POMCoP: Belief Space Planning for Sidekicks in Cooperative Games

Owen Macindoe, Leslie Pack Kaelbling, and Tomás Lozano-Pérez
CSAIL, 32 Vassar Street
Cambridge, Massachusetts, 02139

Abstract

We present POMCoP, a system for online planning in collaborative domains that reasons about how its actions will affect its understanding of human intentions, and demonstrate its use in building sidekicks for cooperative games. POMCoP plans in belief space. It explicitly represents its uncertainty about the intentions of its human ally, and plans actions which reveal those intentions or hedge against its uncertainty. This allows POMCoP to reason about the usefulness of incorporating information gathering actions into its plans, such as asking questions, or simply waiting to let humans reveal their intentions. We demonstrate POMCoP by constructing a sidekick for a cooperative pursuit game, and evaluate its effectiveness relative to MDP-based techniques that plan in state space, rather than belief space.

Introduction

Digital games often feature computer controlled sidekicks or allies that the player must closely collaborate with to achieve their goals. Because players spend more time in contact with these non-player characters (NPCs) than with typical NPCs, and because coordinating with them is often crucial to progression in the game, their behavioral systems have a strong impact on a player's gameplay experience, for better or worse. The controllers for these NPCs are generally carefully crafted behavior trees (Isla 2005) or hierarchical finite state machines. Building these systems burdens designers and programmers with the difficult task of anticipating the possible situations in which an NPC may find itself in order to specify its response.

Alternatively, some games have used automatic planning techniques such as goal oriented action planning (Orkin 2004) or hierarchical task networks (Straatman, Verweij, and Champandard 2009) to automatically generate controllers and relieve some of the development burden on programmers and designers. Recently researchers have also proposed modeling games as Markov decision processes (MDPs) and using a combination of MDP solvers and Bayes filters to construct controllers that infer human intentions from observing their actions and act to support those intentions (Ngyuen et al. 2011).

We present a new method, called partially observable Monte-Carlo cooperative planning (POMCoP), for creating NPC sidekicks that can reason about how their actions affect their knowledge about the intentions of a human collaborator. For instance, a POMCoP sidekick may plan to ask the human directly about his or her intentions if it is confused, or plan to take actions which hedge against its uncertainty about a player's goals. POMCoP achieves this by planning directly in belief space, which is the space of probability distributions over possible configurations of the world, including unobservable elements such as human intentions. Planning in belief space entails reasoning about how an agent's actions will change its own beliefs. This approach contrasts with state space planning approaches, which assume that the world's dynamics are known and its state is completely observable, and so do not take into account how actions can reveal information about the world.

POMCoP models its world as a partially observable Markov decision process (POMDP), a decision theoretic model in which agents have only indirect access, through their observations, to the true state of the world. This naturally models collaboration with humans, since their intentions can never be directly observed in the game state and must be inferred through interaction and observation.

POMDPs are challenging models to work with, since, in general, finding a controller to solve them optimally for fixed time horizons is PSPACE-complete or undecidable in the infinite horizon case (Madani, Hanks, and Condon 1999). This intractability stems from the dual problems of belief space being continuous and planning based on search methods having to contend with a tree of action-observation histories that grows exponentially with the planning horizon.

To address these challenges, POMCoP plans using partially observable Monte-Carlo planning (POMCP) (Silver and Veness 2010). POMCP is based on online tree search techniques that perform well in domains with large state spaces and high branching factors, such as Go. The scalability of its tree search allows it to cope well with large state spaces that would be intractable for POMDP techniques using full-width solution algorithms, such as value iteration (Kaelbling, Littman, and Cassandra 1995), and is competitive with offline point-based POMDP solvers such as SARSOP, which can find approximately optimal plans for domains with thousands of states (Kurniawati, Hsu, and Lee

2008).

POMCP requires a black-box simulator of the POMDP that it is solving. This paper demonstrates through a running example how a multi-agent Markov decision process model, combined with a range of models of human behavior, can be used to construct such a simulator. These elements together with a POMCP solver constitute the POMCoP framework. After discussing each element we present results on the performance of a POMCoP controller on a collaborative game, comparing it to other MDP-based planning approaches and to planners with restricted communication.

