Applying MDP Approaches for Estimating Outcome of Interaction in Collaborative Human-Computer Settings

Applying MDP Approaches For Estimating Outcome of Interaction in Collaborative Human-Computer Settings

Ece Kamar
School of Engineering and Applied Sciences
Harvard University
Cambridge MA 02138 USA
kamar@eecs.harvard.edu

Barbara J. Grosz
School of Engineering and Applied Sciences
Harvard University
Cambridge MA 02138 USA
grosz@eecs.harvard.edu

ABSTRACT

This paper investigates the problem of determining when a computer agent should interrupt a person with whom it is working collaboratively as part of a distributed, multi-agent team, which is operating in environments in which conditions may be rapidly changing, actions occur at a fast pace, and decisions must be made within tightly constrained time frames. An interruption would enable the agent to obtain information useful for performing its role in the team task, but the person will incur a cost in responding. The paper presents a formalization of interruptions as multi-agent decision making. It defines a novel, efficient approximation method that decouples the multi-agent decision model into separate MDPs, thereby overcoming the complexity of finding optimal solutions of the Dec-POMDP model. For single-shot situations, the separate outcomes can be combined to give an exact value for the interruption. In more general settings, the closeness of the approximation to the optimal solution depends on the structure of the problem. The paper describes domain specific heuristic functions that improve the efficiency of the approximation further for a specific application.

1. INTRODUCTION

Effective collaborations require a range of communications among team members. This paper investigates one particular, important class of communication, interruptions. Interruptions are typically required for collaborative efforts when one agent on a team has information that will assist another agent in performing some subtask or will provide information that will enable the other agent to improve the performance of the group activity. The need to get information from another agent arises in mixed human-computer teams as well as in homogeneous computer-agent environments. For example, a (human) driver may see changes in weather conditions that affect route selection as they occur, while an automated navigation system without sensors does not. This information may be important for the navigation system in choosing the best route. However, interruptions are by their very nature disruptive. The extent to which they disrupt depends on what the person or agent being interrupted is doing when the interruption occurs. Thus, it is crucial to time interruptions appropriately. To do so requires accurately estimating the costs and benefits associated with an interruption [10, 3].

This paper reports research that contributes to methods for timing and managing interruptions appropriately in environments such as disaster rescue situations in which agents are distributed, conditions may be rapidly changing, actions occur at a fast pace, and decisions must be made within tightly constrained time frames. In contrast to work on timing interruptions in office or computer workstation settings, the approach to interruptions in such "fast-paced environments" must take into account the high levels of uncertainty and the limited resources (especially time) of the participants. In such settings, an accurate estimation of the usefulness of an interruption is especially crucial. Otherwise, the party being interrupted may refuse to respond, treating the interruption as an unnecessary disturbance so that it has no positive benefit.

The particular problem on which this paper focuses is the evaluation of the costs and benefits associated with an interruption. Prior work on interruption management [9, 4] has focused on the cost of interrupting a person either initially or repeatedly. Prior work on adjustable autonomy [15] has focused on determining when a system does not know enough and should transfer control to a person (i.e., the joint activity will benefit from its doing so). The model described in this paper takes into account the costs and benefits to two parties working together; it considers simultaneously the utility for both person and agent in a human-computer collaboration. Either costs or benefits or both may accrue to the person or for the system, and the overall usefulness of an interruption requires appropriately combining information from these two different perspectives.

The paper presents a decision making model for interruption management based on the Decentralized MDP (Dec-MDP) formalism, but addressing the problem of the complexity of Dec-MDP models which are NEXP-complete [1]. We propose a novel approximate model for interruption management that decouples multi-agent decision making problems into a pair of single agent problems. This approximation method replaces the Dec-POMDP with an MDP for one participant and a POMDP for the other participant, thus reflecting the different information available to the two parties. The elimination of the multi-agent decision mak-

MSDM 2007 May 15, 2007, Honolulu, Hawai'i, USA.

ing overhead reduces the complexity of the solution from doubly exponential to exponential. A further reduction in complexity to close to polynomial is shown to be possible by combining POMDP solution techniques with domain specific heuristic functions.

The research we describe uses Colored Trails (CT), a game infrastructure that provides a clear analogue to goals, tasks, resources and the interactions among them, but abstracts from application domain specifics [8]. CT's abstraction away from complicated underlying domains, enables investigations to focus on decision-making strategies, rather than specifying and reasoning about individual domain complexities. Prior uses of CT as a research test-bed for analyzing decision-making in multi-agent contexts [8, 5] have shown it engages people in playing both with other people and with computer agents. We designed a particular CT-game environment which provides a conceptually simple analogue of interruption-scenarios in fast-paced domains with uncertainties and partial information. Though abstract, the game remains challenging and interesting for people to play.

