Communication Decisions in Multi-agent Cooperation: Model and Experiments *

Ping Xuan, Victor Lesser, and Shlomo Zilberstein
Department of Computer Science
University of Massachusetts at Amherst
Amherst, MA 01003
pxuan,lesser,shlomo@cs.umass.edu
Keywords: multi-agent communication/collaboration, coordinating multiple agents, MDP

ABSTRACT

In multi-agent cooperation, agents share a common goal, which is evaluated through a global utility function. However, an agent typically cannot observe the global state of an uncertain environment, and therefore they must communicate with each other in order to share the information needed for deciding which actions to take. We argue that, when communication incurs a cost (due to resource consumption, for example), whether to communicate or not also becomes a decision to make. Hence, communication decision becomes part of the overall agent decision problem. In order to explicitly address this problem, we present a multi-agent extension to Markov decision processes in which communication can be modeled as an explicit action that incurs a cost. This framework provides a foundation for a quantified study of agent coordination policies and provides both motivation and insight to the design of heuristic approaches. An example problem is studied under this framework. From this example we can see the impact communication policies have on the overall agent policies, and what implications we can find toward the design of agent coordination policies.

1. INTRODUCTION

Multi-agent coordination is the key to multi-agent problem solving. During coordination, communication is crucial for the agents to coordinate properly, since an agent usually only has a partial view of the system. In most occasions, it is unrealistic for the agents to reach perfect communication, i.e., to obtain the global state at all times. Here, we view communication as the abstraction of obtaining non-local information, and in general there should be a cost associated with it. Thus, the optimal policy for each agent must balance the amount of communication such that the information is sufficient for proper coordination but the cost for communication does not outweigh the expected gain. Many coordination policies have been studied [11], but in order to understand agent coordination in a quantitative way rather than qualitatively, we need to have a framework that deals with communication decisions in addition to decisions about agent actions. Such a framework would also be very instrumental for the design of heuristic policies.

However, communication decisions are often overlooked in the study of agent problem solving policies. This is partly due to the complexity involved in introducing communication decisions, and also due to the lack of a clear, quantified model for integrating communication aspects in agent problem solving. To this end, we propose a decision-theoretic framework to model a multi-agent decision process. Our focus is on cooperative, yet distributed systems, where all agents share the same goal of maximizing the total expected reward (utility). The key characteristics are that first, these agents are decentralized, and second, they share a common, *global* utility function, which depends on the global state and the joint action (the parallel invocation of each agent's local action.) This is different from the self-interested agents where each agent maximizes its own (local) utility. In our model, a *local* markov process can be defined for each agent. It is Markovian since the next local state depends stochastically only on the current local state and local action. However, note that because of the use of a global utility function and the absence of local utility functions, the local Markov process is not a standard Markov decision process. To be more specific, each agent knows its current local state, i.e., the agent's local state is fully observable (the local state could be partially observable, but this does not restrict our model, thus we make this simplification). However, an agent cannot observe the global state, i.e., it cannot see the local states of other agents. Instead, an agent can communication with the other agents and obtain such nonlocal infor-

*Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreement number F30602-99-2-0525 and by the National Science Foundation under Grant number IIS-9812755. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory, National Science Foundation, or the U.S. Government.

AGENTS'01, May 28-June 1, 2001, Montréal, Quebec, Canada.

mation, but communication costs may apply. This introduces a new kind of observability, where an agent can decide whether the global state (or part of the global state) need to be observed. More importantly, since the observation incurs cost, the agent's decision problem should now include communication decisions as well. Unlike partially-observable MDP (POMDP), here agents' belief states can potentially be changed by communication.

It has been recently shown that solving a decentralized MDP with global utility functions exactly are NEXP complete (i.e. nondeterministic exponential time) [2]. Thus, even without communication, multi-agent decision problems belong to a higher computational complexity class than standard MDP (P-complete) and POMDP (usually PSPACE complete) [12], and therefore cannot be reduced to them.

For clarity, let us assume that the global utility/reward function is known to all agents. This information is static and may be agreed upon before the agents form the team, i.e., it is assumed to be off-line information. Also, we assume that each agent behaves rationally and has the same mind power, i.e., they will independently (without any communication) reach the exactly same conclusion given a common problem such as solving a Markov decision process (MDP). This implies that all agents would follow the same joint action *if* the agents all know the current global state (perfect coordination). This is because in such case all agents are now presented with the same decision problem (given global state, global reward function, and a common start condition), thus they will independently solve the decision problem, reaching the exactly same decision — which is an optimal decision, and each agent then implements the local part of this decision. Note that all this is done in an independent fashion.

However, an agent cannot observe directly the local state of other agents, which is dynamic information. Instead, an agent has a choice of performing a communication action just after the previous action finishes and before the next action is chosen. The purpose of communication is for one agent to know the current local state of another agent. The content of the communication is local state information. We further assume that all communications are done in a synchronous fashion, which become a sub-stage in the agent's decision-action stage.

Whether the agent chooses to communicate or not, after the communication substage, the agent will now choose a local action based on all information available to this agent. This includes the history (i.e., previous states, previous actions, and previous communications). After the action is chosen, it is executed and the agent will now move to a next state and start the next stage.

