Robust Motion Planning and Safety Benchmarking in Human Workspaces

Shih-Yun Lo, Shani Alkoby, Peter Stone
{yunl,shani,pstone}@cs.utexas.edu
Learning Agent Research Group, The University of Texas at Austin

Abstract

It is becoming increasingly feasible for robots to share a workspace with humans. However, for them to do so robustly, they need to be able to smoothly handle the dynamism and uncertainty caused by human motions, and efficiently adapt to newly observed event. While Markov Decision Processes (MDPs) serve as a common model for formulating cost-based approaches for robot planning, other agents are often modeled as part of the environment for the purpose of collision avoidance. This practice, however, has been shown to generate plans that are too inconsistent for humans to confidently interact with. In this work, we show how modeling other agents as part of the environment makes the problem ill-posed, and propose to instead model robot planning in human workspaces as a Stochastic Game. We thus propose a planner with safety guarantees while avoiding overly conservative behavior. Finally, we benchmark the evaluation process in the face of pedestrian modeling error, which has been identified as a major concern in state-of-the-art approaches for robot planning in human workspaces. We evaluate our approach with diverse pedestrian models based on real-world observations, and show that our approach is collision-safe when encountering various pedestrian behaviors, even when given inaccurate predictive models.

1 Introduction

Planning has a long history in the robotics community, where efficiency, environmental uncertainty, and motion feasibility all central concerns. Cost-based approaches that use the MDP formulation (Watkins and Dayan 1992) can be successful for robot planning in static workspaces (Quinlan and Khatib 1993). When planning in highly-dynamic environments, however, this formulation is limited by its lack of consideration of the dynamic environmental properties.

More specifically, in MDPs, there exist intrinsic *static environment assumptions*, upon which the solutions are built. Those assumptions are: 1.a time-invariant state transition function, 2. a time-invariant state-action reward function, and therefore 3. a time-invariant state value function. As those assumptions no longer hold in dynamic environments, the accuracy of policy evaluation for long-horizon planning deteriorates when using these methods. As a result, MDP-based approaches for the traditional motion planning literature suffer from poor performance when applied in the wild.

Common solutions to the above disadvantages include frequent replanning and finite-horizon planning; those approaches update local information of the environment based on online observations, and replan periodically based on newly observed environmental conditions. Nevertheless, the lack of awareness of future conditions causes the planner to make shortsighted decisions (which leads to socially incompetent behavior for human interaction (Kruse et al. 2012)), or overly conservative behavior when considering long-horizon planning (referred to as the freezing-robot problem (Trautman and Krause 2010)). One example is the commonly-seen flow-following strategy in crowd navigation (Helbing and Molnar 1995), where people follow one another to reach shared short-term subgoals. This strategy relies on policy evaluation based on the future paths of nearby agents. With static cost formulation being applied in dynamic environments, the inaccuracy of policy evaluation makes traditional planning algorithms fail to produce paths with similar performance.

Therefore, in this work, we first propose to formulate robot planning in human environments as a multi-agent planning problem using Stochastic Games, or Markov Games (Littman 1994), to compute the dynamic state-action values that are also influenced by the states and actions of other agents (here, the humans). We use this formulation to design an online algorithm which incorporates the other agents' actions into the planning process for action evaluation.

For achieving the goal of robots being able (and allowed) to smoothly steer around humans, safety guarantees are also critically important. Traditional methods often use a worst-case assumption to ensure safety. This assumption, however, leads to overly conservative planning behaviors, degrading the smoothness of the robot motions among humans. Leveraging the notion of Stochastic Games, we propose to plan based on worst-case predictions only in period games that have a critical impact on the termination values; we seek to *plan carefully only when it matters* and achieves *safe yet not overly conservative* behavior.

Finally, we evaluate our approach using the crowd navigation domain, and discuss potential performance metrics for robot planning in human workspaces. As humans are adaptive and have heterogeneous behaviors, a perfect human behavior model is likely never available; modeling error then seems inevitable for robots to deal with on-the-fly, and we need to quantify its impact on plan quality into the evaluation process. To evaluate this before deploying robots into the wild, we simulate different human behaviors, which are based on real-world observations of pedestrian interactions with robots, and use them to evaluate our approach in unanticipated scenarios caused by modeling errors. We use these scenarios to benchmark the evaluation process in simulation, and show that our planner maintains collision-safe even evaluated with different pedestrian models.

2 Related Work

In crowd navigation domains, the freezing robot problem arises from the challenge of planning while considering the time-variant crowd-interactive dynamics in the environment (Trautman and Krause 2010). Despite efforts to introduce human factors into the planning process, traditional planning approaches which incorporate humans as a part of the environment for collision avoidance have been shown to generate motions that are neither interpretable nor socially competent (Lichtenthäler, Lorenzy, and Kirsch 2012; Kruse et al. 2012).

For incorporating the future motions of other agents to avoid collisions, the reciprocal n-body collision avoidance approach has had success in the multi-agent setting (Van Den Berg et al. 2011), where individuals assume the others move at constant speeds. This approach, and the dynamic window approach (Fox, Burgard, and Thrun 1997), are commonly used for low-level safety checks when planning in dynamic environments. However, due to the constant-speed assumption, the approach is overly conservative when interacting with real-world pedestrians, who are interactive and respond to the robot's motions.

Recently, a community proposed to solve robot planning in human workspaces as a joint multi-agent dynamics learning problem, and uses the predicted motions of all agents to plan for the robot, as if it was one of the members of the crowd (Trautman and Krause 2010; Kuderer et al. 2012; Mavrogiannis and Knepper 2016). Such joint modeling methods have been shown to be effective at outputting smooth human-mimicking trajectories, as they can capture the interactive dynamics among pedestrians. One major drawback of these approaches, however, is that the multi-agent interactive dynamics are typically learnt from data collected by human demonstrations while interacting with other humans; but humans do not act the same way around a robot compared to how they act in fully human environments. This problem has been shown to render those methods ineffective in scenarios where humans exhibit different behaviors around robots – behaviors that humans will not present in front of another human (Pfeiffer et al. 2016).

