autonomy. A robot that properly assesses when and

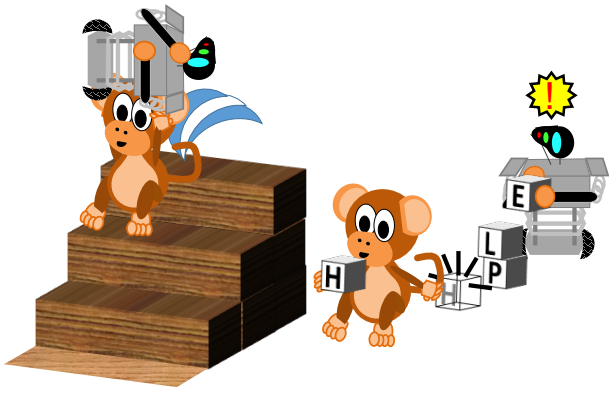


Figure 3: (Left) The monkey’s helpfulness evaluates to ∞ because the robot cannot go down the stairs by itself. (Right) The monkey’s helpfulness evaluates to $-\infty$ because the block they took is required for the robot to achieve its goal of stacking blocks to spell “HELP”.

how to interact with people will enable them to do their job with fewer interruptions and waste less time overseeing the robots. Six such metrics include task effectiveness, neglect tolerance (the robot’s performance over time without human involvement), robot attention demand, free time, fan-out (how well a human can work with multiple robots at once), and interaction effort (amount of human involvement during interaction) (Goodrich and Olsen Jr. 2003). Cognitive metrics for team collaboration, such as the similarity of team members’ mental models, have also been considered (Pina et al. 2008).

In addition to evaluating performance as a post-interaction assessment, Singer and Akin (2011) consider a variety of performance metrics for constructing effective human-robot teams. Their survey of existing metrics spans beyond the robot’s autonomy to further include system-wide evaluations for the human-robot team as a whole.

One metric often used to measure the performance of a team is the overall makespan, or the *time the team takes to complete the task* (Crandall and Goodrich 2002; Shah 2011). With increasing helpful involvement from the robot, this time should decrease since the robot will do more work in parallel with the human. Another related metric is that of *human idle time* (Shah 2011). Intuitively, teams are more efficient when members are less idle; the team members make progress towards their goal by taking action. Sometimes, one teammate is “blocked” by another teammate — i.e., one teammate’s action has preconditions that another teammate’s action achieves. For example, the human may not be able to make the pie crust until the robot fetches a required ingredient such as flour. If one agent is forced to wait for another teammate, the teamwork is less effective. Hence, a helpful robot may choose to unblock the actions earlier (e.g., the robot may fetch the flour earlier in the plan if possible) so as to improve the amount of parallelism of the team and their overall efficiency at completing the task.

Focusing on the robot within a team, it is sometimes useful to measure the *execution latency* of a robot as it

makes decisions. Intuitively, this is the time the robot requires to perform its computations and act after observing the human’s actions. Lower execution latency is preferred because the interaction feels more fluid when the robot acts quickly instead of with pauses. Execution latency was previously evaluated in the contexts of temporal plan executives (Conrad and Williams 2011) as well as human-robot interaction (Shah, Conrad, and Williams 2009; Levine and Williams 2018; Levine 2019).

Another metric identified to correlate with team performance is the *amount and type of communication* between team members. A robot that requires constant, explicit verbal communication (i.e., being commanded what to do) is likely to be less helpful than a robot that infers intent through implicit communication (i.e., observing non-verbal human cues). This may be related to the *switching cost*, where the human must pause the current task in order to assist the robot with explicitly-communicated guidance (Shah 2011).

Finally, qualitative, holistic metrics can be very important metrics for assessing the quality of human-robot interaction. Assessing the degree to which humans agree or disagree with statements such as “I trust the robot” on a Likert scale have successfully been used in prior work (Shah 2011). With respect to humans’ perspective on the potential role of robots on cooperative teams, Gombolay et al. (2014) observed through both human-robot assembly experiments as well as surveys that people are fine giving up their autonomy on scheduling and role allocation for tasks if it leads to greater task efficiency. In a similar study, Baraglia et al. (2017) found similar preferences when a robot decided whether to help people with their tasks mid-execution. Furthermore, they found that people preferred to have some control over telling the robot when to help as well as having the robot be more proactive to offer assistance.

The relationship between being helpful and improving efficiency has also been identified within psychology. An act as simple as holding the door for others not only reduces the effort they need to enter/exit a building, but those individuals also show signs of hurrying their pace to reach the door (Santamaria and Rosenbaum 2011). This responsive behavior implies intent to reduce the time that the person needs to hold the door open, reciprocating the improvement of efficiency. Independently of this research, the notion of helpfulness was quantified as the reduction in an agent’s effort to perform a task when another agent participates (Freedman and Zilberstein 2017; Freedman 2020). The work in this paper builds upon this quantification to consider variations for additional informative metrics and specific applications.

