Communication in Multi-agent Markov Decision Processes *

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

UMass Computer Science Technical Report 2000-01

Abstract

In this paper, we formulate agent's decision process under the framework of Markov decision processes, and in particular, the multi-agent extension to Markov decision process that includes agent communication decisions. We model communication as the way for each agent to obtain local state information in other agents, by paying a certain communication cost. Thus, agents have to decide not only which local action to perform, but also whether it is worthwhile to perform a communication action before deciding the local action. We believe that this would provide a foundation for formal study of coordination activities and may lead to some insights to the design of agent coordination policies, and heuristic approaches in particular. An example problem is studied under this framework and its implications to coordination are discussed.

1 Introduction

In a multi-agent system, each agent normally only sees a partial view of the whole system. This implies that an agent only observes part of the global system state. Although agents do have the ability to communicate with each other, it is usually unrealistic for the agents to communicate

*Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreement number F30602-97-1-0249 and by the National Science Foundation under Grant number IIS-9812755 and number IRI-9523419. 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.

their local state information to all agents at all times, because communication actions are usually associated with a certain cost. Yet, communication is crucial for the agents to coordinate properly. Therefore, 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.

We propose a decision-theoretic framework to model a multi-agent system. Our focus is on fully cooperative systems, where all agents share the same goal of maximizing the total expected reward. This is different from the self-interested agents where each agent maximize its own (local) utility. In our model, we assume that each agent knows its current (local) state, i.e., the agent's local state is immediately (fully) observable. An agent has a set of local actions to choose from, and associated with each action is a probabilistic distribution of resulting (local) states. This defines a local Markov process, because the next state depends stochastically only on the current state and the current action. For cooperative agents, this local Markov process does not have a *local* reward function, rather, there exists a global reward function which depends on the global state (which contains each agent's current local state), and the joint action (the parallel invocation of each agent's local action.) In other words, without knowing the exact local states/actions of other agents, an agent cannot know the exact cost/reward associated with its local action. However, note that it is possible for agents to reason (with uncertainty, of course) about the possible local states/actions in other agents in certain situations.

We assume that such global 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 behave rationally and have the same thought process, 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 know the current global state. This is because in such case all agents are now presented with the same decision problem (given global state and global reward function), 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. Exactly how and which content is shared after the communication depends on the type of communication. For example, one type of communication may simply tell its current local state to some other agent (not asking for other agent's state information), but another type of communication may ask some other agents for their local states, still another type of communication may share both local states between the two communicating agents. 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 sub-stage, 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.

Therefore, the key problem here is to find the optimal 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 an agent. Obviously, since the agents may choose not to communicate, the available local information may not always allow the agent to know (or be able to reason with certainty) the current global state. Thus, the decision has to be based on local information. Also, the agent cannot assume that other agents know its local information, and therefore each agent's set of information are likely to be different, i.e., they are likely to be facing different decision problems with different local information available.

This work is related to Boutilier's work [2], which introduced the multi-agent extension to standard Markov decision process, namely the Multi-agent Markov decision process (MMDP). However, in Boutilier's work, although agents have joint actions consists of individual local agent actions, they do not have local states, instead, each agent observe the global state directly. As a result, there is no communication for local state information, and the problem is to find the optimal local action based on the global state, and the coordination problem there is for the agents to follow the same optimal joint action when there are multiple optimal joint actions. In this work 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 [6, 11]. We focus on primarily how to find the optimal decision, rather than how to deal with multiple optimal joint actions — by assuming the same thought process in all agents, they would select the same optimum when multiple optima exist. If agents do not have the same thought process, then the approaches presented in [2] would complement this work when coordinating agents' local actions to follow the exact same optimum.

This work is also very closely related to theoretic works on decentralized control of finite state Markov processes [1, 7, 8]. There, both partitioned states and partitioned actions are assumed, and each decision making agent's decision is based on its local information. However, they do not explicitly model communication actions, instead a fixed common information structure is assumed, usually in the form of a delay of nonlocal information, i.e., the global state information will be available for all agents after k stages. Thus, agents still do not need to make decision on communication. In our work, however, since agents need to make decision on their communication of nonlocal information, they may not have a fixed common information structure.

The problem of decision making with the cost of communication is also studied in [3, 4, 5], where communication takes the special form as an agent *sensing* the environment, where sensing requires a cost but can provide information to resolve the uncertainty about the environment. Our work extends to the case when a team of decentralized agents are cooperating.