To model joint behaviors among agents – how one's action affects that of the others – another approach is to incorporate other agents' actions into the formulation of the individual's action value function. Such joint behavior formulation is widely studied in Game theory: with different player strategies, the interaction among agents evolve over time and result in different outcomes.

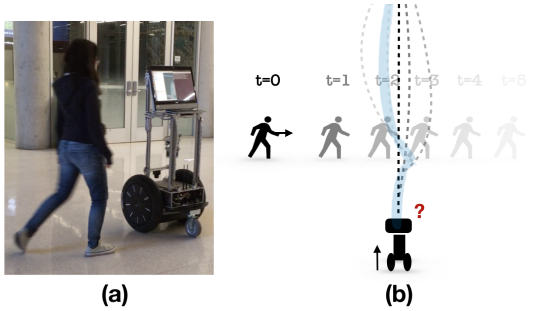


Figure 1: (a) the robot platform passing pedestrians, used in this research to collect human responses. (b) A robot's sequence of replanned paths, resulting in trajectories that change over time (marked by dash black lines, fading over the replanning time horizon) and erratic motion (highlighted by solid blue curve). Such behavior is considered *incompetent* in the social navigation literature.

For interactive agent designs in video games, the dynamics of other agents have been introduced into MDP models to simulate multi-agent planning performance (Nguyen et al. 2011; Macindoe, Kaelbling, and Lozano-Pérez 2012). In these formulations, the AI agent's current actions are assumed to be known by the humans to forward simulate their potential policies. This assumption implies a turn-taking setting, assuming full observability of the past strategies of others, including the current action. This assumption is however not valid when planning in the real world, as humans and robots act simultaneously (Sadigh et al. 2016). The turn-taking formulation is therefore not accurate for describing real-time interaction. For the game-theoretic formulation to be introduced along with Markov Decision Processes, Markov Games have been proposed as a framework for multi-agent reinforcement learning (Littman 1994), to study the interactions among multiple learning agents. In this formulation, as in Stochastic Games in Game Theory, agents act at once and contribute to the joint reward of one another. A number of studies have been proposed in this field, such as on how one agent's learning affects the final outcome and how the other agents should learn at the same time (Foerster et al. 2018). However, to accurately simulate human interactions with the robot, one requires agents modeled after humans, who have different learning mechanisms and decision-making processes from reinforcement learning agents. As humans present heterogeneous behaviors, prior work has proposed to estimate pedestrian types online based on pre-defined models (Godoy et al. 2016); here, instead of trying to perfectly predict humans, we address the inevitable modeling errors by evaluating our approach with various pedestrian models. This evaluation procedure helps bridge the gap between simulation and real-world deployment by evaluating plans under unexpected conditions (Fraichard 2007). Our proposed planner experienced zero collision under this evaluation procedure; with the demonstrated robustness under modeling errors, we suggest this criteria better ensures safety in real-world deployment, where unanticipated scenarios are of major concerns.

3 Problem Formulation

In this section, We first consider a general cost-minimization-based formulation for robot planning in human workspaces. We then introduce the dynamic environment dilemma which is the motivation for our proposed new framework. Finally, we discuss the multi-agent nature of the problem and provide the solution formulation of multi-agent planning problem.

General Cost-Minimization-Based Formulation

The robot has its state x_t at time t defined in the state space, $x_t \in X$, and its action a_t at time t defined in the action space, $a_t \in A$. The collision-free workspace is defined as a subset of the overall workspace $W_{free} \subset W$, which is defined as the feasibly reachable space given robot kinematics. The robot motion planning problem is defined to minimize the accumulated travel cost C_t, such that the robot's final state x_T ends in the specified goal configuration set $X^G \subset W_{free}$:

$$\begin{aligned} a^*_{t:T} &= \underset{a_{t:T}}{\operatorname{argmin}} \Sigma_t^T C_t, \\ s.t. \quad x_T &\in X^G, \\ x_{t:T} &\in W_{free}, \\ x_{t+1} &= \mathcal{T}(x_t, a_t), \forall t, \end{aligned} \tag{1}$$

where $\mathcal{T}$ is the state transition function (or robot dynamics function). The sequence of a variable v from t to T is denoted by $v_{t:T}$.

A common approach for solving the motion planning problem is to assume C_t is a function of the state-action pair, x_t and a_t, represented by $Cost(x_t, a_t)$; we can then assign a negative end state cost C_T for arriving at the goal, represented by $Cost_{to-go}(x_T)$, and transitions out of free space W_{free} to be of high cost to ensure safety. For example:

$$\begin{aligned} a^*_{t:T} &= \underset{a_{t:T}}{\operatorname{argmin}} \Sigma_t^T Cost(x_t, a_t) + Cost_{to-go}(x_T), \\ s.t. \quad x_{t+1} &= \mathcal{T}(x_t, a_t), \forall t \end{aligned} \tag{2}$$

where

$$Cost_{to-go}(x_T) = -1000, x_T \in X^G,$$

$$Cost(x_t, a_t) = \infty, \forall x_t \notin W_{free},$$

to encourage goal-reaching and collision-avoidant behavior. We then can apply some sequential optimizer to solve for the optimal action sequence $a^*_{t:T}$ which follows the state transition constraint and minimizes the overall travel cost, with guarantees to reach the goal in a collision-safe manner. This common formulation follows the MDP setting, which we later refer to as the single-agent MDP formulation and its solution at time t is the *single-agent optimal policy*.

The Dynamic Environment Dilemma

The general formulation for robot planning in human workspaces laid out above relies on the assumption that objects in the robot's environment are static. In many scenarios however, this does not hold. When encountering dynamic objects in the environment, W_{free} changes over time. This means that the optimal sequence solved for time t may no longer hold at time $t+1$. An illustration of this challenge is shown in Fig. 1-Right. The problem raised by introducing dynamic objects into the environment is referred to here as a violation of the static environment assumption in the MDP formulation: with W_{free} being time-variant, $Cost$ becomes time-variant as well.