Although the former work evaluated the helpfulness of an agent that integrated automated planning with plan and intent recognition, the scope of this paper will not specifically cover the impact of recognition algorithms on helpfulness in human-robot interaction. There is certainly a close relationship however, as understanding what others are doing and why are essential to knowing who needs help and in what ways. Previous approaches to recognizing people’s intents in interactions include the theory of mind (Lemaignan et al. 2017), parsing grammars that capture task hierarchies (Geib et al. 2016), and epistemic reasoning (Shvo et al. 2020).

4 Helpfulness Defined

A common output of decision making algorithms is a *plan* π , which is a sequence of actions. Ideally, π is a solution to a problem such that an agent can, from a specified initial state, perform the actions in order and reach a state that satisfies the problem’s goal conditions G . If the solution is optimal, then no other plan can solve the problem with lower cost; we call the optimal plan π^* .

We will refer to the human and robot agents as A_H and A_R respectively. We assume that A_H has some internal goal towards which they plan and act, and A_R is the interactive partner who intends to help A_H (regardless of whether A_R knows A_H ’s goals ahead of time or infers them through observation). Collaboration prior to the interaction is ideal to facilitate explaining intentions and establishing trust between the agents, but these preparations may not always be possible. We consider the impacts of their (de)centralized plans when acting together.

In this case, π_{A_H} and π_{A_R} are the agents’ independent plans where they act on their own—if A_H ’s plan is unknown, then we assume the agent is optimal and use $\pi_{A_H}^*$ for a lower bound helpfulness estimate. As the agents act in the world and observe each other’s actions, then they can adjust their plan and predict how other agents will interact. These responsive plans (Freedman and Zilberstein 2017) have the form $\pi_{A_R \rightarrow A_H}$ such that A_R is performing the plan with respect to A_H ’s interaction. As the agents continue to interact, they may recursively shape their plans to have forms such as $\pi_{A_i \in \{R,H\} \rightarrow \dots \rightarrow A_R \rightarrow A_H}$. Alternatively, agents can have a centralized joint plan $\pi_{A_R+A_H}$ that dictates how they will work together, which *assumes a shared and known goal*. We assume $\text{cost}(\pi) = \infty$ if π does not exist.

Definition 1 *The helpfulness of agent A_R in a joint plan $\pi_{A_R+A_H}$ is the change in cost from agent A_H acting on their own to both agents working simultaneously*

$$H(A_R, \pi_{A_R+A_H}) = \text{cost}(\pi_{A_H}) - \text{cost}(\pi_{A_R+A_H}).$$

Definition 2 *The helpfulness of agent A_R in a responsive plan $\pi_{A_R \rightarrow A_H}$ is the change in cost from agent A_H acting on their own to both agents working simultaneously*

$$H(A_R, \pi_{A_R \rightarrow A_H}) = \text{cost}(\pi_{A_H}) - \text{cost}(\pi_{A_H \rightarrow A_R \rightarrow A_H}).$$

Although these helpfulness metrics evaluate an agent on a plan-by-plan basis, it is difficult to use this for comparison purposes. Assessing an agent’s overall helpfulness between multiple plans and problems, and furthermore comparing agents who have different planning paradigms, poses a risk of unfair bias when the measured plans all have different typical costs. Specifically, there are far fewer opportunities to be helpful when responding to a plan of cost c compared to responding to a plan of cost $k \cdot c$ for some $k \gg 1$.

The difference in cost does not reflect *the potential to be helpful*, which matters in situations where A_H can decide whether to perform the task and they are aware of A_R ’s intentions to interact—in this case, helpfulness can serve like value of information. As a grounded example, consider the two tasks where

- A_H must move a large piece of furniture across a room (Mörtl et al. 2012) and
- A_H must move all furniture in a room onto a truck.

The first task is inherently lower-cost so that A_H can consider moving the piece of furniture alone, and any help A_R provides is a nice bonus. The second task is much more costly to perform and A_H will likely want to ensure that A_R will provide sufficient help *relative to the task’s effort* before starting to execute a plan. This can extend to waiting for A_R to be ready to interact in a more helpful manner (Zilberstein 2015) or finding another interactive partner who is available to provide sufficient help (Rosenthal and Veloso 2012).

We thus propose normalizing helpfulness to account for how much help A_R could provide in the first place:

Definition 3 *The normalized helpfulness of agent A_R in a plan π is the ratio comparing the helpfulness $H(A_R, \pi)$ to the best-case helpfulness where A_H and A_R worked together simultaneously from the start:*

$$H_N(A_R, \pi) = \frac{H(A_R, \pi)}{\text{cost}(\pi_{A_H}) - \text{cost}(\pi_{A_H+A_R}^*)}.$$

The denominator used in normalized helpfulness represents the best-case helpfulness that could be achieved for the team, given the task and domain constraints.