Related Work

An effective sidekick NPC should understand a human's goals in order to help achieve them. Inferring human goals from observing game actions has been a topic of growing interest in game AI research. Researchers have recently proposed methods including Markov Logic Networks (Ha et al. 2011), Bayesian Programs (Synnaeve and Bessire 2011), and inverse planning using MDP solutions and Bayes filters (Ngyuen et al. 2011). The latter approach has also been proposed by cognitive scientists as a model for plan recognition in humans (Ullman et al. 2010). POMCoP goes further by planning actions that help a sidekick to actively gain more information about their human collaborator's intentions.

The CAPIR framework for producing NPC sidekick controllers is closely related to POMCoP (Ngyuen et al. 2011). CAPIR decomposes a game into MDPs that model the subtasks that a human may be trying to accomplish within the game world and uses the value iteration algorithm to compute the expected cumulative reward of states within those MDPs. It then estimates which subtask the human is working on using a Bayes filter derived from assigning likelihoods to actions in proportion to these expected rewards, along with a probabilistic model of subtask switching. It selects its actions according to the QMDP rule, which estimates the expected values of actions based on its current belief, the expected rewards computed, and an assumption that all uncertainty will be gone on the following time step (Littman, Cassandra, and Kaelbling 1995). However, because of this assumption, and unlike POMCoP, the CAPIR will never select an action to gain more information.

Fern et al. proposed a similar framework which lacked the subtask switching model and used approximate solution methods to scale beyond problems for which value iteration was intractable (Fern et al. 2007). This approach has also been demonstrated in a practical development setting in the game Dearth, developed by Singapore-MIT GAMBIT Game Lab (http://gambit.mit.edu/loadgame/dearth.php).

In their work on ad hoc teamwork, Barrett, Stone, and Kraus proposed an agent architecture that used a fixed set of models of teammate behaviors to generate action likelihoods for a range of possible teammate types. These were then used in a Bayes filter to allow an agent to infer its teammate's type from observed behavior (Barrett, Stone, and Kraus 2011). The agent then took the best response move for that teammate type based on expected rewards computed either using value iteration or UCT search on the tree of state-action histories (Kocsis and Szepsvari 2006). POMCoP uses a similar strategy, including a range of possible human models as a part of its problem formulation and using tree search techniques for planning, but uses observations rather than states in its histories, which allows it to account for unobserved state variables in its plans.

Broz, Nourbaksh, and Simmons presented a POMDP framework for planning in socially situated tasks that formulated the transition function of their POMDP representation as a composite of environment and agent dynamics in a similar manner to POMCoP and additionally tuned the parameters of their human models using recorded human data (Broz, Nourbakhsh, and Simmons 2011). They demonstrated their approach on a driving task that was not strictly cooperative, in contrast to POMCoP, but which discouraged overt adversarial behavior.

The Cops and Robbers Game

To motivate POMCoP for building NPC sidekicks we introduce the Cops and Robbers game. In Cops and Robbers, the player and their NPC sidekick are cops, chasing robbers who are fleeing from them through the corridors of a building full of twisting passages and one-way doors. The object of the game is to catch any one of the robbers as quickly as possible, but it takes two cops to overpower a robber, so the player and their sidekick must coordinate to corner a robber and catch them. The presence of one-way doors means players have to be careful about the routes they take or they could find themselves wasting a lot of time.

The play field, shown in Figure 1, is top down, with the positions of all of the cops and robbers fully visible. A key strategy for the game is to cut off robbers' escape routes so that the two cops can trap them. To help coordinate this, the sidekick is allowed to ask the player which robber they should aim for, but doing so spends a turn for the sidekick to ask and for the human to respond.

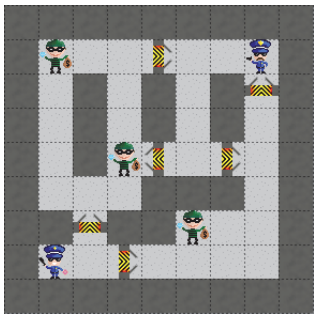


Figure 1: The Cops and Robbers play field. The human is in the top right, the sidekick is in the bottom left. Chevrons indicate the direction of one-way doors.