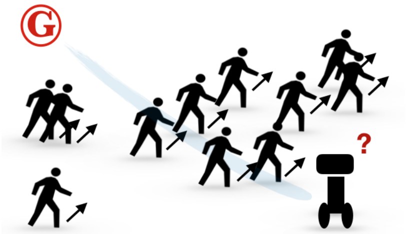


Figure 2: A robot blocked by a flow of the crowd from reaching its destination. Despite the current infeasibility of finding a path all the way to the goal, crowd configurations change over time, yielding space for it to pass through when going closer.

In the motion planning literature, online replanning is a common practice to deal with dynamic environments (Koenig and Likhachev 2002; Quinlan and Khatib 1993), which however leads to inefficient, inconsistent, and even awkward motions for humans to confidently interact with (Lichtenthäler, Lorenzy, and Kirsch 2012; Kruse et al. 2012), as shown in Fig. 1. For long-horizon planning, to ensure collision safety, overly-conservative behavior arises, due to the inability to incorporate future variations into the cost function formulation, as shown in Fig. 2.

We refer the situation as the **dynamic environment dilemma**. To resolve this, instead of introducing other agents as part of the environment and solving the problem in a single-agent MDP setting, we propose to re-define the planning domain using a *multi-agent* MDP setting, which restores the validity of the static environment assumptions by considering the *simultaneous actions* of other agents and their *joint state space*.

Multi-agent MDPs vs. Stochastic Games

In interactive agent design and human-robot interaction, multi-agent MDPs (MAMDPs) can be used to forward simulate human policies by chaining human policy simulation after robot state transitions (Macindoe, Kaelbling, and Lozano-Pérez 2012; Nikolaidis et al. 2016). This model assumes that human agents have access to the action the AI agent is about to make, as if they are omniscient. The underlying game setting follows the turn-taking formulation.

However, in reality, agents act at the same time (illustrated in Fig. 3); in the literature of robot planning in human workspaces (Kretzschmar, Kuderer, and Burgard 2014; Trautman and Krause 2010; Mavrogiannis and Knepper 2016), state-of-the-art approaches generate smooth motions in real-world interaction by *planning while concerning the effects of the simultaneous actions of other agents*(Sadigh et al. 2016). Our proposed dynamic environment dilemma gen-

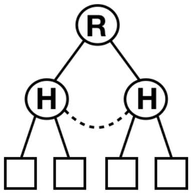


Figure 3: The linked dash line is applied to denote agents sharing the same history information, which exclude the current actions of the other agents (here, the robot). This is applied since in real-time interactions agents act simultaneously and thus each agent does not know the current actions of the others.

eralizes the issues addressed in these work to improve real-world planning in human environments. And to resolve the dilemma, we propose to model the problem using Stochastic Games from the Game Theory literature, in which the cost/reward received after one acts is dependent on other agents' states and actions *at the same time*. This is also true for the state-action value function Q and state value function V.

In Stochastic Games, N agents act at the same time t: the joint-action $a_t = (a_t^1, a_t^2, ..., a_t^N) \in A$ is defined in the joint action spaces of all agents $A = A^1 \times A^2 ... \times A^N$; the joint-state $x_t = (x_t^1, x_t^2, ..., x_t^N) \in X$ is defined in the joint state space of all agents $X = X^1 \times X^2 ... \times X^N$. The state transition function: $\mathcal{T} : X \times A \rightarrow X$ affects the utility of each agent under the same joint-action over time. Time is discretized, and game *periods* are defined as follows. At the start of each period t, each agent selects an action $a_t^i, i = 1 : N$; the transition function $\mathcal{T}$ takes in the current state x_t and determines (probablistically) the state at the beginning of the next period x_{t+1}. The game starts at the initial period $t = 0$ and terminates at the final period $t = T$. The duration of each period is selected along with the lookahead H based on the computation can be afforded while maintaining real-time performance, described in detail in Sec. 5. [1]

In Markov Games, the time-variant utilities are referred to as rewards (inverse of $Cost$), which depend on the joint-states x_t.[2] The reward $\mathbf{r}_t^i \in \mathbb{R}$ of an agent i at time t is defined as follows: $\mathbf{r}_t^i = r^i(x_t, a_t^i, a_t^{-i})$, where r^i is the agent's reward function, and a_t^{-i} denotes actions of all agents except agent i. This formulation conveniently incorporates x_t, the time-variant states of other agents, into the reward/cost formulation, which resolves the dynamic workspace issue. It brings out the notion of *planning while considering the effects of the simultaneous actions of other agents*, upon which we base our solution to robot planning in human workspaces.

The Optimal Solution Formulation

To find the optimal policy of this multi-agent MDP planning problem, we first define a multi-agent policy of agent i as π^i, which takes in the joint-state x_t and outputs agent action a_t^i at time t. We first consider the *known* policies of the other agents π^{-i}; the state-action value function Q of agent i executing policy π^i while other agents follow policy π^{-i} is defined as[3]:

$$Q^{\pi^i|\pi^{-i}}(x_t, a_t^i, a_t^{-i}) = r^i(x_t, a_t^i, a_t^{-i}) + \mathbb{E}_{x_{t+1}}[V^{\pi^i|\pi^{-i}}(x_{t+1})], \tag{3}$$

where $V^{\pi^i|\pi^{-i}}$ is the value function V of agent i executing policy π^i while others using π^{-i}. Note that the expectation is taken over x_{t+1} conditioned on the state transition function $\mathcal{T}$, which is omitted throughout the paper for simplicity. The value function of the *optimal* policy of agent i, when other agents execute π^{-i}, is defined as:

$$V^{i|\pi^{-i}}(x_t) = \max_{a_t^i} \mathbb{E}_{a_t^{-i} \sim \pi^{-i}(x_t)}[Q^{i|\pi^{-i}}(x_t, a_t^i, a_t^{-i})], \tag{4}$$

where $Q^{i|\pi^{-i}}$, the optimal state-action value of agent i given π^{-i}, is defined recursively:

$$Q^{i|\pi^{-i}}(x_t) = \max_{a_t^i} \mathbb{E}_{a_t^{-i} \sim \pi^{-i}(x_t)}[r^i(x_t, a_t^i, a_t^{-i}) + V^{i|\pi^{-i}}(x_{t+1})]. \tag{5}$$