Another useful variation of helpfulness is that of *relative helpfulness*, which intuitively calculates the decrease in cost for the team to perform the task relative to the cost of the human performing the task alone (Levine 2019).

Definition 4 *The relative helpfulness of an agent A_R in a plan π is the ratio comparing the helpfulness to the best-case of the human A_H performing the task by themselves:*

$$H_R(A_R, \pi) = \frac{\text{cost}(\pi_{A_H}^*) - \text{cost}(\pi_{A_H+A_H})}{\text{cost}(\pi_{A_H}^*)}$$

5 The Cost in Helpfulness

The previous definitions of helpfulness made extensive use of the $\text{cost}(\cdot)$ function. In this section, we describe some example costs in greater detail and discuss some ramifications of different cost functions.

In the blueberry pie example, we discussed the case where $\text{cost}(\cdot)$ is the number of steps to reach the goal. Counting the number of steps is the same as assuming that all actions have uniform cost. Traditionally, a plan’s cost is the sum of the costs of all the actions, no matter which agent performs them. This is because *automated planning is primarily concerned with problem solving more than agent contribution*, which means all optimal plans are equally ideal even if each agent’s efforts are disproportional.

The notion of helpfulness is applicable to many different extensions of automated planning and execution. In temporal planning and execution problems, the *makespan* (i.e., time in seconds to completion) is a valuable cost metric (Benton, Coles, and Coles 2012; Wang and Williams 2015). Other domains may measure the cost as the amount of “effort” that the agents are required to exert. In multi-agent package delivery problems, there may be a human

driver picking up and delivering packages as well as a semi-autonomous vehicle capable of most driving tasks. In such a case, the cost of a plan can be measured as the amount of mileage that the human drives. A helpful semi-autonomous vehicle would therefore optimize the pickup and delivery routes to reduce the human’s contributions to driving.

While it is possible to exclusively consider the costs of A_H ’s actions when computing $cost(\cdot)$ for plans, such definitions introduce certain issues. If A_H and A_R are both capable of solving the problem on their own, then normalized helpfulness is only maximized when A_R does *all the work*. If A_R is ever assigned a *utility function to maximize their helpfulness*, then this effectively encourages the robot to act like a lackey and also encourages the human to be as lazy as possible and take advantage of their robot partner(s). One possible solution is to ensure that doing nothing, such as performing a no-op action, has a non-0 cost. Then A_H ’s portion of the plan has some cost even if they do nothing, and A_R is most helpful when all agents split the workload in tandem. However, it is reasonable to question whether no-op should have a cost for some domain and $cost(\cdot)$ function—an agent is *unlikely to expend resources such as energy* when doing nothing, but doing nothing for the duration that the other agent acts is *spending resources such as time*.

It is not necessarily the case that resources alone define the cost function. Humans have a variety of biases, emotions, preferences, and other features that will skew their perspective of performing various actions. If a robot can reasonably model its human partner(s) to account for these aspects, then the robot can also use the information to consider more human-centric costs for being helpful. For example, if the human driver delivering packages prefers to drive on the freeway and dislikes driving through residential areas, then the robot can scale the action costs to enforce $cost(\text{freeway}) < cost(\text{no-op}) < cost(\text{residential})$. It might be more helpful if the robot did not perform some tasks that would require the human to increase their cognitive load managing the robot; likewise, it would be more helpful if a robot performed tasks in which the human desires help. Regardless of the potential loss in resource savings, *acting in reasonable ways that satisfy the human can be perceived as more helpful*.

6 Helpfulness and Uncertainty

Thus far, we have described helpfulness in a deterministic setting. In this section, we generalize helpfulness to stochastic settings and probabilistic planning.

In many settings where a robot interacts with a human, the robot has uncertainty regarding the human’s goals or plan to achieve those goals. For example: given a robot’s observation that the human has obtained blueberries from the refrigerator, will the human be making a blueberry pie, a smoothie with other fruits, or something else? This uncertainty makes it difficult for a robot to estimate its helpfulness in real-time during interaction because the precise joint plan $\pi_{A_R+A_H}$ is uncertain. To cope with this, a robot might maintain a probability distribution over the human’s goals and/or actions (Ramírez and Geffner 2010), and thereby maintain a probability distribution over possible team plans for the task.

Given the uncertainty over team plans, helpfulness becomes a random variable. We may hence compute certain statistics, such as its expected value.

Definition 5 *Given the set of all possible human-robot team plans Π , the probability of each $\pi \in \Pi$ (denoted $\Pr(\pi)$), and the robot’s set of observations so far \mathcal{O} , the expected helpfulness is the expected value of relative helpfulness over the plans given those observations:*

$$\mathbb{E}[H_R \mid \mathcal{O}] = \sum_{\pi \in \Pi} H_R(A_R, \pi) \cdot \Pr(\pi \mid \mathcal{O})$$

Note that other variations of helpfulness can be used in the above (such as, for example, normalized helpfulness).