The key problem here is to find the optimal local decision (whether it is a decision about regular action or the decision about whether to perform communication or not) based on all available information to each agent, and based on the global reward function and communication cost. Note that, because of the cooperative nature of our agents, the optimal policy is a tuple that consists of the local policy of *each* agent. This is different from typical agent decision problems where the goal is to find best local policy for *one* agent situated in a multi-agent environment (such as in game-playing).

We propose a multi-agent extension to Markov decision process to characterize the cooperative multi-agent decision problem. This extension is a decentralized one, with each agent making local decisions. This is different from the centralized extension, namely the Multi-agent Markov decision process (MMDP) defined by Boutilier in [3]. There, although agents have joint actions consisting of individual local agent actions, they do not have local states. Instead, each agent observes the global state directly. This characteristics greatly limits the applicability of MMDP to multi-agent systems, since a key property of multi-agent systems is that agents only have a partial view of the system. As a result, there is no need for communication of local state information, and the decision problem is to find the optimal joint action based on the global state (via solving a MDP or POMDP). The coordination problem there is for the agents to follow the same optimal joint action when there are multiple optimal joint actions. In comparison, in our definition of multi-agent decision process, we assume local agent states, and agents have to communicate to obtain other agent's local state information. This makes our decision problem an inherently *decentralized* one, which is fundamentally different from centralized ones which assume the global state knowledge [8, 15].

Our work is focused on the communication and coordination of cooperative agents, with the goal of finding best policy tuples and achieving the highest global reward. We directly model multi-agent problem-solving and communication into a decision process. This is different from the work by Gmytrasiewicz and Durfee [4], which considers agent decision-making from the perspective of an individual agent in a self-interested environment. In doing so, an agent must maintain its models of the other agents, which can including their models of other agents as well. This creates a recursion and hence the need of a recursive modeling method (RMM). The problem there is to find the best local policy for this agent.

Also related to this work is the theoretical study of decentralized control of finite state Markov processes [1, 9, 13]. There, both decentralized states and partitioned actions are assumed, and each agent's decision is based on its local information. However, they do not have communication decisions as well, instead, a fixed common information structure is assumed, usually in the form of a delay of non-local information, i.e., the global state information will be available for all agents after k stages.

The problem of decision making with the cost of communication is a very important one. In the single agent case, it is studied in [5, 6, 7], where communication takes the special form of an agent sensing the environment. In a multi-agent system, communication costs may relate to transmission fee, resource cost, etc.

In the following sections we present a definition of a decentralized cooperative multi-agent decision process, followed by an example system, and a discussion of some heuristic approaches. We conclude with some future directions.

2. MULTI-AGENT DECISION PROCESS

As mentioned before, our definition of a cooperative multi-agent decision process is based on decentralized decision processes. Each agent has its own Markov process. For clarity, we will assume that the system consists of two agents X and Y in the following notations. The same notations apply to systems with 3 or more agents as well just by increasing the arity of the vectors.

We define the set of agents $\alpha = \{X, Y\}$, and the tuple

$M^x = (S^x, A^x, p^x(s_j^x|s_i^x, a^x))$ defines the Markov process in X: its local state space is S^x, local action space is A^x, and the local state transition probability $p^x(s_j^x|s_i^x, a^x)$ defines the probability of resulting in state s_j^x when taking action a^x in state s_i^x. Similarly we can define Y's process $M^y = (S^y, A^y, p^y(s_j^y|s_i^y, a^y))$, and the global state space is $S^x \times S^y$ and joint action space is $A^x \times A^y$.

The global reward function $r_t(s_i^x, s_j^y, a_k^x, a_l^y)$ defines the reward the system gets when the global state is (s_i^x, s_j^y) and the joint action is (a_k^x, a_l^y). For simplicity we focus on finite-horizon problems only, and thus we define the reward at terminal time T is $r_T(s_i^x, s_j^y)$. Also, if (s_i^x, s_j^y) represents a terminal state (i.e., we allow the process to finish when certain relationship between s_i^x and s_j^y are met even when the current time is less than T), we also define terminal reward for those terminal states as $r_t(s_i^x, s_j^y)$, i.e., there is no further actions after t.

Now we add communication into this system. We assume a communication sub-stage where all communications complete before deciding the regular action. The event flow in one stage is depicted in Figure 1.

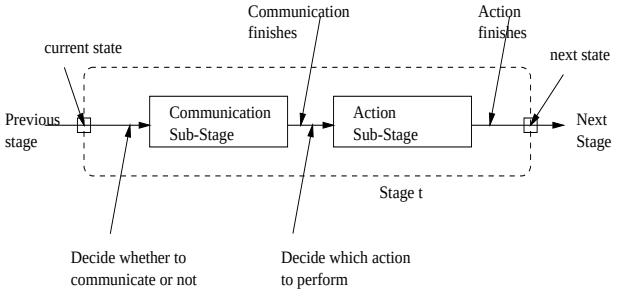

Figure 1: Communication Sub-stage

Let m_t^x and m_t^y denote the content of X and Y's communication during the communication phase. In particular, a *null* content means that the agent chooses not to communicate. Exactly how the information is shared after the communication clearly depends on the nature of communication. There are many communication types, to name a few:

- *tell*: in this type of communication, one agent simply tells its current local state to the other agent, i.e., information going outward. The sender will not know the receiver's local state as a result of this communication. In this type of communication, an agent knows the other agent's local state only when the other agent voluntarily decides to tell.
- *query*: here, the result of the query is that the query message sender agent receives the local state information of the other agent, i.e., information going inward. In reality, this usually means that the receiver sends back a feedback message, but such detail is not necessary in our abstract model of communication. Here, the sender agent does not reveal its local state information to the other agent. In other words, in this type of communication, an agent can know the other agent's local stage whenever it wants to do so, but there is no way to voluntarily tell other agent about its current local state.
- *sync*: this is the combination of the above two, in that when an agent performs a sync communication, it reveals its own state to the other agent, and at the same time obtain the other agent's local state. As a result of sync (regardless of which agent initiates the communication), both agents now know the each other's local state, and also the knowledge that the other agent knows the same. Again, in actual implementation more than one messages may be needed, but in our model it is sufficient to symbolize the process into one message communication.

Obviously, the choice of which communication type to choose is usually constrained by the actual communication ability of the agent. For example, if the agent's only communication means is to broadcast, then only the tell type is possible, and the agent must tell all other agents. However, it is very important to know that each type has different complexity. For example, with the sync type, the agents know that whenever they communicate, they know the global state, and as a result the previous history often becomes less important because the agents do not need the history information to reason under the uncertainty about the other agent's state and belief.

Let $c_t^x(s^x, m^x)$ and $c_t^y(s^y, m^y)$ denote the cost of communication in each agent given a particular time and state. In the simple case that a communication action has a fixed cost regardless of the time and state, a single function $c(m^x)$ (or $c(m^y)$) will suffice, where if m^x is null, the cost is zero, and otherwise, a fixed value c.

In summary, a decentralized multi-agent Markov process is defined by α, M^α, reward function $r(\cdot)$ and terminal conditions, communication actions and type, and communication cost $c(\cdot)$. It is Markov because the global state depends stochastically only on current state and current actions, although now actions include communication.

Now we try to define the decision problem. First, for each stage, each agent first observes its current state, then makes decision about communication, and then chooses an action. Thus, $((s_t^x, s_t^y), (m_t^x, m_t^y), (a_t^x, a_t^y))$ represents global events occurring at stage t. Thus, a *global* episode for this process can be described as:

$$\begin{aligned} I \quad = \quad & (s_0^x, s_0^y), (m_0^x, m_0^y), (a_0^x, a_0^y), \cdots, \\ & (s_t^x, s_t^y), (m_t^x, m_t^y), (a_t^x, a_t^y), \cdots, (s_{t'}^x, s_{t'}^y) \end{aligned} \quad (1)$$

Here, $(s_{t'}^x, s_{t'}^y)$ satisfies the terminal conditions (including the case when $t' = T$). Note that since we assume that initially both agents know each other's initial state, (m_0^x, m_0^y) would always be (*null, null*).

The probability for that episode to happen (i.e., the probability of having the state sequences $(s_0^x, s_1^x, ..., s_{t'}^x)$ and $(s_0^y, s_1^y, ..., s_{t'}^y)$), considering that communication does *not* change agent local state, and each agent's action is independent of the other agent's action, is:

$$p(I) = \prod_{t=0}^{t'-1} p^x(s_{t+1}^x|s_t^x, a_t^x) \cdot p^y(s_{t+1}^y|s_t^y, a_t^y). \quad (2)$$

For such an episode, its total reward is the terminal reward plus rewards collected at intermediate steps, and minus all the communication costs at both agents:

$$\begin{aligned} R(I) \quad = \quad & r_{t'}(s_{t'}^x, s_{t'}^y) + \sum_{t=0}^{t'-1} (r_t(s_t^x, s_t^y, a_t^x, a_t^y) \\ & - c_t^x(s_t^x, m_t^x) - c_t^y(s_t^y, m_t^y)) \end{aligned} \quad (3)$$

Since agents can choose not to communicate and thus they may not always sync themselves, the same past information set in one agent can correspond to many different paths in other agents, in general, each agent's decision about communication/action could be based on all locally available information, including the history and the current information. This means that the decision problem in general is history dependent, not Markovian. Let $H_t^{x,m}$ be the information available to agent X before it makes the communication decision m, and $H_t^{x,a}$ the information before the local action decision a, then,

$$\begin{aligned} H_t^{x,m} &= s_0^x, (m_0^x, m_0^y), a_0^x, \cdots, \\ &\quad s_k^x, (m_k^x, m_k^y), a_k^x, \cdots, s_t^x \qquad (4) \\ H_t^{x,a} &= H_t^{x,m}, (m_t^x, m_t^y) \qquad (5) \end{aligned}$$

Here, we see that the difference between a communication action and a regular action: a communication action is seen by both agents while a regular action is only known to the local agent.