The optimal action of agent i at time t is therefore:

$$a_t^{i*} = \operatorname*{argmax}_{a_t^i} \mathbb{E}_{a_t^{-i} \sim \pi^{-i}(x_t)}[Q^{i|\pi^{-i}}(x_t, a_t^i, a_t^{-i})]. \tag{6}$$

Note that the optimal action a_t^{i*} is defined in the joint state space X, and it depends on agent i's estimate of other agents' policies π^{-i}.

Planning in Real World

Despite the benefits of the Stochastic Game framework for robot planning in human environments, there remain challenges to deploying robots in the real world. In this work, we propose solutions through planning and evaluation approaches, to address *safety guarantees*, and *evaluation under modeling errors*, as detailed below:

Imperfect Prediction of Human Behaviors While in principle, optimizing Eq. 6 leads to an optimal solution to robot planning, note that it relies on having an accurate model of pedestrian motion π^{-i}. In practice, there is no easy way to obtain such a model, as people exhibit heterogeneous (Godoy et al. 2016) and highly adaptive behavior (Nikolaidis et al. 2016). Since planning with incorrect models could lead to safety concerns in human workspaces, in this paper we seek planning methods that are *robust* to modeling errors in the pedestrian transition function.

[1] Here we use periods instead of horizons as in the planning literature, to distinguish our multi-agent formulation from the traditional single-agent setting.

[2] We start with $Cost$ for consistency with traditional planning literature, and continue with the notion of reward for convenience of consistency with MDP planning literature.

[3] The state-action value function Q in the single-agent setting is generally defined as: $Q(x_{t+1}) = \mathbf{r}_t(x_t, a_t) + V(x_{t+1})$. We later refer Q^i as the single-agent state-action value of agent i.

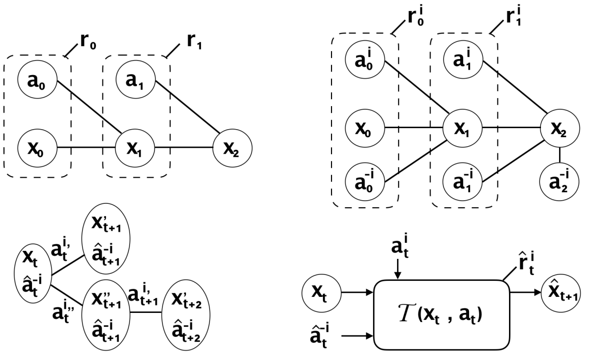


Figure 4: Comparison of single-agent MDPs (Top-Left) and stochastic games (Top-Right). A tree with all-agent simulation expands its node x_t (Bottom-Left) with choice of actions (here, $a_t^{i,}$ or $a_t^{i,,}$) to the children nodes (here, $x^{,}_{t+1}$ or $x^{,,}_{t+1}$). Since actions of other agents a_t^{-i} are unknown, to expand from a node x_t, the state transition function $\mathcal{T}$ needs to first sample potential actions of other agents $\hat{a}_t^{-i}$ to estimate the reward $\hat{r}_t^i$ after taking action a_t^i and arriving at the next state $\hat{x}_{t+1}$ (Bottom-Right).

Performance Degrades for Safety Guarantees To ensure collision safety, traditional robust planning often formulates the problem through the maximin operation, often leading to overly conservative behaviors. This property prevents the robot from navigating agilely and smoothly among human crowds, for which we propose improvement while still guaranteeing safety. This is detailed in Sec. 4. Here, we define safety guarantees as ensuring that the robot does not go within a safety margin from any human while exceeding a critical safety speed, as detailed in Sec. 5.

Evaluation in Simulation Due to the inevitable prediction errors in planning in human environments, evaluation results in simulation often cannot provide sufficient information for real-world deployments. We therefore seek new metrics that better inform the smoothness, efficiency, and safety criteria from simulation trials. The proposed metrics and the experimental results of our proposed approach are shown in Sec. 6 and Sec. 7.

4 Methodology

In this section, we first introduce our proposed planning method based on Stochastic Games. We use tree search, for the convenience of forward simulation/state transitions in MDPs with robot non-holonomic constraints, upon which we introduce an approach featuring robust collision-safe planning, to resolve the challenges raised in Sec. 3 regarding robot planning in the real world.

Tree Structure with All-agent Rollout

A tree starts with a root node x_t, and it expands by forward simulating the state-action pair (possibly through a stochastic function): $x_{t+1} \sim \mathcal{T}(x_t, a_t)$, and a reward $r_t = r(x_t, a_t)$ is received. An illustration of a graphical model of single-agent MDP is shown in Fig. 4: Top-Left.

However, when planning in Markov Games, both the reward function $r^i(x_t, a_t^i, a_t^{-i})$ and the transition function $\mathcal{T}(x_t, a_t^i, a_t^{-i})$ involve other agents' actions, which are not available until they are observed (Fig. 4: Top-Right). Therefore, we need to sample their potential actions for reward estimate $\hat{r}_t^i$ and state transition estimate $\hat{x}_{t+1} \in X$, (Fig. 4: Bottom-Right), to expand the tree with all-agent rollout (Fig. 4: Bottom-Left). Each node then maintains a number of samples of *belief actions* of other agents, which can be later corrected online when new observations come in. The reward r_t^i is then estimated through the unweighted sample averages: $\hat{r}_t^i = \frac{1}{K}\sum_{k=1}^{K} r^i(x_t, a_t^i, a_{t,k}^{-i})$; and the joint state x_{t+1} follows the same fashion:

$$\hat{x}_{t+1} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{T}(x_t, [a_t^i, a_{t,k}^{-i}]). \tag{7}$$

The approach is closely related to Partially Observable Monte Carlo Planning (Silver and Veness 2010), where the *belief state* is maintained by K samples.