Expected helpfulness is useful for a robot to estimate how helpful it will be in the midst of executing a plan, given its uncertainty about the world. Another valuable metric is the standard deviation of helpfulness (which can be defined similarly to Definition 5 above), a measure of the range of different helpfulness values that could result at the end of execution. As execution proceeds and the human-robot interaction unfolds, the responsive robot observes more of the human’s actions \mathcal{O}' , thereby updating the a posteriori probability distribution $\Pr(\pi \mid \mathcal{O} \cup \mathcal{O}')$.

One risk-aware executive for controlling a robot is RIKER (Levine 2019). RIKER explicitly maintains a probability distribution over the human’s choices (such as actions or goals) in contingent, temporally-flexible team plans. RIKER chooses the robot’s actions in a risk-bounded manner: it aims to ensure that the probability that the team will fail their task is at most Δ , where $0 \leq \Delta \leq 1$. By tuning Δ , qualitatively different behaviors result as the robot becomes more or less risk-averse. Low values of Δ make the robot more risk-averse (less likely to take risk) whereas higher values cause the robot to boldly take greater risks.

As the experimental results in both Section 8 and the spelling Blocksworld domain in Levine (2019) illustrate, there is a fundamental trade-off between risk and expected helpfulness. A risk-averse robot (such as when Δ is low) will be less willing to take any actions that may jeopardize execution. As such, a risk-averse robot’s safest action is often to do nothing unless it is positive it can help. This results in team costs being higher, sometimes even approaching the cost of the human acting alone. Hence helpfulness values will be lower. As the risk bound is loosened and Δ increases, the robot will tolerate more risk and act sooner, even if it is not completely certain of the human’s intentions. In this case, helpfulness values tend to be greater for cases when the team succeeds, but lower in the cases where the robot incorrectly assumes the goal and selects incorrect actions that often require backtracking.

7 Helpfulness During Planning

Rather than passively measure helpfulness as a post-hoc analysis, there are also potential uses to measuring it *actively during interaction*. Specifically, when A_R intends to independently define their own goals with respect to A_H ’s, increasing helpfulness is ideal for assistive interaction, and decreasing helpfulness is ideal for adversarial interaction.

In the case of collaborative interaction where A_R and A_H share a common goal throughout the experience, it is ideal to increase helpfulness as an assistive behavior for teammates’ subtasks and maintain non-decreasing helpfulness when working on independent subtasks. These motivate the use of helpfulness to *guide A_R ’s decision making process* throughout their interaction with A_H .

Typically, the type of interaction exclusively plays a role in A_R ’s goal generation (Freedman and Zilberstein 2017), and the planner then finds any solution to this goal from the initial state. Although this sequence of actions satisfies the goal upon completion, there is no guarantee that *the interaction type is acknowledged throughout the responsive plan’s execution*. Research in legible planning (Dragan and Srivastava 2014; Kulkarni, Srivastava, and Kambhampati 2019; Miura and Zilberstein 2020) shows that optimal-cost plans to the goal are often less interpretable for communicating one’s intents, and additional criteria need to influence the planner to account for a higher-cost, but more legible plan. In most cases, these criteria adjust the heuristic or constraints to reshape the search progression because *the path to the goal matters as much as the goal itself*.

In place of legibility, it is ideal for the robot’s plan $\pi_{A_R \rightarrow A_H}$ to be assistive, independent, or adversarial throughout. A planner’s search state should not be considered ‘near a goal state’ simply because its cost is lower, but also because it is a state that exhibits some degree of helpfulness with respect to the interaction type. In assistive interactions, helpfulness can serve as a tie-breaking strategy between states on the frontier with the same lowest expected cost to the goal—the state whose current plan has greatest helpfulness should be considered first. For independent interactions, helpfulness might not matter as much as long as (1) A_R does not interfere with A_H ’s (possibly inferred) goals and (2) A_R does not go too far out of the way from achieving their own goals. To ensure that A_R ’s plan is minimally invasive, helpfulness can serve as a tie-breaking strategy by selecting the state whose current plan’s helpfulness has the least absolute value. As the final extreme, adversarial interactions should prioritize tie-breaking towards states whose current plan has the least helpfulness. To ensure that helpfulness is not ignored when tie-breaking is unnecessary during search, we can adjust the heuristic to account for both expected cost and helpfulness as either a linear combination (Hansen and Zhou 2007) or a lexicographic preference for multiple objectives (Wray and Zilberstein 2015; Yu, Fang, and Williams 2015).