Thus, the local decision problem for agent X is to find out a policy π^x that consists of two parts:

$$\begin{aligned} \pi^{x,m}: \quad H_t^{x,m} &\rightarrow m_t^x \\ \pi^{x,a}: \quad H_t^{x,a} &\rightarrow a_t^x \qquad (6) \end{aligned}$$

Here, $\pi^{x,m}$ defines a mapping from all local information to a communication decision, and $\pi^{x,a}$ defines a mapping from all local information to a decision about the next action. Together, π^x encodes all decisions X needs to make. $H_t^{y,m}$, $H_t^{y,m}$, π^y, $\pi^{y,m}$, $\pi^{y,a}$ can be defined similarly so we omit them here.

Based on a pair of local policies: (π^x, π^y), all possible episodes are defined by the set $\{I^{(\pi^x,\pi^y)}\}$, where

$$\begin{aligned} I^{(\pi^x,\pi^y)} &= (s_0^x, s_0^y), (\pi^{x,m}(H_0^{x,m}), \pi^{y,m}(H_0^{y,m})), \\ &\quad (\pi^{x,a}(H_0^{x,a}), \pi^{y,a}(H_0^{y,a})), \cdots, \\ &\quad (s_t^x, s_t^y), (\pi^{x,m}(H_t^{x,m}), \pi^{y,m}(H_t^{y,m})), \\ &\quad (\pi^{x,a}(H_t^{x,a}), \pi^{y,a}(H_t^{y,a})), \cdots, \\ &\quad (s_{t'}^x, s_{t'}^y) \qquad (7) \end{aligned}$$

$$\begin{aligned} p(I^{(\pi^x,\pi^y)}) &= \prod_{t=0}^{t'-1} (p^x(s_{t+1}^x | s_t^x, \pi^{x,a}(H_t^{x,a})) \\ &\quad \cdot p^y(s_{t+1}^y | s_t^y, \pi^{y,a}(H_t^{y,a}))). \qquad (8) \end{aligned}$$

Thus the total global expect reward for the policy pair (π^x, π^y) is,

$$E(\pi^x, \pi^y) = \sum_{I \in \{I^{(\pi^x,\pi^y)}\}} p(I) \cdot R(I) \qquad (9)$$

The decision problem is, therefore, to find the optimal pair of (π^x, π^y) such that it maximizes $E(\pi^x, \pi^y)$.

Obviously, calculating optimal policy is not going to be computationally feasible in most cases. Decentralized decision problems are NP hard in general [14]. Furthermore, since optimal policy is history dependent, the size of a policy (i.e., all possible histories) is too large to handle even for small problems. Thus, in most cases we cannot afford to calculate the exact optimal policy but rather need an approximation. For example, we can develop policies that use not all local history but only a part of it (presumably only some most recent information), therefore reduce the size of the policy drastically. However, even in those cases the complexity of the approximated policies may still be too high, especially if there is no efficient algorithms (such as dynamic programming for MDP/POMDP) to apply. On the other hand, heuristic solutions exist and are often easy to compute, and by examining a family of heuristic solutions we may indeed gain insight for designing good policies for agent coordination.

3. AN EXAMPLE PROBLEM

Now let's study an example and discuss the issues in decentralized multi-agent cooperation. Like in [3], we use a grid world domain. Assume two robots, X and Y, in a $L \times W$ grid world, (as shown in Figure 2, a 4×4 grid). Each agent's local process is simple: the local state is its position, and the local actions are to move left, right, up, down, and to stay where it is. Agent actions have uncertain outcomes: if an agent chooses to move, there is probability q (called the success rate) that it moves to the neighbor cell in the direction of the move, $(1-q)/4$ chance resulting in any of other neighbor cells, and the rest of times it does not move (i.e., gets stuck in the current cell). Agents cannot move off the grid. Both agents know the map of the grid, know one's own position in the grid (local state is observable), but they do not know the current position of the other agent unless they communicate.

0 X	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15 Y

Figure 2: A Grid World Example

The goal is for the two robots to meet (stay in the same cell) as early as possible, and within a deadline time T. Once the two robots meet, the process finishes even if the time is not yet T. We have a very simple global reward function r: any move is free and receives no reward. If they meet in t steps, the terminal reward is $\beta^t R$, where R is a constant and $\beta \leq 1$ is a time discount factor. If at time T the robots still have not met, the terminal reward is 0. As for communication, each robot can initiate communication, and each communication costs a constant c. The initial condition is that X is in position 0 and Y in 15, and they both know each other's initial position, and thus they are facing the same decision problem of finding an optimal (π^x, π^y).

This is a very simple multi-agent decision problem as we defined earlier. This problem domain is closed related to a class of real-world application domains such as coordinated exploration by multiple unmanned airborne vehicles (MUAV), multi-agent planning, and cooperative distributed problem solving. However, even with this problem, finding the best policy based on all local information, i.e., optimal (π^x, π^y) is computationally infeasible. Since in each stage, each agent can take 5 actions, each local action can have up to 5 resulting positions, and 2 communication choices, while

the other agent may have 16 different possibilities in its message content (assuming a 4×4 grid), that means an up to $(5 \times 5 \times 2 \times 16) = 800$ fold increase of local information history in each stage (since $H^{x,m}_{t+1} = H^{x,m}_t, (m^x_t, m^y_t), a^x_t, s^x_{t+1})$. Obviously, this means an explosion of the size of the local policy, and therefore is infeasible to compute a truly optimal policy.