This paper demonstrates the usefulness of multi-agent decision making models for estimating interruption outcomes in the context of collaborative activities being carried out in environments characterized by uncertainty and partial information. It contributes to research on interruption management by enabling the calculation of the expected combined (person and agent) "theoretical" value of an interruption, which provides an upper bound on the actual value of the interruption. In the calculation of the theoretical interruption outcome, the player corresponding to the person is assumed to be fully rational, computationally unbounded and to act like an agent in a two agent collaboration. This calculation is intended to be used as a baseline with which to compare actual human decision-making in empirical studies of human-computer play of the CT interruption game. Such comparisons provide a basis for determining the kinds of biases people exhibit in such settings (i.e., the extent to which their decisions deviate from that dictated by the optimal calculation). An understanding of these biases will provide the basis for the design of systems able to predict more accurately a person's tendency to accept or reject an interruption in a particular context. The paper also contributes to multi-agent decision making research by defining a novel method for decomposing computationally expensive DecPOMDPs into more tractable individual MDPs and POMDPs. This approximation method can be applied to any DecPOMDP that has a joint reward function decomposable to individual reward functions with nearly-decomposable transition function and action sets.

The next section of this paper introduces the CT-game environment which we use to investigate the problem of timing interruptions. Section 3 describes a decentralized MDP model for the interruption problem, which forms the basis for defining the approximation method. Section 4 describes the approximation method, which decouples multi-agent interruption decision problem to individual models. The paper concludes with a survey of related work and discussion.

2. THE CT INTERRUPTION GAME

To investigate the interruption management problem, we developed a team game of two players in which one of the players has information that will help the other player to perform a task. In this game, the player that lacks information has incentive to ask for it and the player that has the information has a reason to provide it, because interruptions are beneficial for the team's work. The game is a board game, developed using the Colored Trails (CT) infrastructure, played by a person and a computer agent.

The CT game board is divided into cells, and both players and their goals are located on cells of the game board. The players move one square at a time towards their goal squares. Players reach their goals when they move to a square on which their goals are located. CT's scoring function defines the overall objective of the players. It may be set so that, the players are purely self-interested (if players' points only depend on their own performance), or have a social good component. The score is determined by the number of goals reached by the players but other factors can be included into the scoring function.

The CT "interruption game" designed for this research involves two players, one controlled by a computer agent and the other by a person, and individual goals for each of the players. The dynamically changing nature of the real world is modeled by having the goals move stochastically with a probability determined by a Gaussian function with the center at the current position of the player's goal (see Figure 1). The Gaussian function restricts the goal movement; the goal cannot move closer to the player.

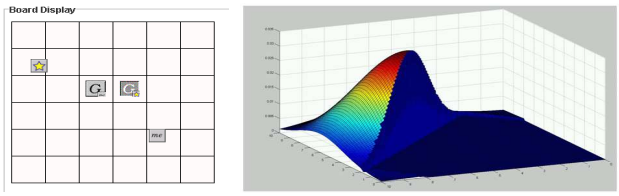

Figure 1: Left: Screenshot; *me* is the person player, star is the agent player, G_{me} is the person's goal, G_{star} is the agent's goal. Right: Corresponding movement function for G_{me}.

As part of the game setting, the person has complete information whereas the computer agent is unaware of the current position of its goal except in the initial position. The agent's information about its goal diminishes over time, and its success in reaching its goal depends on getting information from the person. Thus, interruptions initiated by the agent are critical determinants of the result of the game.

In the CT interruption game, the cost of interruption is set to losing the opportunity to move for one game turn. If an interruption is established, neither player is allowed to move for one game round. Analogously to interruption decisions, at the beginning of each turn, the agent decides whether to continue to move or to interrupt the person to get information but with the cost of losing the opportunity to move. The person is free to accept or reject an interruption request. If the person rejects the interruption request, the players continue their individual play, otherwise an interruption is established, the agent receives the current position of its goal and both players remain in their current locations. Thus to make an interruption decision, the agent should consider the tradeoff between making progress on its current task and suspending the current task to ask for or provide information.

When a goal is reached by a player, this player and her/its

goal are randomly relocated on the board starting another round of play. The game continues until a fixed number of turns are played. The quicker the players reach their goals, the more chance they have to reach additional goals and increase their score.

The CT interruption game gives players an incentive to collaborate by defining a common scoring function S, given below, where s is the points awarded for getting to a goal, h_k is number of turns it takes for the player to get to the goal k, and the sum is over all goals that have been reached by player i where A indicates agent, P indicates person.

$$S = S_A + S_P \text{ and } S_i = \sum_k (s - h_k)$$

The objective of the players is to maximize the shared scoring function S. In this setting, the agent has an incentive to request interruptions from the person for learning its goal position. The person has an incentive to accept the interruption requests as her success depends on the success of the agent since they share a common scoring function.