Compared to the turn-taking formulation introduced in Sec. 3, to roll out the multi-agent state transition, our all-agent rollout better captures the simultaneous-action nature of real-time interactions, as illustrated in Fig. 3.

Planning for Collision Coordination: A Finite-period Game

When sharing a workspace, agents have to plan to avoid colliding with one another; in situations where they are aware of a potential collision, agents should adjust their motions early, to coordinate passing smoothly. The value of the final passing depends on the joint-actions of previous periods; planning lasts for a finite number of periods until the collision threat is resolved. We define the game to terminate once agents' goal-oriented policies no longer lower the values of each other [4].

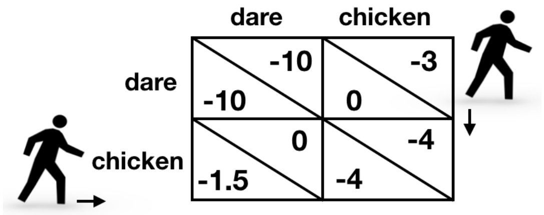
During the period before termination, which we refer to as the *final period*, agents receive immediate penalties (negative reward signals) due to the expected collision when following their goal-oriented policies. They therefore need to make sure they coordinate in the final period, to avoid this high cost. Beyond that point, agents can safely recover back to their goal-oriented policies. Therefore, when planning for collision coordination, we only need to consider the cumulative rewards up to time $t = T - 1$, and the final-period coordination value Q_T^i:

$$a^{i*}_{0:T} = \operatorname*{argmax}_{a^i_{0:T}} \mathbb{E}_{a^{-i}_{0:T}, x_{0:T}|\pi^{-i}} [\sum_{t=0}^{T-1} r^i(x_t, a_t^i, a_t^{-i}) + Q_T^i(x_T, a_T^i, a_T^{-i})]. \tag{8}$$

Note that, instead of $Q_T^{i|\pi^{-i}}$, we use Q_T^i, since we expect no interaction after the final period [5]. An example of the final-period action value is shown in Fig. 5.

[4] The goal-oriented policy is the single-agent optimal policy *without* considering the dynamic objects (here, humans) in the environment for W_{free} construction. It maintains the static-environment assumptions, resulting in a policy that acts as if no other agents are around

[5] This assumption holds among goal-oriented agents, but does



	dare	chicken
dare	-10 -10	-3 0
chicken	0 -1.5	-4 -4

Figure 5: An example final-period value of two-agent crossing: as the advancing agent has higher time delay to yield to the other, it has the format as an asymmetric Chicken game.

A Game-theoretic Decision-making Model

Robust Planning for Safety Guarantees

To ensure collision safety, a common practice is to use a worst-case analysis for reward estimates. Following our finite-period planning, the objective becomes:

$$a^{i*}_{0:T} = \underset{a^{i}_{0:T}}{\operatorname{argmax}}\ \underset{a^{-i}_{0:T}}{\min} \mathbb{E}_{x_{0:T}}[\sum_{t=0}^{T-1} r^i(x_t, a^i_t, a^{-i}_t) + Q^i_T(x_T, a^i_T, a^{-i}_T)] \tag{9}$$

which however results in overly conservative behavior, commonly seen in robust planning. To avoid such behavior, we leverage the following observation: when planning for collision coordination, the large collision penalty does not apply until the *final period*, and thus the planner does not need to plan conservatively until then. Therefore, for *collision-safe* yet *not overly conservative* planning, we propose to use the average reward for $t = 0 : T-1$ as in Eq. 8, and use the worst-case state-action value estimate at $t = T$:

$$\begin{aligned} a^{i*}_{0:T} = \underset{a^{i}_{0:T}}{\operatorname{argmax}}\ & \mathbb{E}_{a^{-i}_{0:T}, x_{0:T}|\pi^{-i}}[\sum_{t=0}^{T-1} r^i(x_t, a^i_t, a^{-i}_t)] \\ & + \underset{a^{-i}_{T}}{\min} Q^i(x_T, a^i_T, a^{-i}_T), \end{aligned} \tag{10}$$

We refer this notion as to **plan carefully only when it matters**.

5 Problem Instantiation

We instantiate our problem formulation in the navigation domain, where a robot moves in a shared workspace with humans. This scenario is motivated by service or guidance robots in malls and museums. Although service robots are designed to assist humans, they have specified users who may have conflicting interests with other users. For example, a guidance robot may have path conflicts with people it is not leading. We therefore define robot reward function to reflect a weighted value (by parameter w) among the individual interests of all parties involved. With the weight mostly on the robot itself, it leads to aggressive behavior; with it evenly on all agents, it leads to collaborative behavior; and with it mostly on others, it presents altruistic behavior. An example is shown in Fig. 6. In this section we consider ways to reduce the computational complexity while planning online, and provide a description of the search algorithm used. Since pedestrian modeling is centrally important to our experiments, we devote a separate section to it (Sec. 6).

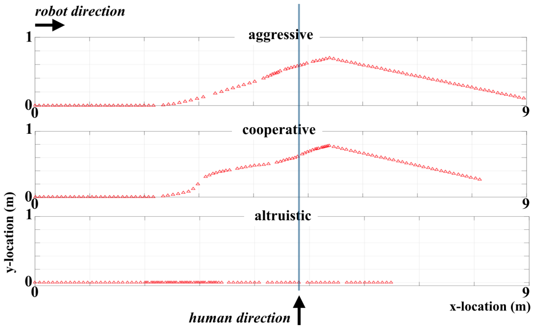


Figure 6: Example trajectories under different robot objective functions (x-y in meters): a robot goes from $-x$ toward $+x$ while avoiding a human going from $-y$ toward $+y$ at $x = 4.8$ (indicated by the solid blue line). Trajectories end after 12 sec. In the initial condition the robot will arrive at the intersection slightly later than the human. The altruistic setting optimizes the pedestrian's efficiency much more than that of the robot, resulting in yielding behaviors that always waits until the pedestrian passes and results; the aggressive setting is the opposite, resulting in high robot travel efficiency by reaching the furthest at the end. The cooperative setting puts equal weights on the efficiency of both agents, and produces passing behaviors that less hinder the pedestrian considering his/her future trajectories.