From the perspective of the sidekick, the game state is fully observable, with the exception of the human's intentions which it cannot directly observe. The player and their sidekick take turns moving, with the robbers moving after the sidekick in the direction that takes them furthest from the nearest cop, breaking ties randomly. If the player and the sidekick catch a robber within 100 turns they win, scoring one point for each remaining turn, if not they lose.

POMCoP requires a black-box simulator of Cops and Robbers played from the perspective of the sidekick in order to use the POMCP algorithm for planning. This simulator must take an action from the sidekick and a state of the world, including any unobserved state variables, and produce a successor state, reward signal, and an observation generated from that successor state.

We will describe the construction of this simulator in two steps. First we will formulate Cops and Robbers as a multi-agent Markov decision process (MAMDP), which models the general dynamics of the game. We will then describe how to formulate Cops and Robbers as a single-agent POMDP, which models the game played from the perspective of the sidekick playing with a human partner whose intentions it cannot directly observe, and over whose actions it has no direct control. The POMDP will make use of human models, described later, to produce actions for the human to play in the previously constructed MAMDP. Figure 2 shows the structure of the simulator.

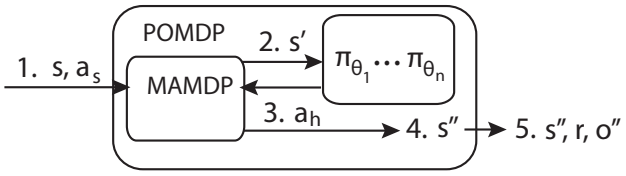


Figure 2: The structure of the black-box simulator. 1. The simulator is passed a state s and sidekick action a_s. 2. The MAMDP dynamics update the state to $s' = \mathcal{T}(s, a_s)$ 3. The human model π_θ for human type $\boldsymbol{\theta}$ in s' selects the human action $a_h = \pi_\theta(s')$. 4. The MAMDP dynamics update the state to $s'' = \mathcal{T}(s', a_h)$. 5. The POMDP observation dynamics and reward signal produce o'' and r, returning them with the successor state s''.

Cops and Robbers as a Multi-Agent MDP

The first step in constructing a simulator for POMCoP is to formulate the game as a MAMDP. In the specific case of Cops and Robbers this formulation is slightly awkward because the dynamics are largely deterministic, aside from some stochasticity in the movement of the robbers. However, performing this formulation will demonstrate the general POMCoP approach and also motivate some of the elements of the POMDP that we will soon be formulating.

The Cops and Robbers MAMDP consists of a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, T \rangle$, in which

- $\mathcal{S}$ is the set of game states. A state $s \in \mathcal{S}$ consists of the locations of each of the cops and robbers and the turn number.
- An action $a \in \mathcal{A}$ is a movement in one of the cardinal directions, a pass, or communication. The sidekick can ask for the human's target and the human can reply. For clarity we will use a_h and a_s to refer to human and sidekick actions respectively.
- The transition function $\mathcal{T}(s, a, s') = \Pr(s'|s, a)$ is the probability that performing a in state s will lead to state s' on the next turn. In Cops and Robbers the human and sidekick move deterministically. Robbers move to the furthest space away from any cop within 3 spaces, breaking ties uniformly at random. Locations containing one-way doors can only be entered from the direction in which the door opens. As mentioned above, the human and the sidekick take alternating turns.
- $\mathcal{R}(s, a, s')$ is the immediate reward received by both players after transitioning to s' from s by performing a. In Cops and Robbers all moves receive 0 reward, except if they result in both cops standing on a robber, when the reward is $100 - t$, where t is the number of turns that have been taken in the game.
- T is the horizon, i.e. maximum number of turns in the game, in this case 100.

A plan for acting in an MDP is a policy, π, which selects an action conditioned on the current state. The value function for π is $V^\pi(s_n) = \mathbb{E}\left[\sum_{t=n}^{T} \mathcal{R}(s_t, \pi(s_t), s_{t+1})\right]$, which gives the expected cumulative reward for starting in s_n and acting according to π. Given a starting state s_0, the goal of planning in MDPs is to find the optimal policy $\pi^* = \underset{\pi}{\operatorname{argmax}}\, V^\pi(s_0)$. This can be found exactly using the value iteration or policy iteration algorithms, but both algorithms scale exponentially with the number of state variables in the MDP.