3. DEC-POMDP FORMALIZATION OF THE INTERRUPTION PROBLEM

The overarching goal of this research is to provide an accurate estimate of the value of an interruption at a particular time in a human-computer collaborative activity. Ideally, the agent would initiate an interaction with the person only when the interaction has positive value for the person. However, in many situations a person's estimate of the value of an interruption may not match a fully rational computational estimate. Such mismatches may lead the person to refuse to respond to an interruption, thus further decreasing task performance. We are addressing this problem in two phases. First, we obtain a baseline of the "theoretical" value of an interruption, assuming the person to be fully rational and without any computational-resource limitations. We then use this baseline to enable empirical investigations of actual human behavior in mixed human-computer collaborative settings, and adapt the computer agent model based on these empirical results. This paper addresses the first phase, and this section presents a computational model that accurately captures the theoretical baseline of interruption values for the CT interruption game. Although we are calculating the fully rational, theoretical baseline, for clarity of presentation we continue to refer to the agent or player with complete information as "the person player".

Although an interruption is an action taken by an agent with partial information, this action affects both agents' states, and consequently both reward functions. Thus, to compute the value of an interruption, its effect on both the person's (agent with full information) and the computer agent's (agent with partial information) individual performance must be taken into account. It is the aggregate of these two effects that determines the interruption's value. It is notable that from an individual perspective, the performance of the individual being interrupted (i.e., the person) always decreases, while the performance of the individual making the interruption (i.e., the computer agent) may either increase or decrease. The challenge for the interrupter is to accurately identify situations in which the overall expected benefit to the team of the interruption is positive.

This section describes a Dec-MDP model for interruption decision making. We use a Dec-MDP model to derive policies that maximize the joint reward function, because of the uncertainty in the way state changes and the fact that the interruption action affects both players. Finding optimal solutions to the Dec-MDP problems has been proven to be NEXP-complete [1], and the complexity of finding an optimal solution to the Dec-MDP described in this section is doubly exponential in the number of players and the time horizon H. Although the optimal solution is infeasible even for moderately sized problems, this Dec-MDP provides the right theoretical baseline against which to compare the approximate model we define in Section 4.

The Dec-MDP model for interruption decision making in the CT interruption game uses the following terms: B is the set of board positions; $|B|$ is the size of the game board; $p_P, g_P, p_A, g_A \in B$, are the positions of the person, person's goal, the agent, and the agent's goal respectively; b is the belief state of the agent about its goal position; for $c \in B$, $b(c)$ is the probability of agent's goal being on square c; H is the horizon of the game; $a_A \in A_A$ is an action for the agent, where $A_A = \{up, left, right, down, interrupt\}$ is the set of actions for the agent; $a_P \in A_P$ is an action for the person, where $A_P = \{up, left, right, down, accept(interruption)\}$ is the set of actions for the person; s is the reward associated with catching the goal; S^h is the state at time h; $P_M(g', p, g)$ is the probability of a goal move from position g to position g' when the player is in position p. Dec-MDP also requires a state estimator function for updating the belief state. The state estimator (SE) function updates the belief state b of the agent to b' given agent position p, where $\forall c' \in B$, $b'(c') = \sum_{c \in B} b(c) \times P_M(c', p, c)$.

Our Dec-POMDP is modeled by the tuple $< I, S, A_i, T, \Omega_i, O, R, H >$ where I is the finite set of players, $I =< Person, Agent >$; S is set of world states, represented as a cross product of p_P, g_P, p_A, g_A and b; A is the set of actions, $A =< A_P, A_A >$; Ω is the set of observations where $\Omega = \emptyset$; O, the observation probability function is undefined; $T : S \times A \times S \longrightarrow [0, 1]$ is the state transition function. State transition probability is basically the multiplication of move probabilities of player's goals to the squares given in the next state description, except where the agent or the player or both reach their goals. In those cases, uniform distribution probability is used instead of move probability for the player that catches her/its goal.
The reward function $R : S \times A \longrightarrow R$ is defined as,

$$R(S^h, A) = \begin{cases} (s - h) + b(p_A) \times (s - h) & \text{if } p_P + a_P = g_P \\ b(p_A) \times (s - h) & \text{otherwise} \end{cases}$$

4. APPROXIMATING THE OPTIMAL POLICY

In this section, we describe an efficient way of approximating the joint decision making for the interruption game with individual MDP solutions. The approximation decouples nearly-decomposable Dec-MDP models into individual decision-making constituents in such a way that the combination of the results of the two models gives an accurate estimation of the collaborative outcome. This approximation reduces the complexity of decision making in such models to the complexity of solving the individual constituent MDPs.

We describe the use of this approximation to estimate the outcome of interruption provided individual decision making models in CT interruption game in Section 4.1. Decision making models for the agent and person perspectives are presented in Sections 4.2 and 4.3 respectively. In Section 4.4,

this approximation approach is generalized to any nearly-decomposable decentralized decision making setting.

4.1 Decoupling to Single Agent Decision Making

This section presents an approximate algorithm for determining agent decisions in the CT interruption game. The collaborative value of interrupting the person is approximated by decoupling the Dec-MDP model described in Section 3 into two individual decision models, one for the agent and one for the person perspectives, and then combining the outcome of individual models.