In the following sections we first present our definition of a decentralized multi-agent Markov decision process (MMDP), define the problem and notations. Then we study an example system and discuss the issues associated with solving such a problem. We discuss some heuristic approaches and study their performances and give some insight on the design of agent coordination strategies. Finally, we draw some conclusions, and also point out some future directions.

2 Multi-agent MDP

As mentioned before, our definition of an MMDP 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 by increasing the arity of the vectors.

We define the set of agents $\alpha = \{X, Y\}$, and $M^x = (S^x, A^x, p^x(s_j^x | s_i^x, a^x))$ 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. Clearly, this process is Markovian since the next state depends stochastically only on current state and current action. Similarly we can define Y's process $M^y = (S^y, A^y, p^y(s_j^y | s_i^y, a^y))$. Clearly, 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.

So far we defined a decentralized Markov process with a global reward function, which implies a team of autonomous agents doing cooperative work. Now we add communication into this system. As discussed before, we assume a communication sub-stage where all communications complete before deciding the regular action. The control flow in one stage in depicted in Figure 1. Let m_t^x and m_t^y denote the content of X and Y's communication during the communication phase.

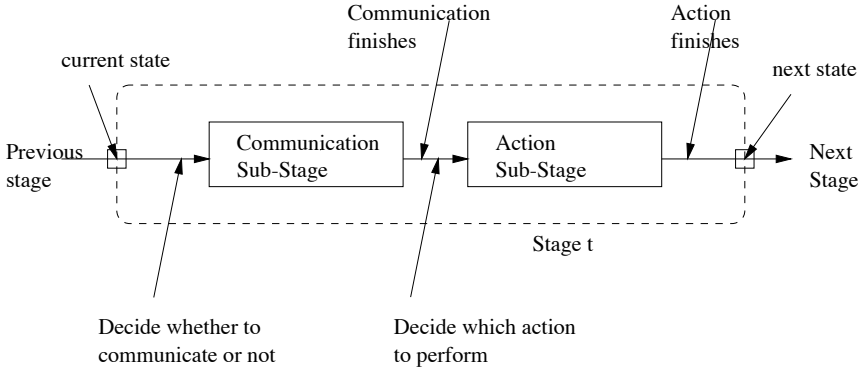

Figure 1: Communication Sub-stage

In particular, if the agent choose not to communicate, the content will be *null*. Each agent can initiate communicate independently, we assume that the message format is mutually understood and that no message is lost/changed during communication.

As discussed before, there exists many communication types which define different content exchanges. These are some simple examples:

- *tell*: in this type of communication, each agent simply send a message telling its current local state to the other agent. The sender will not know the receiver's local state as a result of the 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 when an agent sends a query message, it is expected to get the other agent's local state when the communication completes (in reality, this usually means that the receiver sends a feedback message). However, 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 global state (and the knowledge that the other agent knows the same as I do).

Obviously, the choice of which communication type to choose usually is 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. 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 current 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 process. First, for each stage, each agent first observe its current state, then make decision about communication, and then choose an action. Thus, we can use (s_t^x, m_t^x, a_t^x) and (s_t^y, m_t^y, a_t^y) to represent all events occurring at this stage, and thus $((s_t^x, s_t^y), (m_t^x, m_t^y), (a_t^x, a_t^y))$ for the global events. Thus, a *global* episode for this process can be described as:

$$\begin{aligned} I \;=\; & (s_0^x, s_0^y), (a_0^x, a_0^y), \\ & (s_1^x, s_1^y), (m_1^x, m_1^y), (a_1^x, a_1^y), \cdots, \\ & (s_t^x, s_t^y), (m_t^x, m_t^y), (a_t^x, a_t^y), \cdots, (s_{t'}^x, s_{t'}^y) \end{aligned} \tag{1}$$

Here we assume that initially both agents know each other's initial states thus there is no need for (m_0^x, m_0^y) (they would both be null), and $(s_{t'}^x, s_{t'}^y)$ satisfies the terminal state conditions (including the case when $t' = T$).