One caveat to computing helpfulness during planning, as opposed to afterwards, is the need to *approximate helpfulness* given the plan up to a search state. As a post-interaction computation, helpfulness is easy to compute because the (possibly inferred) goal is known when searching for A_H ’s single-agent plan *and* we know what actions A_H took to accomplish their goal. In a centralized multi-agent setting, the planner should know the shared goal G and at least have access to a heuristic that involves all agents present h_{joint} in addition to individual heuristics for each agent h_A . Such

planners can estimate helpfulness from any search state s :

$$h_{help}^+ = h_{A_H}(s, G) - h_{joint}(s, G). \quad (1)$$

If the planner uses a heuristic search technique such as A* and prefers to only sort candidate states by potential helpfulness, then we can extend this to compute a priority value for the frontier using initial state I and path-so-far cost functions g_{joint} and g_A . (which can be as expensive to compute as a new search problem for A_H finding a path from I to s):

$$\begin{aligned} f_{help}^+ &= g_{help} + h_{help}^+ \\ g_{help} &= g_{A_H}(I, s) - g_{joint}(I, s) \end{aligned}$$

On the other hand, a decentralized agent A_R is not guaranteed to know A_H ’s true goal or future actions; the planner must approximate both of these in the search. The computational efforts to predict A_H ’s remaining actions could be high unless we make trivializing assumptions such as A_R no longer acting, which would reduce predicting the cost of A_H ’s remaining actions to the traditional single-agent cost heuristic h_{A_H} from s and effectively set helpfulness to 0. As this is not ideal and counteracts the entire purpose of heuristic search, we instead replace G in Equation 1 with some inferred goal conditions \hat{G} and apply scalar $\alpha \in [0, 1]$ for the proportion of the joint plan that we expect A_H to perform:

$$h_{help}^{\rightarrow} = h_{A_H}(s, \hat{G}) - \alpha \cdot h_{joint}(s, \hat{G}). \quad (2)$$

If the planner uses heuristic search and prefers to only prioritize states by their helpfulness, then the above equations almost identically apply. g_{help} does not change because the path-so-far from I to s is already complete. f_{help}^{\rightarrow} simply replaces h_{help}^+ from f_{help}^+ with h_{help}^{\rightarrow} .

8 Preliminary Results

Due to the current safety concerns of performing human subjects experiments during a global pandemic, we investigate the use of helpfulness and its proposed variations with simulated AI-controlled agents. The human agent A_H uses simple heuristic search algorithms and the robot agent A_R employs either the RIKER (Levine 2019) executive or a responsive-planning implementation of the PRETCIL framework (Freedman 2020).

Domain: Foodworld

To follow our running example of preparing a blueberry pie, we use a food-assembly-themed domain that is effectively the traditional Blocksworld domain. Rather than stacking blocks with inscribed letters to spell words (when read from top-to-bottom), the *Foodworld* domain stacks food items and utensils to create dishes. For example, making the blueberry pie requires the following stack from bottom-to-top: pan, butter, dough, blueberries, and sugar.

Each agent may hold up to one item at a time via pick-and-place actions, and they can only pick up items on the top of each stack—likewise, placing an item puts it on top of the chosen stack. The action set for the human alone A_H is picking up an item, placing an item, or doing nothing; all

actions have uniform cost. The action set for the centralized team $A_R + A_H$ extends this for parallel execution where both agents may act simultaneously; each human-robot pair of actions has uniform cost. By this design, our cost function is defined relative to step time.

For the experiments that take place in the Foodworld domain, we had twelve food items and utensils and up to six possible dishes. Dishes required as few as three and as many as five of the twelve items available. To illustrate the impact of the initial state on solving a problem, we created two different initial states: the organized kitchen has almost all items alone in a single stack (easy access to most items), and the cluttered kitchen condenses all the items into six vertical stacks containing up to three items (some items require more effort to access).

Helpfulness Computations

Table 1 lists the costs of optimal plans (we simply ran A* Search) for single- and joint-agent scenarios in our Foodworld problem instances (two initial states \times six goals). In general, the cost to prepare a dish increases as more items are required, and the cost is also greater when preparing the dish in the cluttered kitchen.

The helpfulness is positive, which tells us that the robot always reduced the time taken to prepare a dish. For comparison, a vacuum robot without any limbs cannot perform these pick-and-place actions so that $cost(\pi_{A_H}^*) = cost(\pi_{A_R+A_H}^*)$ and $H = 0$. However, these integer measurements lack standardization as the optimal plan costs vary greatly between problem instances. The normalized helpfulness provides context to how these helpfulness values compare between problem instances—because all these plans are optimal, we know that the robot has reduced the plans’ costs *as much as possible* for a consistent value of 1. On the other hand, the relative helpfulness portrays the portion of the human’s overall effort that was relieved. Unsurprisingly, the robot’s relative helpfulness was greater in the cluttered kitchen per dish; the lack of organization provides more opportunities to unstack items so that the human can access them more easily without rummaging through the stacks.