The optimal value for an interruption is estimated by the Expected Outcome of Interruption (EOI) which is the difference between the Expected Utility (EU) of the current state when the interruption is established and the EU when no interruption is established.

$$EOI = EU^{I} - EU^{NI}$$

where EU^{I} indicates the expected utility of interruption, and EU^{NI} indicates the expected utility of no interruption. An interruption is beneficial when EOI is positive.

Having two different players leads to two different perspectives on the interruption outcome; EOI_P is the person's perception of interruption outcome which we will refer to as "person's perspective", EOI_A is the agent's perception of interruption outcome which we call "agent's perspective". With decoupling, EOI is approximated by combining the value of interruption for individual perspectives of the person (P) and the agent (A).

$$EOI = EOI_P + EOI_A$$

The EOI of a single player is the difference in expected values of two possible states, one in which an interruption is established and other in which it is not.

$$EOI_P = EU_P^{I} - EU_P^{NI} \text{ and } EOI_A = EU_A^{I} - EU_A^{NI}$$

The expected values for individual states are provided by the decision making models for the person (P) and the agent (A) which are given in Sections 4.2 and 4.3. A state of the model of the person's perspective is represented by the current person position p, person's goal position g and current turn number h. $P_M(g', p, g)$ is the probability of a goal move from position g to g' with player position p. B is the set of all board positions.

$$EU_P^{NI} = EU_P(p, g, h)$$
$$EU_P^{I} = \textstyle\sum_{g' \in B} MP(g', p, g) \times EU_P(p, g', h+1)$$

The state of the model of the agent's perspective is represented by agent position p, agent's goal position belief state b, and current turn h. After each turn, b is updated to b' using the $StateEstimator(SE)$ function given in Section 3.

$$EU_A^{NI} = EU_A(p, b, h) \text{ and } EU_A^{I} = EU_A(p, b', h+1)$$

For any given world state, our algorithm approximates EOI, the overall expected benefit of interrupting the person. If EOI is estimated to be beneficial, an interruption is established between the person and the agent; otherwise both players follow their individual optimal policy. The resulting policy is a complete description of the agent and person actions. The policy is optimal for players' individual actions, but suboptimal for the timing of interruptions. This approximation reduces the complexity of the multi-agent decision making process to that of two separate single agent decision making processes.

4.2 Individual Decision Making Model for the Person Perspective

The person's perspective is fully observable and is modeled as a Markov Decision Process (MDP). The terms used for the person perspective MDP formalization are as follows: B is the set of board positions; $|B|$ is the size of the game board; $p \in B$, $g \in B$ are the positions of the person and her goal respectively; H is the horizon of the game; $a \in A$ is an action where $A = \{up, left, right, down\}$ is the set of actions; s is the reward associated with reaching the goal; $P_M(g', p, g)$ is the probability of a goal move from position g to position g' when the player is in position p; S^h is the state at time h.

MDP that represents the person perspective is the tuple $< S, A, T, R >$ where S is the set of states, expressed as a cross product of p and g; A is the finite set of allowed actions for the person; $T : S \times A \times S \longrightarrow [0, 1]$, the state transition function is defined as,
$Pr(S^{h+1} = [p', g', h+1] | S^h = [p, g, h], a) = T(S^{h+1}, a, S^h)$

$$T(S^{h+1}, a, S^h) = \begin{cases} P_M(g', p', g) & \text{if } p + a = p' \text{ and } p + a \neq g \\ 1/|B|^2 & \text{if } p + a = g \\ 0 & \text{otherwise} \end{cases}$$

$R : S \times A \longrightarrow R$, the reward function is defined as,

$$R(S^h, a) = \begin{cases} s - h & \text{if } p + a = g \\ 0 & \text{otherwise} \end{cases}$$

The value function for optimal policy calculation is:

$$V^{\Pi}(S^h) = max_a[R(S^h, a) + \sum_{S^{h+1}} T(S^{h+1}, a, S^h) \times V^{\Pi}(S^{h+1})]$$

MDPs can be solved in the number of arithmetic operations polynomial to the size of the state space, number of actions and the number of bits required to represent the transition function and the reward function [11]. We use ExpectiMax solution algorithm, because it is sufficiently efficient, simple to implement and easy to adopt to solve NOMDP models.

V^{Π} is the value function that maximizes the utility of the person's perspective.

$$V^{\Pi}([p, g, h]) = max_a[R([p, g, h], a) + \sum_{c \in B} T([p + a, c, h+1], a, [p, g, h]) \times V^{\Pi}([p + a, c, h+1])]$$

where $S^h = [p, g, h]$. ExpectiMax algorithm constructs a decision tree that computes the value function V^{Π}. Starting from an initial state, ExpectiMax branches into two levels, one branch over the four possible actions, and the other over the $|B|$ possible positions to which the goal can move. The expected outcome is calculated by taking the weighted average of the outcome of each branch with its branching probability. The size of the decision tree is $(|B| \times |A|)^H$ which is exponential in the length of the horizon. With memoization, the number of nodes visited by the complete policy search is bounded by $|B|^2 \times |H|$ and the solution becomes practical.