For such an episode, its total reward is,

$$R(I) = r_{t'}(s_{t'}^x, s_{t'}^y) + \sum_{t=0}^{t'-1} r_t(s_t^x, s_t^y, a_t^x, a_t^y) - \sum_{t=1}^{t'-1}(c_t^x(s_t^x, m_t^x) + c_t^y(s_t^y, m_t^y)). \tag{2}$$

And 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 $(s_0^y, s_1^y, ...s_{t'}^y)$ is,

$$p(I) = \prod_{t=0}^{t'-1} p^x(s_{t+1}^x | s_t^x, a_t^x) \cdot p^y(s_{t+1}^y | s_t^y, a_t^y). \tag{3}$$

This is because communication does not change agent local state, and each agent's action is independent of the other agent's action.

As discussed before, each agent's decision about communication could be based on all locally available information, and this include the history. Let $H_t^{x,m}$ be all the information available to agent X before it makes the decision about communication, then,

$$H_t^{x,m} = s_0^x, a_0^x, \cdots, s_k^x, (m_k^x, m_k^y), a_k^x, \cdots, s_t^x. \tag{4}$$

Similarly, all information available just before the agent makes the decision about the local action is,

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

Here, we see that the difference between a communication action and a regular action: a communication action is observed 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:

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

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.

Now we are ready to define the decision process. 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

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

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

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

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

The MMDP 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 [10], and in our case 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 needs an approximation. For example, we can develop policies that uses not all local history but only a part of it (presumably only some most recent information), therefore reduces the size of the policy drastically. However, even in those cases the complexity of the appoximated policies may still be too high, especially when there is no efficient algorithm such as dynamic programming 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

Now let's study an example and discuss the issues in MMDP. Assume two robots, X and Y, in a 4 by 4 grid world, as shown in Figure 2. An agent's local state is its position, from 0 to 15, and their local actions (move left, right, up, down, and stay where it is.) If an agent chooses to move, there is probability q that it moves to the neighbour cell in the direction of the move, $(1-q)/4$ chance resulting in any of other neighbour cells, and the rest of times it does not move (i.e., get stuck in the current cell). Agents cannot move off the grid. Both agents know the map of the grid, know its own position in the grid (local state is observable), but they do not know the current position of the other agent unless they communicate (nonlocal state not observable but can be obtained via communication.)

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

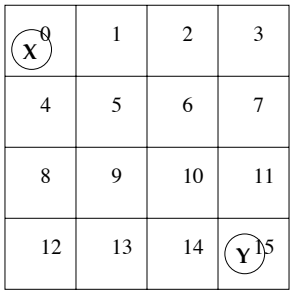

Figure 2: A Grid World Example

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 MMDP as we defined earlier. However, even with this problem, finding a 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 have 2 communications choices, while the other agent may have 16 different possiblities in its message content, that means a 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 truely optimal policy.

Thus, 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. 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 does not communication if the plan is still achievable. This corresponds to the so-called "No news is good news" 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), and they will not communicate if in each step they are getting closer to block 3. However, if X slipped into block 4 when it tries to move to block 1, X will sync with Y and 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" 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.

4 Comparison of the Two Heuristics

Here we compare the two heuristics mentioned above, namely the "No news is good news" (NN) one and the silent commitment (SC) one.

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), i.e., reduces $H_t^{x,m}$ and $H_t^{x,a}$ to l, s_l^x, s_l^y (and of course current information t, s_t^x), where l is the last time that $m_l^x \neq$ null or $m_l^y \neq$ null, and s_l^x is X's local state at time l, and s_l^y is Y's local state at time l (transmitted as part of the content of m_l^x or m_l^y).

The NN policy is based on a heuristic function $f(s_l^x, s_l^y)$, which decides a best short-term goal: a global state $(\hat{s}^x, \hat{s}^y)$, and progress functions for current state $g_l^x(s_t^x, \hat{s}^x, t)$ tells if X (or Y) has made sufficient progress at current t toward the 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,

$$\pi^{x,m}(t, s_t^x, l, s_l^x, s_l^y) = \begin{cases} \text{sync} & \text{if } g_l^x(s_t^x, \hat{s}^x) \text{ is false;} \\ \text{null} & \text{otherwise.} \end{cases}$$

$\pi^{x,a}$ 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 $h(s_0^x, s_0^y)$ – 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 agents have 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 Markov decision process and thus can be solved using typical dynamic programming such as backward induction (in finite-horizon problems.)