These interpretations empirically hold for decentralized agents as well. Figure 4 plots the remaining plan costs as two agents interact in the organized kitchen to make a blueberry pie. The human agent simply runs A* search in the current state, but the robot employs an implementation of the PRETCIL framework that integrates probabilistic recognition as planning (Ramírez and Geffner 2010) with heuristic search. The PRETCIL framework is an architecture for closed-loop interaction that observes other agents, infers those agents’ goals, generates its own intermediate goals based on these inferences, plans for the intermediate goals, and continues to monitor the interaction for more observations while executing the plan. In this case, the robot observes without acting for the first two time steps so that the human does all the work. After two time steps, the robot identifies that the human is making a blueberry pie and assists with the task. Despite the lost time, both agents can finish the blueberry pie together in four time steps so that it still takes six steps overall; this was the optimal cost for the joint plan. Therefore, the

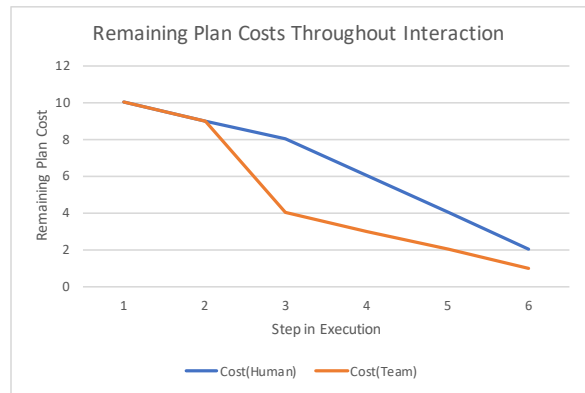


Figure 4: Plan costs to make a blueberry pie, starting from each state throughout the interaction between A_H and A_R .

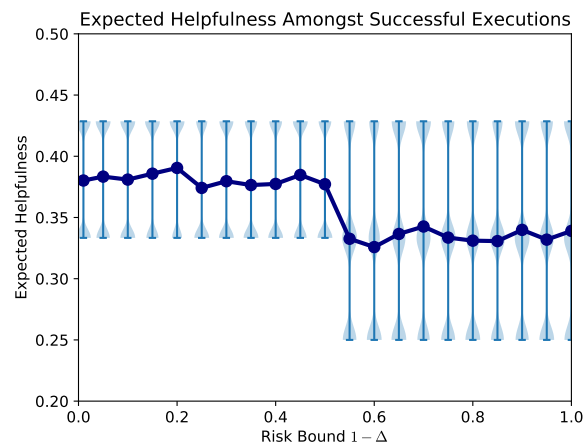


Figure 5: Expected helpfulness $\mathbb{E}[H_R | \mathcal{O}]$ as a function of RIKER’s risk bound Δ . As RIKER becomes more conservative (towards right), its expected helpfulness decreases.

helpfulness changes from zero to four after two time steps, and the normalized and relative helpfulness instantly jump from 0 to 1 and from 0 to 0.4 respectively.

Expected Helpfulness Results

In this section, we illustrate the trade-off between expected helpfulness and risk using RIKER. We encoded a breakfast-making scenario between a human and a robot as a contingent, temporally-flexible plan (i.e., as a probabilistic temporal plan network (Santana and Williams 2014)). In this plan, the team prepares either oatmeal, cold cereal with milk, or waffles. For each breakfast, the contingent plan encodes two methods of achieving the goals: one in which the robot acts immediately, and the other in which the robot begins acting after a short delay (and is able to observe some of the human’s actions first and hence infer intent). We also encoded a PDDL model (Fox and Long 2003) so that RIKER could derive causal links in order to find relations between choices in the contingent plan. We lastly encoded a Bayesian

Table 1: Helpfulness Evaluations in Foodworld (Joint Plan)

Dish (# Items in Stack)	Organized Kitchen					Cluttered Kitchen				
	Cost ($\pi_{A_H}^*$)	Cost ($\pi_{A_R+A_H}^*$)	H	H_N	H_R	Cost ($\pi_{A_H}^*$)	Cost ($\pi_{A_R+A_H}^*$)	H	H_N	H_R
Sugar Cookie (3)	6	4	2	1	0.3	10	6	4	1	0.4
Blueberry Pie (5)	10	6	4	1	0.4	16	8	8	1	0.5
Fudge (4)	8	5	3	1	0.375	12	6	6	1	0.5
Jelly Donut (3)	6	4	2	1	0.3	14	8	6	1	0.429
Choco-Chip Cookie (3)	6	4	2	1	0.3	12	6	6	1	0.5
Cake (5)	12	6	6	1	0.5	12	6	6	1	0.5

network to describe the probability distribution over the human’s intentions in this plan, with all breakfasts being equally likely. This model is analogous to the collaborative spelling blocksworld model that Levine (2019) describes.