4.3 Individual Decision Making for the Agent Perspective

Agent decision making is modeled with a No Observation Markov Decision Process (NOMDP) because the interaction component is excluded from the individual model.

The terms used for the agent perspective NOMDP formalization are as follows: B is the set of board positions; $|B|$ is the size of the game board; $p \in B$ is the position of the agent; b is the belief state of the agent about its goal position; for $c \in B$, $b(c)$ is the probability of agent's goal being on square c; H is the horizon of the game; $a \in A$ is an action, where $A = \{up, left, right, down\}$ is the set of actions; s is the reward associated with catching the goal; S^h is the state at time h. The state estimator (SE) function updates the belief state b to b' given agent position p, where $\forall c' \in B$, $b'(c')=\sum_{c \in B} b(c) \times P_M(c', p, c)$.

The NOMDP that represent the agent perspective is the tuple $< S, A, T, R >$ where S is the set of states, expressed as a cross product of p and b, A is the finite set of allowed actions for the agent, $T : S \times A \times S \longrightarrow [0, 1]$ state transition function is defined as,

$Pr(S^{h+1} = [p', b', h+1] | S^h = [p, b, h], a) = T(S^{h+1}, a, S^h)$

$$T(S^{h+1}, a, S^h) = \begin{cases} 1 & \text{if } p + a = p' \text{ and } SE(p', b) = b' \\ 0 & \text{otherwise} \end{cases}$$

$R : S \times A \longrightarrow R$ the reward function is defined as,

$$R(S^h, a) = b(p) \times (s - h)$$

The value function for optimal policy calculation is:

$$V^{\Pi}(s) = max_a R(S^t, a) + \sum_{S^{t+1} \in S} T(S^{t+1}, a, S^t) V^{\Pi}(S^{t+1})$$

The set of initial states is the finite combination of agent positions and goals. An infinite number of state possibilities exist, because the belief distribution is incorporated into the state space, however only small regions of the state space are reachable from the finite set of initial positions. In order to eliminate the unreachable states and only consider those that are reachable, we customize the well-known Expecti-Max algorithm, which is a version of Value Iteration that focuses on reachable states [13].

V^{Π} is the value function that maximizes the utility of the agent's perspective.

$$V^{\Pi}([p, b, h]) = max_a [R([p, b, h], a) + (1 - b(p)) \times V^{\Pi}([p + a, SU(p + a, b), h + 1])]$$

where $S^h = [p, b, h]$. Starting from an initial state, a policy tree is constructed that maximizes the value function V^{Π}. At each level, the tree selects an agent action that maximizes the value function and the agent's belief state is updated by the Special Update (SU) function (See Appendix 2 for the SU function). The branching continues until the horizon is reached. Even with dynamic programming, the complexity of the complete search is exponential in the length of the horizon, $(|A| \times |B|)^H$.

To overcome this complexity, we define a domain specific heuristic to cut down the search space. A basic heuristic for the CT interruption game is selecting actions that take the agent closer to its goal. To have a more efficient policy search algorithm, this heuristic is integrated into the complete search algorithm. While constructing a policy tree, only actions that take the agent closer to its goal are included in the search.

Figure 2 compares the running times of exact and heuristic-approximate NOMDP search algorithms. By pruning moves that are estimated to be unbeneficial, the branching factor becomes either 1 or 2 for each node of the decision tree. For the CT interruption game where $|A| = 4$ and complexity of the exact search is $(4 \times |B|)^H$, the best case complexity of the heuristic-approximate NOMDP search algorithm is $H \times |B|$.

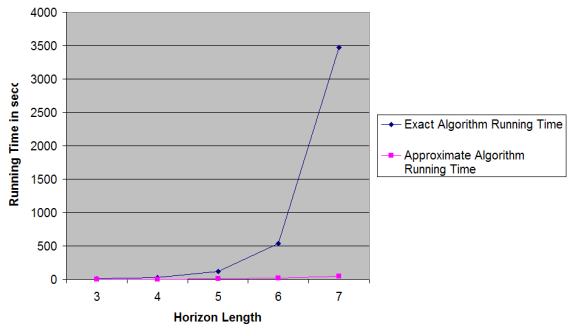

Figure 2: Comparison of exact algorithm and heuristic-approximate algorithm running times

The optimality of the heuristic-approximate policy search algorithm depends on the characteristics of the setting. Experiments on a CT Interruption game with moderate uncertainty (0.5 goal movement probability, 1.0 Gaussian variance (see Appendix 2 for the goal movement calculation)) shows that in 3% of the states, the best action assignment differs for the approximate-heuristic and exact policies. In such states, an action that takes the agent away from its goal is preferred, as this action limits its goal's movement and helps the agent to reach its goal faster. If the environment is certain, the heuristic-approximate algorithm is equivalent to the exact solution. When uncertainty is high in the environment, it is possible that this simple heuristic becomes insufficient and diverges from optimality. In this case, more sophisticated heuristics can be introduced to achieve better accuracy.