In the following we evaluate the example problem and try to discuss the implications of these heuristics with regard to multi-agent coordination. Communication in multi-agent system is often associated with negotiation in addition to information sharing. However, we notice that the reason for negotiation is that agents having different information and also different belief about other agents, therefore when agents have the same decision process, negotiation really reduces to information sharing since each agent could individually reason the same result of negotiation after the information sharing.

Using our example, we study how the expected global rewards change with the two heuristics, when we vary the deadline T, the cost of communication c, and the time discount factor β, and the certainty factor 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.

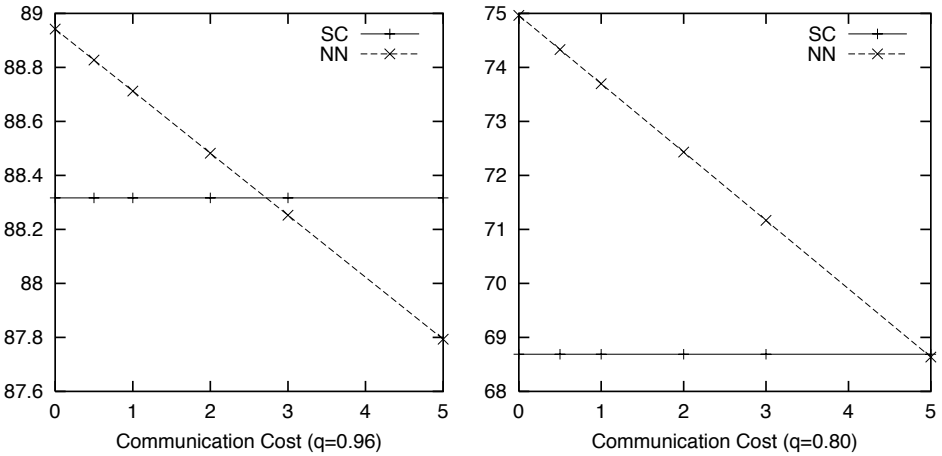

Figure 3: Communication Cost

First we study the expected rewards of NN and SC with respect to the communication costs, as in Figure 4. In the left sub-figure $q = 0.96$, and the right one $q = 0.8$. Both use deadline $T = 5$, and $\beta = 0.95$. Evidently, in SC, the expected reward (y-axis) does not change at all, 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 crosspoint 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 much. 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.

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 much, and hence the increase of performance due to the reduction of uncertainty via communication is not much. Therefore, we can see that the crossing in the left sub-figure come quite earlier than in the right sub-figure.

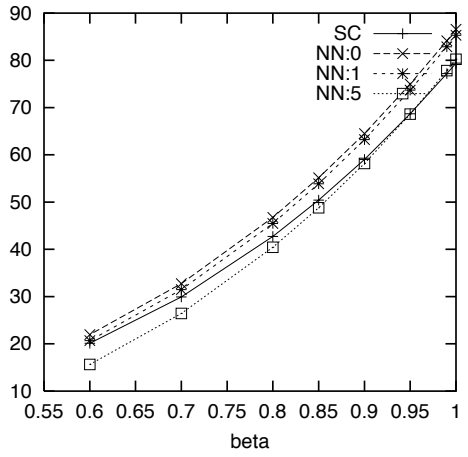

Figure 4: Beta: Discount Factor

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 in general NN is better (unless c is too large), 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, where c is the communication cost. The reason here is that, when β decreases, the cost of communication 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. The implication is that, in a time critical system, the agents should choose to communication 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)