The Sidekick's Perspective: Cops and Robbers as a Single Agent POMDP

The Cops and Robbers MAMDP describes a two player game, but when playing with a human it becomes a single-player game from the sidekick's perspective, since it has no direct control over the player's actions. A POMDP is a generalization of an MDP to partially observable domains, where rather than directly observing the state of the world an agent instead receives observations through its sensors, from which it must then infer the underlying state of the world. This reorientation can be formalized as a single-agent POMDP described by the tuple $\langle \boldsymbol{\theta}, \Pi_{\boldsymbol{\theta}}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, O, T, b_0 \rangle$, where

- $\boldsymbol{\theta}$ is the set of possible human types. In Cops and Robbers a type $\theta \in \boldsymbol{\theta}$ represents an intent to pursue a particular robber, with one type for each robber.
- $\Pi_{\boldsymbol{\theta}}$ is a set of stochastic policies representing the behaviors for each human type. A policy $\pi_\theta \in \Pi_{\boldsymbol{\theta}}$ is a non-deterministic decision rule that selects an action a_h given that the game is in state s. In Cops and Robbers each π_θ corresponds to a plan for pursuing robbers for humans of type θ. We will discuss methods for generating these policies later.
- A state $s \in \mathcal{S}$ comprises not just the location of each of the cops and robbers and the turn number, but also unobservable type of the human, θ. In a POMDP the state is not directly observable by the agents within it. Instead they receive observations conditioned on the state.
- $\mathcal{A}$ in this case is the sidekick actions from the MAMDP.

- $\mathcal{T}$ for Cops and Robbers is formed from the underlying MAMDP's transition function by chaining it with the human policy, i.e. $\mathcal{T}(s, a, s'') = \sum_{s'} \Pr(s'|s, a) \Pr(s''|s', \pi_\theta(s'))$.
- $\mathcal{R}$ in Cops and Robbers is $50-t$ in a terminal state, as each time step in the POMDP is two turns in the MAMDP.
- $\mathcal{O}$ is the set of observations that the sidekick can receive from the game. In Cops and Robbers an observation is a vector of the actions taken by the cops and the robbers since the sidekick's last action. These actions are movements in each of the cardinal directions, passing, or communicating. Note that although Robbers can select actions non-deterministically in Cops and Robbers, the outcomes of their actions are always deterministic, so it is possible to reconstruct the current positions of each entity in the game from their starting locations and a history of observations.
- $O(s, s', o) = \Pr(o|s, s')$ is the probability that the sidekick will receive observation o given that the previous state of the game was s and the current state is s'. For Cops and Robbers O is deterministic, with $\Pr(o|s, s') = 1$ iff o is the vector of actions that when taken in s results in s' according to the game's deterministic dynamics.
- The horizon T is 50, since the number of turns is halved as discussed previously.
- The initial belief $b_0 = \Pr(s_0)$ is a prior distribution over the possible starting states of the game. In Cops and Robbers the game begins with each of the cops and robbers in known starting locations, so b_0 is a uniform distribution over the states with this initial configuration, i.e. with only θ varying across the states in the distribution.

In planning using the simulator, POMCoP will using of histories and beliefs. A history in a POMDP is a sequence of actions and observations $h_t = \langle a_1, o_1, ..., a_t, o_t \rangle$. A belief b_t at time t is the distribution $b_t(s) = \Pr(s|h_t)$. This represents the sidekick's belief about the true world state, in particular the human's intent θ, given the actions taken and observations received by the sidekick up to that point.

The goal of planning in a POMDP is to find a policy π that selects actions based on the agent's current belief to maximize the value function $V^\pi(b_0) = \mathbb{E}\left[\sum_{t=0}^{T} \mathcal{R}(s_t, \pi(b_t), s_{t+1})\right]$.

POMCP

Having formulated our multiplayer game as a single-agent POMDP we can now build a sidekick controller by solving the POMDP. Since finding an optimal policy for a POMDP the size of Cops and Robbers using exact methods is intractable, we instead approximate optimal behavior by using the POMCP algorithm to select actions (Silver and Veness 2010). POMCP performs UCT search on the tree of action-observation histories rooted at the sidekick's current history, generated via Monte-Carlo simulation using the simulator. The rewards accrued during a given simulation are used to bias future searches towards promising parts of the search space. Aside from the simulator, it requires specifying a tree policy, used for trading off exploration and exploitation in its search and a rollout policy used for exploratory search. After taking actions in the game, POMCP uses a particle filter to updates its belief over the hidden states of the world based on the observations it receives.