For our experiments, we used a heuristic that was specific for the CT interruption game. Heuristics that are suitable to the properties of the model should be explored to generalize the heuristic search approach to different settings. If such heuristics exist for the given setting, combining them with policy search methods is promising to significantly increase the efficiency of the solution.

4.4 Analysis of the Approximation Method

Previous sections have described how decoupling a single decentralized decision making model into individual models provides efficiency gains in the CT interruption game setting. This approximation method can be further generalized to collaborative decentralized decision making problems that are nearly decomposable to individual models. The necessary conditions for applying the decoupled approximation techniques are having a joint reward function that can completely decompose into individual rewards and a joint transition function. This section describes a generalization

of our approximate approach to general problems that satisfy this set of conditions.

In a general fully collaborative setting, there exists a set of agents $I = \{I_1, \ldots, I_n\}$ which share a common reward function R that is a combination of individual reward functions $R_1, \ldots, R_n$:

$$R(R_1, \ldots, R_n) \rightarrow \Re$$

The agents have a single joint action JA to be taken when all n agents agree on doing JA and a set of individual actions. When agents do not agree on taking action JA, the agents act individually. The individual decision making model of agent i selects the best individual actions a_i for the current state S_i. The estimated outcome of agent i for taking JA is calculated using two possible next states S_i^{JA} and S_i^{NJA} of agent i, which are explored by feeding the joint transition function T with S_i and JA. $S_i^{JA} = T(S_i, JA)$ is the next state of agent i after taking JA. $S_i^{NJA} = T(S_i, a_i)$ is the next state of agent i after taking individual best action a_i (NJA refers to "No Joint Action").

From the individual decision making model of agent i, two expected utility values are extracted; $EU_i(S_i^{JA})$ which is the expected utility of agent i at state S_i^{JA}. $EU_i(S_i^{NJA})$ which is the expected utility of agent i at state S_i^{NJA}.

The expected outcome (EO) of taking joint action JA is calculated as:

$$\begin{aligned} EO_{JA} =& R(EU_1(S_1^{JA}), \ldots, EU_n(S_n^{JA})) \\ & - R(EU_1(S_1^{NJA}), \ldots, EU_n(S_n^{NJA})) \end{aligned}$$

For any time step t, EO_{JA} is calculated for the current world state. JA is taken when EO_{JA} is positive. Otherwise individual optimal policies are followed.

The approach given above has two assumptions. It is assumed that the model has a single joint action and the deterministic transition function maps a given state to a single next state given action JA. Both of these assumptions can be relaxed. Given multiple joint actions $JA_1, \ldots, JA_K$, EO values are estimated for every JA_i. Among positive EO_{JA_i} values, the joint action with highest EO_{JA_i} is selected as the next action. If the joint transition function maps a current state S and joint action JA to a set of next states S_j^{t+1}, EO_{JA} is calculated as the weighted sum of S_j^{t+1} with corresponding transition probabilities.

As a result of the high complexity of constructing the optimal joint policy, we cannot provide an empirical comparison of the approximate policy with the optimal policy. However, observations of the agent player's interruption decisions during empirical studies using the CT interruption game, suggest the policy generated by the approximate method leads to good decisions. The agent appears to interrupt only when its goal has moved significantly. This behavior results from the approximate method embodying individual policies for the agent which are optimal, and interruption decisions that consider the effect of an interruption for both players. The person is only interrupted if the expected value of the interruption is larger than the expected value of acting individually. The method also takes into account the near decomposability of the problem and the additive structure of the reward function.

In calculating an expected value of a given world state with the decentralized MDP model in CT interruption game, Dec-MDP uses look ahead and includes the effect of future interactions in the utility calculation. The optimal joint policy may schedule multiple interruptions to maximize the joint reward. In contrast, our approximation method only includes the immediate effect of the interruption in the expected outcome calculation by ignoring the future possibilities of interaction. This is one reason why the joint policy constructed by our approximate method is sub-optimal. On the other hand, for single-shot games, where the agent is allowed to interrupt only once, the value for interaction is the same between the optimal policy and the approximate policy. If joint actions are always unbeneficial for the collaborative benefit, then the policy generated by the approximate algorithm is optimal because in this case the optimal joint policy is just the combination of optimal individual policies. The approximation method is expected to diverge from the optimal policy as the number of joint actions increases.

In many human-computer interaction settings, the interaction frequency is much lower than the frequency of individual actions taken by the computer agent and the person. In such settings, having optimal policies for individual actions and sub-optimal timing for the joint actions may be preferred for avoiding the significant overhead of solving the decentralized decision model. Therefore, we believe that this approximation may be useful for many human-computer interaction settings.