As a side note, if the communication cost is 0, which means that both agents can communicate to obtain global information at all times (and hence no communication issue anymore), this problem reduces to a centralized problem (a typical MMDP problem mentioned earlier): we can construct a standard MDP based on global state, joint action, and global utility, and then solve the optimal global policy. Obviously, the expected utility of this optimal global policy gives us an upper-bound for our decentralized policies, since it does not consider communication costs. Later in our discussion we will use this upper-bound as our baseline results.

4. HEURISTIC APPROACHES

To deal with the complexity illustrated in the previous section, we seek to reduce the size of the policy by defining approximation policies that based on only a subset of $H^{x,m}$ and $H^{x,a}$, and use heuristic approaches. At one extreme, agents can communicate (assuming sync type is used) at every stage regardless of the history. In this case, global states are known to both agents at all times, and thus we can regard it as a centralized problem where global states are observable. Thus, we are facing a standard MDP, and we can use the standard value iteration algorithm to solve the optimal global policy and then partition the global policy into local policies, in other words, simulating a central controller. This is obviously not very good since many of the communications are redundant (too much coordination). At the other extreme, both agents can be totally silent and performs random actions (no coordination). Obviously this is also bad since they can do much better if they have a plan.

Thus, we modify these two extremes and compare two heuristic approaches. Both heuristics correspond to some popular social analogies, and they are general strategies that can be applied to many domains besides the grid world domain.

In one policy, agents select an optimal plan based on their last observed global state (i.e., the state where they last performed a sync communication), and they communicate (sync) whenever their current plan cannot be achieved (so that a new plan can be selected), but do not communicate if the plan is still achievable. This corresponds to the so-called "No news is good news" (NN) type of social convention, where if both parties are making progress as intended, they do not communicate (no news), however they will negotiate a new plan if the progress is not as intended. An example in this grid world problem is that assuming both agents first choose to meet at position 3 (top-right corner) in 3 steps, and they will not communicate if in each step they are getting closer to block 3. However, if X slips into block 4 when it tries to move to block 1, X will sync with Y and the agents will reselect a best position to meet, possibly block 6.

The other policy, in which no communication is needed, basically divide the problem into two independent parts and then each agent is committed to perform their part. In this case, this division of work may have high probability of success (i.e., in some cases agent may be able to recover from adverse outcomes), however they cannot change their plan dynamically, partly because they choose not to communicate at all. Of course, this approach depends on both agents knowing their initial global state so that they can choose the best division. We call this "silent commitment" (SC) approach. This approach also has its social counterpart, where when two parties decide to coordinate, they divide the work, set up a deadline when each party's work has to be completed, and then work on their own. Normally the deadline should be far enough so that both party feel comfortable. In our grid world problem, the agents may agree to be both at block 3 by time T (the deadline). Thus, even if X's first move to the left resulted in block 4, X will try to correct that and possibly still be able to enter block 3 by time T.

To compare these two heuristics (NN and SC), we note that, in NN, X's local policy uses only part of the history information, namely the time they last communicated, and the global state they discovered at that time (using the sync type of communication). This reduces $H^{x,m}_t$ and $H^{x,a}_t$ to l, s^x_l, s^y_l (and of course current information t, s^x_t), where l is the last time that $m^x_l \neq$ null or $m^y_l \neq$ null, and s^x_l is X's local state at time l, and s^y_l is Y's local state at time l (transmitted as part of the content of m^x_l or m^y_l).

The NN policy is based on a heuristic function $f(s^x_l, s^y_l)$, which decides a best short-term goal: a global state $(\hat{s}^x, \hat{s}^y)$, and progress functions for current state $g^x_l(s^x_t, \hat{s}^x, t)$ tells if X (or Y) has made sufficient progress at current t toward the goal state $\hat{s}^x$ $(\hat{s}^y)$. For our example, f simply tells the mid-point of a shortest path between the two agents, and g tells if the distance from the current local position to the mid-point has been shortened as planned (i.e., reduced by $t - l$). Thus, the policy π^x is,

$$\pi^{x,m}(t, s^x_t, l, s^x_l, s^y_l) = \begin{cases} \text{sync} & \text{if } g^x_l(s^x_t, \hat{s}^x) \text{ is false;} \\ \text{null} & \text{otherwise.} \end{cases}$$

$\pi^{x,a}$ would choose the best local action so that the short term goal $\hat{s}^x$ is mostly likely to be reached.

On the other hand, the SC heuristic chooses a completely different subset from local history: only the initial global state! It uses a heuristic function $h(s^x_0, s^y_0)$ – note the initial global state here – which also decides a goal state $(\hat{s}^x, \hat{s}^y)$, in our case the mid-point of a shortest path between X and Y's initial states. The difference between NN and SC is that now in SC the agent has T time to reach its own goal state, but in NN a progress function imposes stronger constraints and thus becomes a dynamic plan.

SC never communicates, thus, $\pi^{x,m}(s^x)$ is always null. $\pi^{x,a}$ then chooses the best action so that $\hat{s}^x$ may be reached. Note that this policy is independent of time, i.e., similar to a stationary policy for infinite horizon problems.