POMCP is an online algorithm which is guaranteed to converge to the optimal policy given sufficient simulations. In practice, there is a tradeoff between the runtime cost of running more simulations and the quality of the actions produced. The horizon of the optimal plan also affects the number of simulations required to produce high quality actions.

Human Models

Constructing POMCoP's black-box simulator requires the human types $\boldsymbol{\theta}$ and policies $\Pi_{\boldsymbol{\theta}}$, which serve as human models. As previously discussed, the POMDP's transition function depends on the human models, because it is formed by chaining together the MAMDP's transition model with the human model's actions. In principle a human model could be any arbitrary policy, and in the worst case we might not know what a reasonable human policy would be.

For Cops and Robbers we implemented two human models, which represent a human who has some target robber in mind and moves by the shortest path towards their target, relying on the sidekick to react to their actions appropriately. We implemented this behavior using both A* search and simple Manhattan distance minimization.

In general, the problem of constructing a human model is challenging, particularly for domains where human strategies are unpredictable. One possible option would be to learn a model from observed human data, either online (Barrett, Stone, and Kraus 2011) or offline (Tastan and Sukthankar 2012; Orkin 2008; Broz, Nourbakhsh, and Simmons 2011). Alternatively a human could be modeled as a noisy optimal solver for an MDP or POMDP formulation of the game, assuming such policies could could be tractably found or approximated. We are currently investigating methods for generating human models automatically by using statistics from the POMCP search to estimate an optimal policy, parameterized only by a reward signal.

Evaluation

To evaluate our framework we built a POMCoP controller for the Cops and Robbers game and tested its performance on a range of different maps, in which to perform well it is critical for a sidekick to reason about how its actions will affect its own beliefs. We compared its performance against a state space planner inspired by CAPIR. For evaluation purposes we paired each planner with a simulated human that took the myopic strategy of choosing a robber to pursue and using A* to select actions with 10% noise. The simulated human also answered questions about its intended target with 10% noise, and never changed targets.

We implemented two versions of the POMCoP planner; one which was allowed to take communication actions and one which was not. In Cops and Robbers a player's response to a communication actions can disambiguate the player's goal completely, meaning that communication actions act as

a noisy oracle for the player's goal. Disallowing communication actions allowed us to more clearly see the effect of hedging actions, such as strategically passing to wait for the human's actions to reveal more about their goal.

Our communicative POMCoP planner used a simulator constructed as described previously. We tested the communicative planner with two different human models types, a set of A* models and a set of Manhattan distance minimizing models, each parameterized by the robber they were chasing. The models responded truthfully to sidekick queries and took greedy actions with respect to their heuristics 70% of the time and responded noisily or chose actions uniformly at random 30% of the time. The non-communicative planner also used a set of A* models.

To give real-time responses for the sidekick, our implementation of POMCP used 50,000 simulations for each sidekick turn, resulting in actions being selected by POMCoP in under 2 seconds of wall clock time. One possible direction for improving the performance of POMCoP would be to split the POMCP searches into an ensemble run across multiple cores using root parallelization, which has been shown to be an effective parallelization strategy for UCT search across a range of domains (Fern and Lewis 2011). For POMCP's tree policy we used the standard UCB-1 rule with an empirically chosen exploration constant of $c = 100$. For its rollout policy we used a policy that selected randomly from the sidekick's legal actions, with a bias towards actions resulting in a decrease in the Manhattan distance between the sidekick and the target for the human model in the state.

As a point of comparison we built a QMDP planner, inspired by CAPIR and other MDP value iteration methods, that planned in state space and used a Bayes filter to track its belief. Our QMDP planner decomposed the game into MAMDPs modeling the subtasks of catching each robber individually then used offline value iteration to solve for the joint policies of these subtask MAMDPs. Using the human halves of the joint policies it then formed single-player MDPs from each subtask MAMDP by chaining the human's half of the joint policy, with 10% noise introduced, to the MDP's transition function. It then used value iteration again to find an optimal policy for interacting with the resulting close-to-optimal, but noisy, human model. At runtime the planner maintained a belief about the human's goal using a Bayes filter on action likelihoods derived from the value function of the human's half of the joint policies and used the QMDP rule to select the best action as described for other value iteration methods (Fern et al. 2007; Ngyuen et al. 2011). Unlike CAPIR this planner did not attempt to model goal switching.