Our future work focuses on investigating the effect of interaction frequency on the optimality of approximate policy. We know that the approximate policy is optimal if there is no interaction and wish to determine the accuracy of our approximate algorithm as it is modified to produce multiple interruptions.

5. RELATED WORK

Optimally solving Dec-MDP models is known to be NEXP [1], making it necessary for us to investigate an efficient approximation to Dec-MDP solutions in the CT interruption setting. Several approximate solution methods have been investigated in prior work that also aimed to overcome the complexity barrier. This section briefly compares our approach to these alternatives.

Joint Equilibrium based Search for Policies (JESP) use dynamic programming to reach local optima [12], and its worst case complexity is greater than what we achieve with our approach. Several proposals rely on problem specific heuristics [14, 16]. Such problem specific heuristics do not address the general human-computer interaction settings considered in this paper. Xuan et al. introduces multiple heuristic functions to make communication decisions. Their hybrid heuristic is similar to our Expected Outcome of Interruption calculation [16]. It compares the information gain with the communication cost to decide whether to communicate, but the introduction of their idea is primarily to give details of such calculation. Our work can be considered as an extension that investigates the applicability of this approach in more detail. Beynier et al. is interested in more general settings in which time and resource constraints exist between multiple decentralized collaborative agents [2]. Similar to our approach, their work estimates the optimal joint policy by creating individual optimal policies for agents. Their work focuses on scheduling individual tasks by considering time and resource constraints. They don't consider the problem of estimating the value of joint actions. In future work, we hope to combine our EOI calculation with this algorithm to find new ways to better es-

timate the value of interruption.

The multi-agent decision making literature presents a variety of models that consider communication as a component of decision making. Decentralized Partially Observable Markov Decision Process with Communication (Dec-POMDP-Com) is an extension of the Dec-POMDP framework that has a distinct component for deciding when and how to communicate. Finding optimal solutions to Dec-POMDP-Com models is as hard as optimally solving Dec-MDP models [7]. For our problem, Dec-MDP-Com does not provide a more powerful representation than Dec-MDP models. Having an individual communication component as Dec-POMDP-Com models do, is not compatible with the interruption management problem, because it is necessary to incorporate interruption into the general decision making to investigate the trade off between acting individually and initiating interruptions.

A more general framework that covers cases in which agents may have different models is the Interactive Partially Observable Markov Decision Process (I-POMDP). I-POMDP allows defining individual reward functions for agents and therefore representing their preferences [6]. We will investigate I-POMDPs in future work, after collecting data for human behavior in CT interruption game.

6. CONCLUSION

Accurately calculating the expected costs and utilities of interruptions, taking into account the perspectives of both the interrupter and the party being interrupted, is important for effective interruption management. This paper investigates the estimation of interruption outcomes for two-party, collaborative human-computer activities in environments in which conditions may be rapidly changing, actions occur at a fast pace, and decisions must be made within tightly constrained time frames. This estimation problem is a new, interesting area of application for Dec-MDP methods. The paper presents a novel approximation approach which decouples computationally expensive Dec-MDPs into multiple individual MDP models and then recombines the interruption output of the models to provide an estimate of the multi-perspective, collaborative value of an interruption. This approximation technique can be generalized to Dec-MDPs that have completely decomposable joint reward functions and nearly decomposable transition functions and action sets. The approach provides significant efficiency gains when compared with the complexity of finding optimal solutions. In future work, we plan to compare the results of this approach empirically to exact methods and to other approximate solution methods to provide a detailed optimality and efficient analysis.

7. ACKNOWLEDGMENTS

The initial design of the CT interruption game was suggested by David Sarne to whom we are also grateful for many insightful discussions about interruption management. We thank David Sarne, Philip Hendrix and Heather Pon-Barry for helpful comments on earlier drafts of this paper. The research reported in this paper was supported in part by NSF grants IIS-0222892 and CNS-0453923 and in part by contract number 55-000720, a subcontract to SRI International's DARPA Contract No. FA8750-05-C-0033. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, DARPA or the U.S. Government.

APPENDIX 1: Creating Compact State Space

The set of configurations of our game framework is the combination of possible board positions of agent, person, and their individual goals. Using the symmetric geometry of our game board significantly reduces the search space to be considered to construct a complete policy.

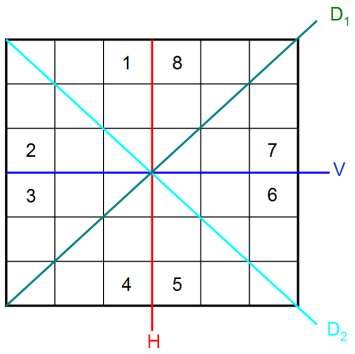

Figure 3: 4 symmetries divide game board into 8 symmetric pieces.

The game board has vertical (V), horizontal (H), and 2 diagonal (D_1 and D_2) symmetries that divide the board into 8 regions (see Figure 3). Every point in the game board can be transformed into the selected region by using these symmetries. Table 1 shows the set of symmetries that maps any point on the given board into region 1.