Execution of this breakfast-making scenario was simulated thousands of times with RIKER to show the relationship between expected helpfulness and aversion to risk. During these simulations, RIKER’s risk tolerance Δ was varied. For each value of Δ , execution was simulated 50 times and the corresponding helpfulness of the resulting execution was measured. The expected helpfulness $\mathbb{E}[H_R \mid \mathcal{O}]$ was then computed by taking the mean helpfulness (hence, this was a Monte Carlo simulation). In total, 2100 simulations were performed.

The results are visualized in Figure 5. RIKER’s risk bound Δ is plotted along the x -axis, and the expected helpfulness $\mathbb{E}[H_R \mid \mathcal{O}]$ is shown on the y -axis. Light blue bands illustrate the relative distribution of discrete helpfulness values, and the dark blue curve illustrates $\mathbb{E}[H_R \mid \mathcal{O}]$. Our key observation is that, as RIKER becomes more cautious and more risk-averse (towards the right), we see that the expected helpfulness decreases (this is the drop that occurs near $\Delta = 0.5$). While the expected helpfulness was greater for cases where RIKER took more risk, execution was also more likely to fail in these cases (for instance, by the robot guessing the wrong breakfast and taking action prematurely). The simulations described here assume that execution ends when such failures occur, and calculate $\mathbb{E}[H_R \mid \mathcal{O}]$ up to that point. Alternatively, it is possible to assume that execution continues after a failure – which could possibly result in a lower helpfulness (but no failure) for those runs.

9 Conclusions and Future Work

The abstract nature of *helping* is generally evaluated more holistically compared to the corresponding tasks that have concrete goal conditions. It is often easy to be successful with respect to the interaction’s description, but the human could deem the robot’s *level of success* lackluster. Poor evaluations could impact the human’s trust. For example:

- A robot carries *only* a single bag of pretzels to the kitchen to help, but the human carries the remaining groceries including heavy milk cartons and canned food.
- A robot treads around the carpet’s perimeter to avoid getting in the way of a human vacuuming the carpet, but trips

over the cord and disconnects the vacuum from the wall outlet. So the human has to stop and plug it back in.

- A robot hides all the musical instruments and the television remote in another room away from noisy children, which only delays the children from creating noise until they find the hiding spot.

We introduced helpfulness and some variations as a means of measuring the degree to which a robot reduced a human’s efforts, and ideal values for this measurement depend on the type of interaction. This creates a consistent, quantified evaluation of a robot’s involvement in interactions with people.

It is likely that helpfulness can play a deeper role in the decision-making process so that responsive plans adhere to helping for the entirety of the interaction, rather than be a byproduct of achieving each intermediate concrete task. We explored this potential through both expected helpfulness and a helpfulness heuristic.

Future work includes extending the efforts above to consider additional variations and interpretations of helpfulness, impacts and properties of various cost function definitions, and applications of helpfulness throughout the interactive experience. In addition to the theoretical perspective, we intend to explore the practical side of helpfulness to evaluate decision making in human-robot interaction. We would like to perform similar experiments to those in this paper with actual people rather than simulated agents. Furthermore, as people do not always perceive things computationally, it is important to compare how people evaluate a robot’s helpfulness (via Likert scales and other metrics) to our proposed quantified metric. If we expect humans to use various forms of helpfulness to assess their robot partners, then they need to match humans’ expectations in order to be informative and helpful.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful feedback and insightful perspectives.

References

- Baraglia, J.; Cakmak, M.; Nagai, Y.; Rao, R. P.; and Asada, M. 2017. Efficient human-robot collaboration: When should a robot take initiative? *The International Journal of Robotics Research* 36(5–7):563–579.