Figure 3 shows the maps used for our evaluation. Each map emphasized the importance of planning with respect to beliefs by requiring the human to initially take actions that were ambiguous with respect to their goal, whereas the sidekick's choice of early actions could allow the human and the sidekick to catch the human's target much sooner if it could disambiguate the goal. Maps *b-e* included one-way doors that made it hard for the sidekick to recover from pursuing the wrong target. Intuitively, the best way to ensure a good outcome in each map was to either ask the human for their target early, or to stall and wait for the human to perform an action that disambiguated their target before committing to chase any given robber.

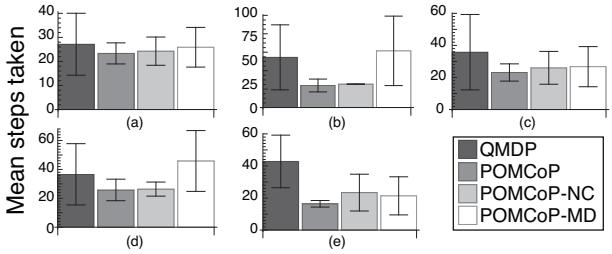
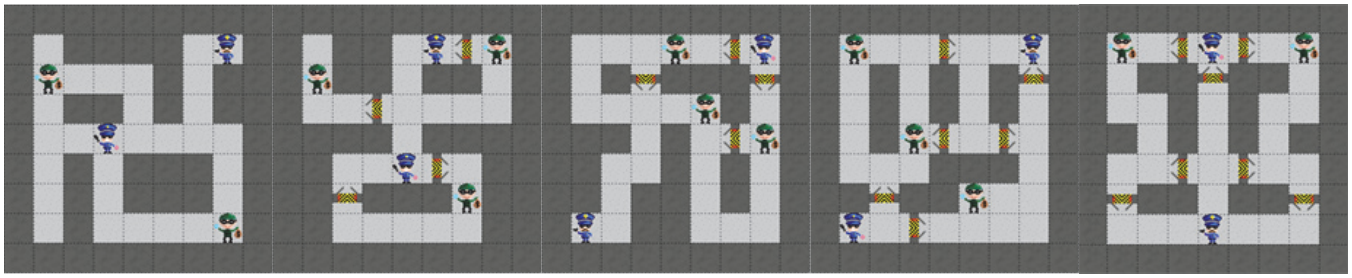


Figure 4: The mean number of steps taken to complete each map over 100 runs for QMDP, POMCoP and non-communicative (POMCoP-NC) with A* models, and POMCoP with Manhattan distance models (POMCoP-MD). Error bars indicate the standard error of the sample mean.

The results in Figure 4 show that, for the planners with A* models, POMCoP significantly out-performed the QMDP planner in the mean number of steps taken to win the game, averaged across 100 runs. The high variance in the number of steps taken by the QMDP planner demonstrates that the QMDP rule can commit to actions that have a high expected reward, weighted by the sidekicks current belief, and yet are suboptimal. This is particularly pronounced when there are three possible targets, the sidekick is completely uncertain about the human's intended target, and the best action for pursuing 2 of the targets is deeply suboptimal for the remaining target, as in map *b*. In this case QMDP typically selects the possibly suboptimal action that is best for the majority of targets, rather than waiting for the human's actions to disambiguate its intentions. By contrast, POMCoP will either ask the human for clarification, or plan to wait and see what the human intends before committing to a potentially costly action. On maps *b* and *d* especially, POMCoP with the Manhattan distance model was on par with, or outperformed by, QMDP. The maze layout of these maps made Manhattan distance minimization a particularly bad approximation for the simulated human behavior, which led to poor performance in updating beliefs over human targets, demonstrating the importance of a good human model.

Figure 4 also shows that the non-communicative planner performed only marginally worse than the planner that had communication actions available, with the exception of map *e*, in which the communicative agent benefited from being able to get a head start on the relatively long travel distances from asking for the target early. This demonstrates that often simply waiting and performing actions that hedge against the sidekick's uncertainty about the humans intentions can be almost as effective as directly asking the human about their intentions, and may also be less intrusive.