Robot Motion Generation

In our implementation, a robot stays at a nominal speed during normal navigation, and can vary its speed between [0.3, 1.3] m/s and its acceleration between [-0.4, 0.4] m/s^2, in collision-dangerous situations. Collision-dangerous situations are defined as a potential collision will occur within $3s$ when both agents follow self-interested policies. Our choice of speed, acceleration and time frame prior to a collision follows the study on pedestrian crossing for crowd simulation (Paris, Pettré, and Donikian 2007), in which it is found that humans start adapting their motions about $2.5s$ ahead of a potential collision in the most collision-dangerous situations (with two pedestrian having the same estimated arrival time at their path intersection).

In the nominal operation phase, we sample navigation actions with constant heading acceleration (rotation around z axis) of [-15, 15] deg/s^2. This encourages path exploration under constant speed (0.7 m/s). In collision avoidance phases, we sample quartic (4th-order) polynomials with safety margin (estimated minimum distance with the other agent, using his/her current speed) of [0.2, 1.8] m.

During node expansions, new robot actions and human actions are sampled, and potential collisions are checked under the condition that the distance between two agents is within 0.9 m while the velocity of the robot is greater than 0.3 m/s. We define the critical safety speed as 0.3 m/s (defined as the safety action a_s) instead of stopping, since

not hold among adversarial agents, who intentionally block the others even when their paths are clear. We do not consider adversarial agents here.

people are very capable of avoiding low-speed objects. Additionally, full stops are considered to be unnatural in human crowds (Trautman and Krause 2010). Only valid nodes (those satisfying a collision-free check) are added into the tree.

Online Computation

We plan in a receding-horizon fashion. This means that the planner replans online at each period, up to the final period, estimated based on forward simulation. To ensure search is complete and the solution is returned at each period, the planner needs to balance the computation of each run, which depends on the number of particles K, sampled actions $|A|$, and search depth H (or lookahead, periods to $t = T$), with the complexity of: $K|A|^H$ [6].

Here we use $K = 1$ in our implementation when $t < T$, which leads to the nominal/most-likely actions being selected, as if we assume maximum likelihood observations in belief space planning (Platt Jr et al. 2010). We do so until it is the final period $t = T$, and apply $K = 10$ to sample for worst-case estimates, to maintain the safety mechanism from Eq. 10. We sample $5 - 15$ actions depending on how crucial the state is to collision coordination. Finally, we consider $H = 3$, to keep the worst-case complexity under 500 nodes to search for for real-time computation. The time duration of each period is 1 sec, to ensure the robot starts planning 4 secs before the estimated collision timing, since humans usually reacting to collision threats $2.5s$ ahead on average (Paris, Pettré, and Donikian 2007). Due to the tree structure, once a node is visited, it is put in the closed set (for the evaluated nodes), but not considered in the future for re-evaluation.

6 Human Behavior Modeling

It is known that humans have heterogeneous behaviors. Moreover, as suggested in state-of-the-art approaches that have been deployed for real-world interactions (Trautman et al. 2015; Pfeiffer et al. 2016), humans interact with robots much differently than with other humans. As a result, no sufficiently accurate human model is available for robot planning. Thus, in order to obtain a level of assurance that the robot will behave safely when deployed in the wild, it is important to evaluate it when its human simulation model is inaccurate. To construct such scenarios, and to evaluate a planner on simulated humans in general, it is important to know the behavior assumptions to interpret the results accordingly. It is dangerous to evaluate an approach using a similar simulated human behavior to what being used in the planner for forward simulation (Macindoe, Kaelbling, and Lozano-Pérez 2012), as it hides the effect of modeling inaccuracy. Therefore, in this section, we leverage pedestrian simulators, specifically, social force models with explicit collision avoidance (Karamouzas et al. 2009), interpret their underlying behavior assumptions using a game-theoretic decision-making model, and then modify those assumptions to simulate behaviors we observed in real-world interactions. We use them to evaluate our approach, and discuss about the performance brought by model inaccuracy, to benchmark the worst-case scenario.

Popular approaches for behavior modeling are data-driven, either from the behavior cloning community (supervised learning), or inverse optimal planning community. In the latter, agents are assumed to be *rational*, which means their decisions optimize a certain objective. Here, we consider the agent-based models in crowd simulation (Helbing and Molnar 1995; Paris, Pettré, and Donikian 2007; Karamouzas et al. 2009). Among those models, nominal goal-driven navigation and interactive collision avoidance motions are simulated through reactive policies. We adapt the social force model with collision prediction (SF-CP (Karamouzas et al. 2009)) to sample avoidance motions. In their work, collision avoidance behaviors are based on the relative position at their closest point, which can be categorized into two groups: one is anticipated to pass the path intersection earlier, which then accelerate (to attempt to pass in front); the other is anticipated pass later, which then decelerate (to attempt to yield).

In real world, however, those reactions are not based on perfect timing estimates, and people make attempts based on other criteria as well. Therefore, we sample avoidance motions for both attempts, by sampling relative velocity estimates for both earlier and later arrival timings, and use a game-theoretic decision-making model to decide which attempt to go for, to simulate heterogeneous behaviors we observe in real world.

For generating the worst case scenario we model the pedestrian collision-avoidance (crossing) scenario as an (two-agent) asymmetric game of Chicken, shown in Fig. 5, where each agent knows the value estimates $Q^{i,-i}$ of their last-period actions a_T, and their decisions are based on the design of their own objectives, and the *inference* of the other agents' strategies, as listed in Eq. 6.