We can see that in both heuristics the size of policy is significantly reduced so that it is computationally feasible. Also, we note that the calculation of $\pi^{x,a}$ involves optimization, but in both cases the optimization is completely local, i.e., both try to maximize the probability that $\hat{s}^x$ (or $\hat{s}^y$) to be reached. In other words, a local utility measure is introduced. In NN the utility measure is a short-term, dynamic one, and in SC it is a fixed one. As a result, the local optimization problem in each is now a standard MDP and thus can be solved using typical dynamic programming.

Intuitively, NN pays communication costs to reduce un-

certainty in coordination, but SC avoids communication at the cost of increased uncertainty. However, we note that both NN and SC do not respond to communication costs — the calculation of communication decisions does not involve communication costs. Thus, there might be cases that less uncertainty does not offset the cost paid for communication (when using NN), or that a communication could have resulted in a large increase of expected utility (when using SC). This prompt us to try to find a hybrid policy that has the best of both worlds: it communicates to reduce uncertainty when the cost is less than expected gain, and avoids communication when the cost would be greater than expected gain.

Just like SC or NN, the hybrid heuristic starts by introducing local goals. Like NN, this goal could be a dynamic goal: an agent assumes that the other agent is making progress toward its local goal if it does not receive communication from the other agent. But unlike NN, where goals are changed only when the progress is not as expected, in this hybrid heuristic, the agent is always deciding if there is a better goal (assuming the other agent is making good progress). However, even when a better goal exists, the agent will first calculate the expected gain, and compare it to the communication cost. If the latter is greater, the agent remains silent, otherwise, it communicates, and a new goal will be established. Thus, the goal could potentially be a long term goal (like in SC), when communication cost is high enough.

We need to note, though, that it may be quite difficult to precisely decide if there exists a better goal, and how much the expected gain is. This is also where heuristic functions may be applied. In our example we use this simple heuristic: try the adjacent positions of the current goal and check if the agents may meet sooner in any of these positions. If so, a better goal is found, and the potential gain is the utility difference due to meeting earlier, times the probability that both agents always making good progresses toward the new goal. For example, if current time is 2 and the agents have a chance of meeting at time 4 instead of 5, the difference is $\beta^4 R - \beta^5 R$, and since both agents have 2 more steps to go, the chance that they are always "on track" is $q^2 \times q^2$, and therefore the expected gain (heuristic value) is $q^4 R(\beta^4 - \beta^5)$.

5. RESULTS AND DISCUSSIONS

In the following we evaluate the example problem and try to discuss the implications of these heuristics with regard to multi-agent coordination. We will compare the baseline (the centralized, no communication cost case mentioned in section 3), the SC policy, the NN policy, and the Hybrid policy.

Using our example, we study how the expected global reward changes with the heuristics, when we vary the deadline T, the cost of communication c, the time discount factor β, and the success rate q. We assume $R = 100$. To define our heuristic functions f and h when there exists more than one shortest path, we use the path that closest to the straight line between X and Y, i.e., the mid-point is the one that closest to the straight line mid-point between X and Y.

First we study the expected rewards of NN, SC, and Hybrid with respect to the communication costs, as in Figure 3. Here $q = 0.96$, $T = 5$, and $\beta = 0.95$. The baseline value is 89.28, which gives an upper bound. Evidently, in SC, the expected reward (y-axis) does not change at all,

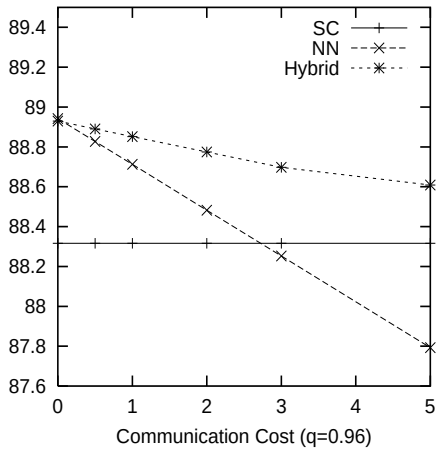

Figure 3: Communication Cost

because this policy never utilizes communication. The NN policy has better performance when communication is free, but it does not scale with communication cost, thus we see the crossing point when communication cost increases. This illustrates the general intuition: communication is a rational thing (will achieve better performance) unless the cost of communication is too high. In our case, communication in NN indicates a change of short-term goal (de-commitment or goal modification in typical multi-agent coordination language). This is rational as long as the communication cost is low. Otherwise, SC (where commitment cannot be changed and the agent always tries to honor the commitment despite local failures) would be a better solution. The Hybrid line clearly reinforces our points, and it shows that such a hybrid heuristic indeed gets the best of both worlds, and performs better than any of the other two heuristics when communication cost is present. It performs as good as NN when cost is 0, and it scales like SC when cost is very high.

How soon the cost of communication outweighs the benefit of more information depends on the uncertainty in the system. Clearly, with a higher q, meaning the robots' movements are more reliable, the amount of uncertainty in the system is not high, and hence the increase of performance due to the reduction of uncertainty via communication is not much. Therefore, the crossing point in the figure would come earlier when q is greater. This is confirmed in our study although due to space limitations we do not display the results here. The benefit of the Hybrid heuristic is that it eliminates the guess work when choosing over SC or NN, since it adjusts to communication cost automatically.