On maps *b-e* the POMCoP planner with communication actions queried the human for its intention within its first 4 moves on every run. On map *a* it either queried the human in 34% of runs within its first 4 moves or not at all. In any given run the POMCoP planner queried the human at most once, which reflects the fact that the humans response was



(a) (b) (c) (d) (e)

Figure 3: The maps used in evaluating the three planners.

almost completely guaranteed to disambiguate their intentions, making further communication unhelpful.

Conclusion

Planning in belief space allows NPC sidekicks to reason about how their actions can affect their uncertainty about human intentions. In particular this allows them to reason about how communication with humans can reveal a human's goals and about how to act in a way that hedges against their uncertainty. This kind of reasoning is particularly helpful in games when prematurely committing to actions that help a human to pursue a goal that they are not actually pursuing can be costly to undo.

We presented the POMCoP framework for developing sidekicks that plan in belief space. POMCoP uses a POMDP model of a game's structure and a set of human models representing different possible human intentions to build a simulator for belief space planning using POMCP.

We demonstrated constructing a POMCoP sidekick for the Cops and Robbers game and showed that it outperforms QMDP-based state space planning. This performance gain comes from explicitly reasoning about how asking the human about their intentions can reveal the human's goal, and about how taking stalling actions to wait for the human to disambiguate their goal, allowing the sidekick and human to catch robbers faster than prematurely committing to a course of action as a state space planner would do.

Acknowledgements

We would like to thank the Singapore-MIT GAMBIT Game Lab staff for their helpful feedback. This work was supported in part by MDA GAMBIT grant R-252-000-398-490.

References

Barrett, S.; Stone, P.; and Kraus, S. 2011. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *AAMAS*.

Broz, F.; Nourbakhsh, I.; and Simmons, R. 2011. Designing POMDP models of socially situated tasks. In *IEEE Robot and Human Interactive Communication*.

Fern, A., and Lewis, P. 2011. Ensemble Monte-Carlo planning: An empirical study. In *ICAPS*.

Fern, A.; Natarajan, S.; Judah, K.; and Tadepalli, P. 2007. A decision theoretic model of assistance. *IJCAI*.

Ha, E. Y.; Rowe, J. P.; Mott, B. W.; and Lester, J. C. 2011. Goal recognition with Markov logic networks for player-adaptive games. In *AIIDE*.

Isla, D. 2005. Handling complexity in the Halo 2 AI. *GDC*.

Kaelbling, L.; Littman, M.; and Cassandra, A. 1995. Planning and acting in partially observable stochastic domains. *AI* 101.

Kocsis, L., and Szepsvari, C. 2006. Bandit based monte-carlo planning. In *NIPS*.

Kurniawati, H.; Hsu, D.; and Lee, W. S. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. *Robotics: Science and Systems*.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning policies for partially observable environments: Scaling up. In *ICML*.

Madani, O.; Hanks, S.; and Condon, A. 1999. On the undecidability of probabilistic planning and infinite-horizon partially observable decision problems. In *AI*.

Ngyuen, 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*.

Orkin, J. 2004. Symbolic representation of game world state: Towards real-time planning in games. In *AAAI Workshop on Challenges in Game AI*.

Orkin, J. 2008. Automatic learning and generation of social behavior from collective human gameplay. In *AAMAS*.

Silver, D., and Veness, J. 2010. Monte-Carlo planning in large POMDPs. In *NIPS*.

Straatman, R.; Verweij, T.; and Champandard, A. 2009. Killzone 2 multiplayer bots. In *Paris Game AI Conference*.

Synnaeve, G., and Bessire, P. 2011. A Bayesian model for plan recognition in RTS games applied to StarCraft. In *AIIDE*.

Tastan, B., and Sukthankar, G. 2012. Learning policies for first person shooter games using inverse reinforcement learning. In *AIIDE*.

Ullman, T. D.; Baker, C. L.; Macindoe, O.; Evans, O.; Goodman, N. D.; and Tenenbaum, J. B. 2010. Help or hinder: Bayesian models of social goal inference. In *NIPS*.