Region	Transformations	Region	Transformations
1	-	5	H, V
2	D_2	6	D_1
3	H, D_1	7	H, D_2
4	V	8	H

Table 1: Symmetry transformations that maps given regions into region 1

If a goal position is transformed into region R with a set of symmetries S, applying S to the player position preserves the relative positions of the player and the goal. A complete policy calculated for every initial state that has its goal position inside region R, is indeed a complete policy for the full game board as each initial state corresponds to a calculated configuration on region R. Therefore the size of the complete policy and the number of calculations required reduces to 1/8.

APPENDIX 2: CT Interruption Game Specific Functions

We present three CT interruption game specific functions that are used in Section 3 and Section 4 to update the belief state of the agent.

SpecialUpdate (SU) function is a special state estimator function that performs a game specific update on the belief

Algorithm 1 GaussianTable: Constructs a probability table that holds probability of goal's movement from initial position g to each board position

Require: Updated player position p', goal position g, gaussian variance τ
Ensure: Corresponding Gaussian Table GT
  for each cell $c \in B$ **do**
    if distance(c,p')<distance(g,p') **then**
      set $Table(c) = 0$
    else
      set $Table(c)=\frac{e^{\frac{-distance(c,g)}{\tau^2}}}{2\Pi\tau^2}$
    end if
  end for
  normalize GT
  return GT

Algorithm 2 P_M: Calculating goal's movement probability to a given cell

Require: Updated goal position g', updated player position p', current goal position g
Ensure: Probability of goal g moving to position g'
  $GT = GaussianTable(p',g)$
  return $Table(g')$

state before branching the policy tree to the next time steps (see Section 4.3 for more information). Given that p_A is the current position of the agent, branching to the next time step without reaching to the goal indicates that the goal is not located on position p_A. b is updated accordingly by setting $b(p_A)$ to 0. Next, b is normalized in a way that the distribution sums to 1, and it is updated with the state estimator function (see Section 4.3 for SE function).

Algorithm 3 SpecialUpdate (SU): Special Update to given Belief State

Require: Player position p, belief state b
Ensure: Updated belief state b'
  set $b(p)$=0
  normalize b
  $b' = SE(p, b)$
  return b'

8. REFERENCES

[1] D.S. Bernstein, S. Zilberstein, and N. Immerman. The Complexity of Decentralized Control of Markov Decision Processes. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence table of contents*, pages 32–37, 2000.

[2] A. Beynier and A.I. Mouaddib. A polynomial algorithm for decentralized Markov decision processes with temporal constraints. *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 963–969, 2005.

[3] M. Fleming and R. Cohen. A decision procedure for autonomous agents to reason about interaction with humans. *the AAAI 2004 Spring Symposium on Interaction between Humans and Autonomous Systems over Extended Operation*, pages 81–86, 2004.

[4] Michael Fleming and Robin Cohen. A user modeling approach to determining system initiative in mixed-initiative ai systems. In *UM '01*, pages 54–63, 2001.

[5] Y. Gal, A. Pfeffer, F. Marzo, and B. Grosz. Learning social preferences in games. *Proc. 19th National Conference on Artificial Intelligence (AAAI)*, 2004.

[6] PJ Gmytrasiewicz and P. Doshi. Interactive POMDPs: properties and preliminary results. *Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. Proceedings of the Third International Joint Conference on*, pages 1374–1375, 2004.

[7] C.V. Goldman and S. Zilberstein. Mechanism design for communication in cooperative systems. *Fifth Workshop on Game Theoretic and Decision Theoretic Agents*, 2003.

[8] B. Grosz, S. Kraus, S. Talman, B. Stossel, and M. Havlin. The influence of social dependencies on decision-making: Initial investigations with a new game. *AAMAS'04*, 02:782–789, 2004.

[9] E. Horvitz and J. Apacible. Learning and reasoning about interruption. In *ICMI '03*, pages 20–27, 2003.

[10] E. Horvitz, C. Kadie, T. Paek, and D. Hovel. Models of attention in computing and communication: from principles to applications. *Communications of the ACM*, 46(3):52–59, 2003.

[11] M.L. Littman, T.L. Dean, and L.P. Kaelbling. On the complexity of solving Markov decision problems. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 394–402, 1995.

[12] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 705–711, 2003.

[13] S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1995.

[14] D. Szer, F. Charpillet, and S. Zilberstein. MAA*: A Heuristic Search Algorithm for Solving Decentralized POMDPs. *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[15] Milind Tambe, Emma Bowring, Jonathan Pearce, Pradeep Varakantham, Paul Scerri, and FD Pynadath. Electric elves: What went wrong and why. In *AAAI'06*, 2006.

[16] P. Xuan, V. Lesser, and S. Zilberstein. Communication decisions in multi-agent cooperation: model and experiments. *Proceedings of the fifth international conference on Autonomous agents*, pages 616–623, 2001.