Next, in Figure 4 we vary the time discount factor β and see how these heuristics react. The smaller β is, the quicker the reward decreases, thus the agents have an interest to achieve the goal as soon as possible (if β=1 then the reward is the same as long as they meet before the deadline.) First we note that not surprisingly, Hybrid is the best. NN is very close to Hybrid, and in general is better than SC, since by resolving the uncertainty via communication they agents can adapt quicker. Also, it is interesting to note that when β decreases, performance of SC decreases slower than the NN policy, and depends on the cost of communication, the SC line can meet with NN:c lines (although it is always under Hybrid:c lines), where c is the communication cost. The reason here is that, when β decreases, the cost of com-

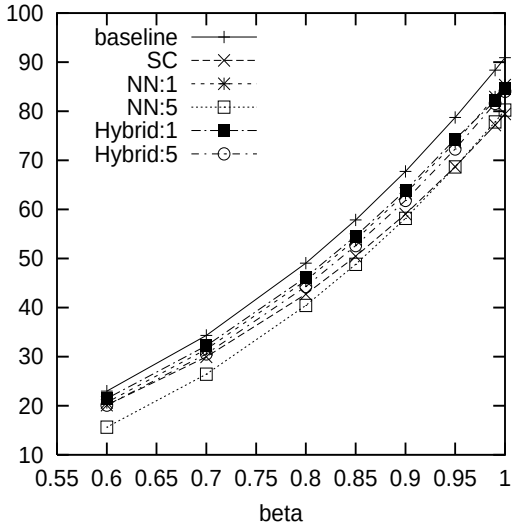

Figure 4: Beta: Discount Factor

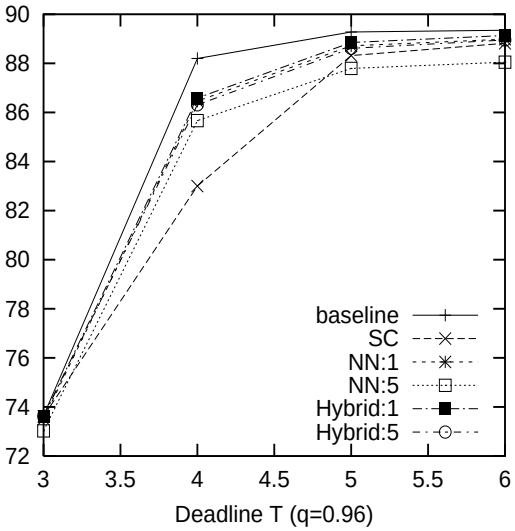

Figure 5: Deadline

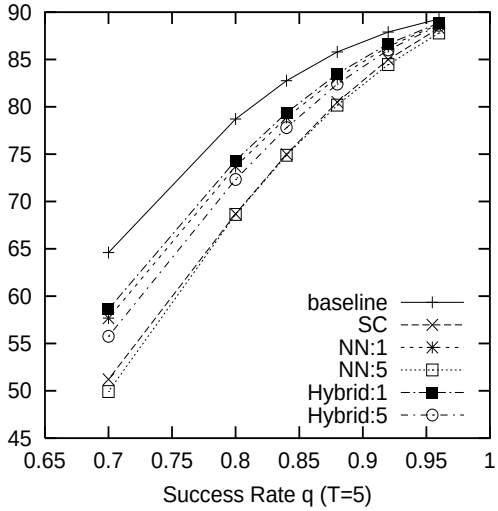

Figure 6: Success Rate

munication becomes more and more comparable with the reward, since the communication cost is fixed. In the extremely case, the reward can be discounted so much that it is smaller than the cost of communication. Obviously in this case the rational decision is not to communicate (Hybrid will reach the same conclusion via its calculation, so it performs consistently better than SC). The implication is that, in a time critical system, the agents should choose to communicate earlier than later, since the weight of communication may become greater when time passes.

Next, in Figure 5 we vary deadline T and see how they perform from very time-constrained (3) to having plenty slack time (6). Here β is fixed at 0.95. We notice that when deadline is tight, Hybrid and SC are slightly better than NN since agents do not have time for an alternative plan when their initial plan fails. In these cases all three heuristics are quite close to optimal (baseline upper bound). On the other hand, when the deadline is far away, both NN and SC would allow agents to reach their eventual goals (in the case of SC, agents have enough time to recover from earlier failures), thus in this case their performance again becomes close. (Of course, β close to 1 still needed). In this case, all three heuristics are also quite close to the optimal. The most interesting case is in the middle of the lines, when the deadline is not so tight, the reduction of uncertainty and the use of dynamic goal adaption can certainly help agents achieve their group goals in a timely fashion, and hence Hybrid and NN performs better. However, NN communicates whenever there are uncertainty about the current commitment, so it may communicate too much in high cost situations, and results in a lower expected utility than SC.

It is also observed that with higher uncertainty (lower q) the agents needs to communicate much more often in NN policy, thus causing the performance difference between SC and NN to be smaller. Again, Hybrid is the dominating one among the three heuristics, suggesting the need for explicitly reasoning about communication costs.