When agents share the same objective and it is of common knowledge to all agents i.e., all agents know they share an objective and they all know others know that and so on, all agents share the same optimal policy. In order to solve this case, we can treat it as if one agent has full control to the others, which all optimize that agent's state-action value function Q^i:

$$a^{i*}_{0:T} = \underset{a^i_{0:T}}{\text{argmax}} \max_{a^{-i}_{0:T}} \mathbb{E}_{x_{0:T}}[\sum_{t=0}^{T-1} r^i(x_t, a^i_t, a^{-i}_t) + Q^i_T(x_T, a^i_T, a^{-i}_T)]. \tag{11}$$

If an agent's policy is to maximize the social welfare and it assumes the others know that and do the same, the overall function in Eq. 11 converges to the coordination policy where agents yield whenever it has later arrival timing and vice versa, such that the joint efficiency is maximized. This behavior, which is simulated in SF-CP, is referred here as the *reciprocal* behavior, as agents are cooperative and assume the others are the same. In the real world, there are many cases in which humans act strategically and thus do not always act according to the reciprocal behavior assumptions. Some people constantly yield and wait for the others to pass first, while others are doing the opposite. This can be ob-

[6] If we maintain K samples and do not apply sample average for rollouts, as suggested in Eq. 7, the complexity will be $|K|^H|A|^H$

served not only among crowds, but also among pedestrians who encounter a robot for the first time [7]. Among pedestrians who exhibit the constant-yielding behavior, some suggested that they do not know how to predict what the robot will do; we therefore refer those pedestrians as being *cautious*, and simulate their decisions through the maximin operation, as suggested in Eq. 9. Among the pedestrians who exhibited the non-yielding behavior, some strong provided feedback that the robot should have waited for them. We simulate such behavior through Eq. 11, with self-interested objective and altruistic assumptions for the other agent. We refer to such behavior as being *aggressive*, which assumes both parties attempt to maximize the one agent's individual efficiency.

7 Performance Metrics and Experiments

In this section we evaluate our proposed robust planner under randomized initial conditions, with the proposed three types of pedestrian behaviors in simulation. We consider plan *safety*, *efficiency*, and *smoothness* as performance metrics. We show performance deterioration when different types of pedestrians are simulated, and collision safety is still ensured among all scenarios. We sample 100 initial configurations for testing in a two-agent crossing scenario, with the robot starting at its nominal speed, 0.7 m/s, the human at a speed range of [0.9, 1.3] m/s, and the initial positions controlled such that the estimated arrival timing difference range of [-0.8, 0.8] s. We evaluate the planner using reciprocal human model (SF-CP) for forward simulation, and compare the performances among crossing 1. aggressive, 2.cautious, and 3. reciprocal pedestrians. As suggested in Eq. 11, we expect optimal joint efficiency among reciprocal-reciprocal agents with equal travel time, highest pedestrian individual efficiency with the aggressive model and the opposite with the cautious model.

We consider the **safety** metric as: the distance between two agents never comes within $0.6m$ while the robot speed is greater than $0.3m/s$. Efficiency is measured by **travel time**. We look into number of executed minimum-speed actions (a_s) as an indicator of how **smooth** the crossing interaction is. The result can be seen in Fig. 7. We can see the planner experiences **zero** collisions among the three types of simulated pedestrians [8]. Due to the conservative prediction at final periods, the expected safety actions a_s in the first periods of planning are more than the executed ones at the final periods. This is true with both reciprocal (of accurate prediction) and cautious pedestrian models, but the opposite occurs

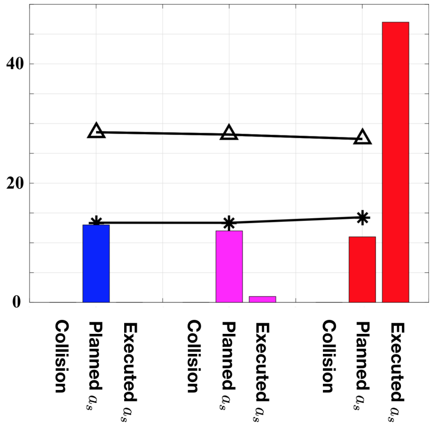


Figure 7: Comparison of robot planning with different simulated humans: cautious (Left), reciprocal (Middle), and aggressive (Right). The bar graph lists three metrics: counts of collisions, counts of a_s planned at first period, and counts of a_s executed at final period. The lower starred solid line indicates the robot's average travel time, and the upper triangular-marked solid line indicates the average joint travel time.

when encountering aggressive agents, where the robot's individual travel efficiency deteriorates due to frequent slow-downs. Although it appears to have highest joint efficiency on average, unexpected slow-downs in general cause non-smooth interaction; along with the frequent close-distance interaction, we expect extra delays to both agents in the real world.

8 Conclusion and Future Work

We introduce a method for robot planning in human workspaces as a Markov Game, to resolve the **dynamic environment dilemma** in the motion planning literature when planning in dynamic workspaces. We then proposed an algorithm that provided safety guarantees in simulated environments yet prevented the robot from exhibiting overly conservative behavior through final-period worst-case simulation, seeking to **plan carefully only when it matters**. We also proposed pedestrian behavior assumptions based on real-world observations, to benchmark the worst-case scenarios caused by modeling inaccuracy, which is one of the most difficult issues to deal with for state-of-the-art approaches as they have been deployed in the real world. The proposed approach relies on accurate modeling of the action space of other agents A^{-i} for safety guarantee (as computed in Eq. 10), which can not be validated for real-world application unless deployed in real human workspaces. Evaluation in real environments is then necessary to better support the safety guarantee in the real world. We did not detail the choice of heuristic in this paper, and desire to better leverage those applied in grid-based multi-agent path coordination and scheduling for fast online computation. Lastly, there is limited literature on human behavior using multi-agent decision-making models; we desire to extend the models to incorporate more diverse interactive behavior, including behavior adaptation, which is commonly seen in human-robot interaction.