- Benton, J.; Coles, A. J.; and Coles, A. 2012. Temporal planning with preferences and time-dependent continuous costs. In McCluskey, L.; Williams, B. C.; Silva, J. R.; and Bonet, B., eds., *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling*. Atibaia, São Paulo, Brazil: AAAI Press.
- Conrad, P. R., and Williams, B. C. 2011. Drake: An efficient executive for temporal plans with choice. *Journal of Artificial Intelligence Research* 42:607–659.
- Crandall, J. W., and Goodrich, M. A. 2002. Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, 1290–1295 vol.2.
- Dragan, A., and Srinivasa, S. 2014. Integrating human observer inferences into robot motion planning. *Autonomous Robots* 37(4):351–368.
- Fox, M., and Long, D. 2003. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research (JAIR)* 20:61–124.
- Freedman, R. G., and Zilberstein, S. 2017. Integration of planning with recognition for responsive interaction using classical planners. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 4581–4588.
- Freedman, R. G. 2020. *Integrating Recognition and Decision Making to Close the Interaction Loop for Autonomous Systems*. Ph.D. Dissertation, University of Massachusetts Amherst, Amherst, MA, USA.
- Geib, C. W.; Weerasinghe, J.; Matskevich, S.; Kantharaju, P.; Craenen, B. G. W.; and Petrick, R. P. A. 2016. Building helpful virtual agents using plan recognition and planning. In Sturtevant, N., and Magerko, B., eds., *Proceedings of the Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 162–168. Burlingame, California, USA: AAAI Press.
- Gombolay, M.; Gutierrez, R.; Sturla, G.; and Shah, J. 2014. Decision-making authority, team efficiency and human worker satisfaction in mixed human-robot teams. In *Proceedings of Robotics: Science and Systems*, 1–9.
- Goodrich, M. A., and Olsen Jr., D. R. 2003. Metrics for evaluating human-robot interactions. In *Measuring the Performance and Intelligence of Systems: Proceedings of the PerMIS Workshop*.
- Hansen, E. A., and Zhou, R. 2007. Anytime heuristic search. *Journal of Artificial Intelligence Research* 28(1):267–297.
- Kulkarni, A.; Srivastava, S.; and Kambhampati, S. 2019. A unified framework for planning in adversarial and cooperative environments. In *Proceedings of the The Thirty-Third AAAI Conference on Artificial Intelligence*, 2479–2487.
- Lemaignan, S.; Warnier, M.; Sisbot, E. A.; Clodic, A.; and Alami, R. A. 2017. Artificial cognition for social human–robot interaction: An implementation. *Artificial Intelligence* 247:45–69. Special Issue on AI and Robotics.
- Levine, S. J., and Williams, B. C. 2018. Watching and acting together: Concurrent plan recognition and adaptation for human-robot teams. *Journal of Artificial Intelligence Research* 63:281–359.
- Levine, S. J. 2019. *Risk-bounded Coordination of Human-Robot Teams Through Concurrent Intent Recognition and Adaptation*. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA.
- Miura, S., and Zilberstein, S. 2020. Maximizing plan legibility in stochastic environments. In *Proceedings of the AAAI Workshops: Plan, Activity, Intent Recognition (PAIR)*, 1–7.
- Mörtl, A.; Lawitzky, M.; Kucukyilmaz, A.; Sezgin, M.; Basdogan, C.; and Hirche, S. 2012. The role of roles: Physical cooperation between humans and robots. *International Journal of Robotics Research* 31(13):1656–1674.
- Pina, P.; Cummings, M.; Crandall, J.; and Della Penna, M. 2008. Identifying generalizable metric classes to evaluate human-robot teams. In *Workshop on Metrics for Human-Robot Interaction, 3rd Ann. Conf. Human-Robot Interaction*, 13–20.
- Ramírez, M., and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1121–1126.
- Rosenthal, S., and Veloso, M. 2012. Mobile robot planning to seek help with spatially-situated tasks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2067–2073. Toronto, Ontario, Canada: AAAI Press.
- Santamaria, J. P., and Rosenbaum, D. A. 2011. Etiquette and effort: Holding doors for others. *Psychological Science* 22(5):584–588. PMID: 21515739.
- Santana, P., and Williams, B. C. 2014. Chance-constrained consistency for probabilistic temporal plan networks. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling*.
- Shah, J. A.; Conrad, P. R.; and Williams, B. C. 2009. Fast distributed multi-agent plan execution with dynamic task assignment and scheduling. In *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling*, 289–296.
- Shah, J. A. 2011. *Fluid Coordination of Human-Robot Teams*. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA.
- Shvo, M.; Klassen, T. Q.; Sohrabi, S.; and McIlraith, S. A. 2020. Epistemic plan recognition. In *Proceedings of the Nineteenth International Conference on Autonomous Agents and Multi-Agent Systems*, 1251–1259. Auckland, New Zealand: IFAAMAS.
- Singer, S. M., and Akin, D. L. A. 2011. A survey of quantitative team performance metrics for human-robot collaboration. In *Forty-first International Conference on Environmental Systems, AIAA-2011-5248*.
- Steinfeld, A.; Fong, T.; Kaber, D.; Lewis, M.; Scholtz, J.; Schultz, A.; and Goodrich, M. 2006. Common metrics for human-robot interaction. In *Proceedings of the First ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, 33–40. Salt Lake City, Utah, USA: Association for Computing Machinery.
- Wang, D., and Williams, B. C. 2015. tBurton: A divide and conquer temporal planner. In *Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence*.
- Wray, K. H., and Zilberstein, S. 2015. Multi-objective POMDPs with lexicographic reward preferences. In *Proceedings of the Twenty-Fourth International Conference on Artificial Intelligence*, 1719–1725. Buenos Aires, Argentina: AAAI Press.
- Yu, P.; Fang, C.; and Williams, B. 2015. Resolving over-constrained probabilistic temporal problems through chance constraint relaxation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 3425–3431. Austin, Texas: AAAI Press.
- Zilberstein, S. 2015. Building strong semi-autonomous systems. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 4088–4092. Austin, Texas: AAAI Press.