Finally, it is interesting to see how SC, NN, and Hybrid differ with the success rate q – the indicator that how reliable the agent's actions are. In Figure 6, we again see that when the uncertainty is low, all policies achieve about the same performance (possibly close to the optimum). The Hybrid policy is again the dominating policy, outperforms both NN and SC solidly. Between NN and SC, though, we can see that if communication cost is zero or low, when uncertainty increases NN is much better than SC. We need to note that the time constraints play a very importance role here: when T becomes greater (longer deadline) the SC line can become better than some of the NN lines, in particular, the ones with high communication costs (not shown here due to the space constraint). The underlying reason is that when agents have enough time to perform local recovery (as in SC) without any communication, the lost of performance due to not being able to de-commit can be offset by not spending on communication, especially when the cost of communication is quite high, and the amount of communication needed could be quite large when uncertainty is high.

Overall, these three policies give us some intuition about when to use a policy that relies heavily on communication, and when to use a policy that relies little on communication. In general, frequent communication (such as NN)

often means short-term/dynamic commitments, while low communication policies (such as SC) often use long-term, unchangeable, commitments. The optimum may be somewhere in the middle, although the computation demand is prohibitive. The Hybrid policy demonstrates the need of following these intuitions. More importantly, however, the success of the Hybrid policy indicates that it is necessary that we deal with communication cost directly and explicitly when designing a policy. The result of this integration is that the new policy can adapt better to different situations, and also is more effective in terms of correctly reasoning about expected utilities and decision making. In other words, we should begin to treat communication decisions the same way we do for agent action decisions.

6. SUMMARY AND FUTURE WORK

In this paper we study the impact of communication decisions on the construction of control policies in a multi-agent setup. We argue that communication decisions are a fundamental aspect of the agent decision problem, and that the problem solving model should integrate these decisions explicitly. We have defined a decentralized framework of a multi-agent MDP, described how communication and the cost of communication should be modeled into such a framework, what is optimality in this framework, and what kinds of approximations can be used. Although the optimality problem usually is computationally prohibitive, approximation methods and heuristics exist and can give us very important insights into some of the most important problems of multi-agent coordination, for example, when the agents should use dynamic commitments.

The study of the foundation of coordination in multi-agent system has become more and more important, and we believe that a decentralized approach provides a formal foundation and captures the complexity of the problem of coordination. A lot of work remains to be done. First, since the optimal policy is history-dependent, it would be very interesting to see that under what situations an approximation still maintains the optimality, i.e., under what conditions it is safe to ignore a large part of the history information?

We are still in search for efficient algorithms for approximation approaches. Since in general dynamic programming (hence the standard value iteration and policy iteration algorithms) cannot be used [16] in decentralized decision problems, we need to know if there are special cases that dynamic programming is possible, and if there are other efficient computation techniques that are suitable for multi-agent MDPs. Besides efficient approximation methods, learning techniques such as [10] may also contribute to decentralized decision-making.

Finally, decentralized MDPs may be extended so that they cover infinite-horizon processes and also are able to deal with the case where the agents do not have the same static global understanding (for example, the robots do not have the complete map). Also, it will be very interesting to study communication when agents are clustered into sub-groups in a multi-agent system. This would be very important when the system scales up.

Acknowledgements

The authors would like to thank Andy Barto and Dan Bernstein for fruitful discussions of this problem.

7. REFERENCES

[1] M. Aicardi, F. Davoli, and R. Minciardi. Decentralized optimal control of markov chains with a common past information set. *IEEE Transactions on Automatic Control*, AC-32:1028–1031, 1987.

[2] D. S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of markov decision processes. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, 2000.

[3] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conferences on Artificial Intelligence (IJCAI-99)*, July 1999.

[4] P. J. Gmytrasiewicz and E. H. Durfee. Rational interaction in multiagent environments: Coordination. *Autonomous Agents and Multi-Agent Systems Journal*, 1999.

[5] E. Hansen. Cost-effective sensing during plan execution. In *Proceedings of the Twelth National Conference on Artificial Intelligence*, 1994.

[6] E. Hansen, A. Barto, and S. Zilberstein. Reinforcement learning for mixed open-loop and closed-loop control. In *Proceedings of the Ninth Neural Information Processing Systems Conference*, December 1996.

[7] E. A. Hansen and S. Zilberstein. Monitoring the progress of anytime problem-solving. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 1229–1234, 1996.

[8] Y. C. Ho and T. S. Chang. Another look at the nonclassical information problem. *IEEE Transactions on Automatic Control*, AC-25:537–540, 1980.

[9] K. Hsu and S. I. Marcus. Decentralized control of finite state markov processes. *IEEE Transactions on Automatic Control*, AC-27:426–431, 1982.

[10] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. 11th International Conf. on Machine Learning*, pages 157–163, 1994.

[11] G. O'Hare and N. Jennings, editors. *Foundations of Distributed Artificial Intelligence*. John Wiley, 1996.

[12] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[13] N. R. Sandell, P. Varaiya, M. Athans, and M. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, AC-23:108–128, 1978.

[14] J. N. Tsitsiklis and M. Athans. On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control*, AC-30:440–446, 1985.

[15] H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):138–147, 1968.

[16] T. Yoshikawa. Decomposition of dynamic team decision problems. *IEEE Transactions on Automatic Control*, AC-23:443–445, 1978.