[7] We put a robot in a public space, running a policy that never slows down until imminent threat of collision is detected. We observed pedestrian responses, and ask them questions about what they thought of the robot's behavior afterwards.

[8] In real-world interactions, another worst-case scenario is to encounter a robot-interested pedestrian, who follows the robot after it has passed, and block the robot when passing in front. We acknowledge but do not explicitly consider this behavior here for performance evaluation. A potential colliding situation when simulating this adversarial type of agents for crossing is when they continue with high speed while the robot has passed in front and slowed down out of safety concern.

References

[Foerster et al. 2018] Foerster, J.; Chen, R. Y.; Al-Shedivat, M.; Whiteson, S.; Abbeel, P.; and Mordatch, I. 2018. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 122–130. International Foundation for Autonomous Agents and Multiagent Systems.

[Fox, Burgard, and Thrun 1997] Fox, D.; Burgard, W.; and Thrun, S. 1997. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine* 4(1):23–33.

[Fraichard 2007] Fraichard, T. 2007. A short paper about motion safety. In *IEEE int. conf. on robotics and automation*.

[Godoy et al. 2016] Godoy, J.; Karamouzas, I.; Guy, S. J.; and Gini, M. 2016. Moving in a crowd: Safe and efficient navigation among heterogeneous agents. In *Proc. Int. Joint Conf. on Artificial Intelligence*.

[Helbing and Molnar 1995] Helbing, D., and Molnar, P. 1995. Social force model for pedestrian dynamics. *Physical review E* 51(5):4282.

[Karamouzas et al. 2009] Karamouzas, I.; Heil, P.; van Beek, P.; and Overmars, M. H. 2009. A predictive collision avoidance model for pedestrian simulation. In *International Workshop on Motion in Games*, 41–52. Springer.

[Koenig and Likhachev 2002] Koenig, S., and Likhachev, M. 2002. Dˆ* lite. *Aaai/iaai* 15.

[Kretzschmar, Kuderer, and Burgard 2014] Kretzschmar, H.; Kuderer, M.; and Burgard, W. 2014. Learning to predict trajectories of cooperatively navigating agents. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 4015–4020. IEEE.

[Kruse et al. 2012] Kruse, T.; Basili, P.; Glasauer, S.; and Kirsch, A. 2012. Legible robot navigation in the proximity of moving humans. In *Advanced Robotics and its Social Impacts (ARSO), 2012 IEEE Workshop on*, 83–88. IEEE.

[Kuderer et al. 2012] Kuderer, M.; Kretzschmar, H.; Sprunk, C.; and Burgard, W. 2012. Feature-based prediction of trajectories for socially compliant navigation. In *Robotics: science and systems*. Citeseer.

[Lichtenthäler, Lorenzy, and Kirsch 2012] Lichtenthäler, C.; Lorenzy, T.; and Kirsch, A. 2012. Influence of legibility on perceived safety in a virtual human-robot path crossing task. In *RO-MAN, 2012 IEEE*, 676–681. IEEE.

[Littman 1994] Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*. Elsevier. 157–163.

[Macindoe, Kaelbling, and Lozano-Pérez 2012] Macindoe, O.; Kaelbling, L. P.; and Lozano-Pérez, T. 2012. Pomcop: Belief space planning for sidekicks in cooperative games. In *AIIDE*.

[Mavrogiannis and Knepper 2016] Mavrogiannis, C. I., and Knepper, R. A. 2016. Decentralized multi-agent navigation planning with braids. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics. San Francisco, USA*.

[Nguyen et al. 2011] Nguyen, T.-H. D.; Hsu, D.; Lee, W. S.; Leong, T.-Y.; Kaelbling, L. P.; Lozano-Perez, T.; and Grant, A. H. 2011. Capir: Collaborative action planning with intention recognition. In *AIIDE*.

[Nikolaidis et al. 2016] Nikolaidis, S.; Kuznetsov, A.; Hsu, D.; and Srinivasa, S. 2016. Formalizing human-robot mutual adaptation: A bounded memory model. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, 75–82. IEEE Press.

[Paris, Pettré, and Donikian 2007] Paris, S.; Pettré, J.; and Donikian, S. 2007. Pedestrian reactive navigation for crowd simulation: a predictive approach. In *Computer Graphics Forum*, volume 26, 665–674. Wiley Online Library.

[Pfeiffer et al. 2016] Pfeiffer, M.; Schwesinger, U.; Sommer, H.; Galceran, E.; and Siegwart, R. 2016. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2096–2101. IEEE.

[Platt Jr et al. 2010] Platt Jr, R.; Tedrake, R.; Kaelbling, L.; and Lozano-Perez, T. 2010. Belief space planning assuming maximum likelihood observations.

[Quinlan and Khatib 1993] Quinlan, S., and Khatib, O. 1993. Elastic bands: Connecting path planning and control. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, 802–807. IEEE.

[Sadigh et al. 2016] Sadigh, D.; Sastry, S.; Seshia, S. A.; and Dragan, A. D. 2016. Planning for autonomous cars that leverages effects on human actions. In *Proceedings of the Robotics: Science and Systems Conference (RSS)*.

[Silver and Veness 2010] Silver, D., and Veness, J. 2010. Monte-carlo planning in large pomdps. In *Advances in neural information processing systems*, 2164–2172.

[Trautman and Krause 2010] Trautman, P., and Krause, A. 2010. Unfreezing the robot: Navigation in dense, interacting crowds. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 797–803. IEEE.

[Trautman et al. 2015] Trautman, P.; Ma, J.; Murray, R. M.; and Krause, A. 2015. Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research* 34(3):335–356.

[Van Den Berg et al. 2011] Van Den Berg, J.; Guy, S. J.; Lin, M.; and Manocha, D. 2011. Reciprocal n-body collision avoidance. In *Robotics research*. Springer. 3–19.

[Watkins and Dayan 1992] Watkins, C. J., and Dayan, P. 1992. Q-learning. *Machine learning* 8(3-4):279–292.