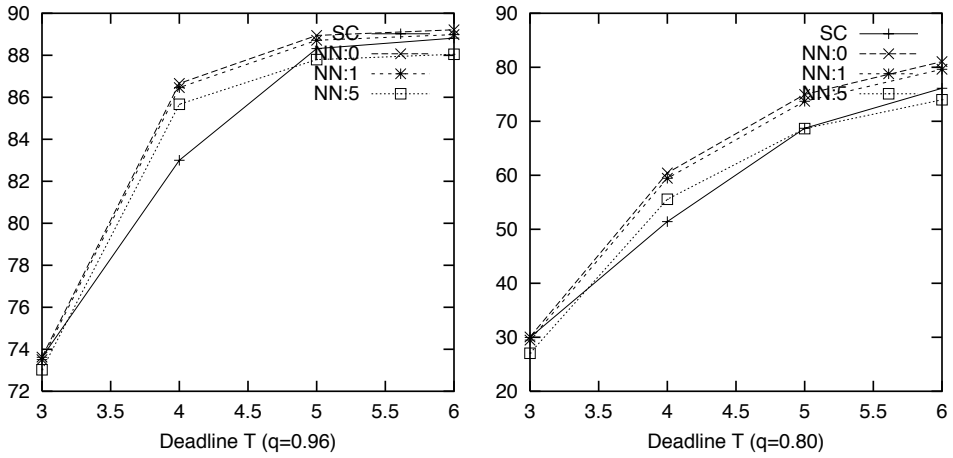

Figure 5: Deadline

to having plenty slack time (6). Here β is fixed at 0.95. We notice that when deadline is tight, SC is slightly better than NN since agents do not have time for an alternative plan when their initial plan fails. Of course, when the deadline is not so tight, reduction of uncertainty and the use of dynamic goal adaption can certainty help agents achieve their group goals therefore NN (communication whenever there are uncertainty about the current commitment) in a timely fashion. Finally, 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, $beta$ close to 1 still needed).

With higher uncertainty ($q = 0.8$) the agents needs to communicate much more often in NN policy, and thus we see that the performance difference between SC and NN is smaller than the case when $q = 0.96$.

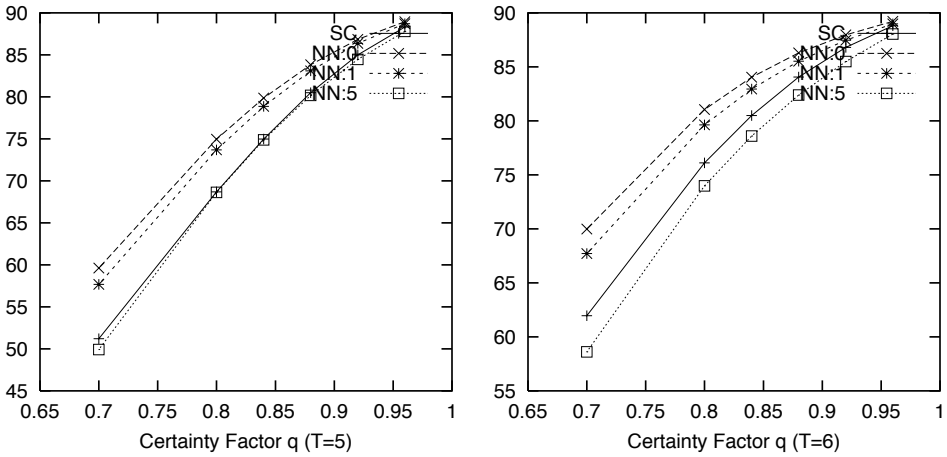

Figure 6: Certainty Factor

Finally, it is interesting to see how SC and NN differs with the certainty factor q – the indicator that how reliable the agent's actions are. In Figure 6, we again see that when the uncertainty is low, both policies achieve about the same performance (possibly close to the optimum). If communication cost is zero or low, when uncertainty increases NN is much better than SC. But the time constraints play an very importance role here: in the left figure the SC line is close to the NN:5 line, but in the right figure (with a longer deadline) SC becomes better than NN:5. The underlying reason is that when agents have enough time to perform local recovery (as in SC) with 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 two 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. One of the future directions in developing the heuristics may be to combine the both policies and to develop a situation-specific policy.

5 Summary and Future Work

In this paper we defined a decentralized framework of a multi-agent MDP, described how communication and the cost of communication should be modelled into such a framework, and what is optimality in this framework. Although the optimality problem usually is computational prohibitive, approximation and heuristics exist and can give us very important insights into the problem of multi-agent coordination.

The study on 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 [12] in decentralized decision problems, we need to know if there exist special cases that dynamic programming is possible, and if there exists other efficient computation techniques that is suitable for multiagent MDP.

Also, learning may be a very importance approach in this problem. It is already proved to be a very good approach for traditional MDPs POMDPs, such as in [9], and may be a feasible solution for MMDP as well.

Finally, MMDP may be extended so that it covers infinte-horizon processes and also be 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 in a multi-agent agents. This would be very important when the system scales up.

Acknowledgement

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

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

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

[4] 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.

[5] 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.

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

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

[8] N. R. Sandell Jr., 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.

[9] 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.

[10] 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.

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

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