Computing effective communication policies in multiagent systems

Doran Chakraborty
doran@utulsa.edu

Sandip Sen
sandip@utulsa.edu

Mathematical & Computer Sciences Department
University of Tulsa
Tulsa, Oklahoma, USA

1. ABSTRACT

Communication is a key tool for facilitating multiagent coordination in cooperative and uncertain domains. We focus on a class of multiagent problems modeled as Decentralized Markov Decision Processes with Communication (DEC-MDP-COM) with local observability. The planning problem for computing the optimal communication strategy in such domains is often formulated with the assumption of the knowledge of optimal domain-level policy. Computing the optimal communication policy is NP-complete. There is a need, then, for heuristic solutions that trade-off performance with efficiency. We present a decision theoretic approach for computing optimal communication policies in stochastic environments which uses a branching future representation and evaluates only those decisions that an agent is likely to encounter. The communication strategy computed off-line is used in the more probable scenarios that the agent would face in future. Our approach also allows agents to compute communication policies at run-time in the unlikely event of the agents facing scenarios that were discarded while computing the off-line policy.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Performance

Keywords

communication, teamwork, multiagent planning

2. INTRODUCTION

Planning under cooperative settings have been studied extensively in the literature of multiagent systems. In this paper we develop an algorithm that tries to generate partial communication strategies based on system-level constraints and requirements. We decouple the planning problem of solving for optimal communication strategy from the problem of solving for optimal domain-level strategy. We assume that the agents know their optimal domain-level strategy. Even then the problem falls in a higher complexity class due to the large number of policies that need to be evaluated. Till date, research has focused on building communication strategies that try to solve the problem over all possible future uncertainties and therefore include evaluation of decisions on states that have a remote chance of occurrence. Our algorithm based on decision theoretic paradigms evaluates only those decisions that the agent has a high probability of facing in future and prunes off with an admissible heuristic, branches that have a low probability of occurrence. Needless to say, systems with higher computational capabilities would search deeper in the evaluation tree and come up with better communication strategies in comparison to systems which have computational constraints. Our algorithm incorporates this using a system level parameter that guides the evaluation process. Off-line communication strategies serve as guidelines for the more probable scenarios that the agent would face in future. Our approach however also involves agents computing run-time policies in case they face scenarios that were discarded while computing the off-line policy.

AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.

3. THE THEORETICAL FRAMEWORK

We consider evaluating communications decisions in the Dec_MDP_COM model[2] where the property of transitional independence and observational independence holds. It can be shown that given a Dec_MDP_COM with constant message cost, the value of the optimal joint policy with respect to any set of messages Σ^* cannot be greater than the value of the optimal joint policy with respect to the language of communications ($\Sigma = \Omega$)[2]. The goal of the set of agents is to maximize their expected reward over the finite horizon T. The decision making of each agent at each time step is divided into two parts. The first part is the communication step, where the agent decides whether to communicate and waits for communications from other agents. In the second part the agents decide on the domain-level action.

The policy of an agent i is given by the tuple $\pi^i =< \pi^i_a, \pi^i_c >$ where π^i_a is the domain-level action selection strategy and π^i_c is the communication strategy.

Domain-level action strategy (π_a^i): A local domain level action policy can be represented as a mapping of the last synchronized global state, the current local state for the agent and the time instant to a domain level action.

$$\pi_a^i : S \times S_i \times T \rightarrow A_i$$

Communication strategy (π_c^i): A local communication policy can be represented as a mapping of the last synchronized global state, the current local state for the agent, and the current time instant to either the current local state or σ_ϕ (does not communicate).

$$\pi_c^i : S \times S_i \times T \rightarrow S_i \cup \{\sigma_\phi\}$$

4. MEETING UNDER UNCERTAINTY

We use meeting under uncertainty as the domain of our research. The domain consists of a grid world where two agents have to meet starting at two initial positions in the grid domain. Due to locally full-observable property of our domain, the agents precisely identify their grid positions at all times. Each agent also knows the initial grid position of the other agent. The domain-level action set of each agent consists of $(Left, Right, Up, Down)$. The agent succeeds in moving to its intended grid position with some probability p and stays in its current position with probability $(1-p)$. At $T = 0$, the intended meeting position of the agents is decided as the mid-point of the starting grid positions of the two agents. However, since domain-level actions have stochastic outcomes, the agents may deviate from their intended path therefore requiring re-calculation of the meeting position and hence their domain-level action strategy.

5. NOAC

Communication in Dec_MDP_COM has been studied in details by Zilberstein and associates[1, 3]. We present a Near Optimal algorithm for communication ($NOAC$) that considers the scope of further communications in future while evaluating communication decisions for a particular state and time. To counter the computational overhead involved in computing complete communication strategies, $NOAC$ computes partial off-line policies that provide a communication decision for decision contexts that are more likely to occur in future. Each call to the $NOAC$ contains a probability estimate, p_r, which is the likelihood that the agent will face the corresponding decision in future. We approximate such branches with a appropriate heuristic function and prevent further recursive calls originating from that branch. A system-level factor P_c provides a lower-bound on the acceptable value of the likelihood of occurrence of a sub-branch for it to be expanded.

Algorithm 1: NOAC

```
begin
    input  : π_a^1, π_a^2, x_1, y_1, bs, t, p_r
    output: π_c^1, reward
1   π_c ← φ , π_c^com ← φ , r ← 0 , r^com ← 0 , bs^com ← φ
2   if t <= T then
3       for ∀ < x_2, y_2, prob > ∈ bs do
4           bs^com ← {(x_2, y_2, 1)}
5           if (x_1 = x_2) and (y_1 = y_2) then
6               r ← r + (prob × REWARD)
7               r^com ← r^com + (prob × REWARD)
8           else if (t = T) then
9               r ← r − (prob × 2 × STEP_COST)
                r^com ← r^com − (prob × 2 × STEP_COST)
10          else
11              π_a^{1^com} ← recompute action policy
12              for agent1 assuming com occurred;
13              π_a^{2^com} ← recompute action policy
14              for agent2 assuming com occurred;
15              if pr < P_c then
16                  r ← expected reward assuming
17                  no communication from here on
18                  with action policies π_a^1 and π_a^2
                    r^com ← expected reward assuming
19                  no communication from here on
20                  with action policies π_a^{1^com} and π_a^{2^com}
21              else
22                  < π_c', r' > ← COMPUTE − SUB −
                    POLICY(π_a^1, π_a^2, x_1, y_1, bs, t, p_r)
23                  r ← r + r' , π_c ← π_c ∪ π_c'
24                  < π_c', r' > ← COMPUTE − SUB −
                    POLICY(π_a^{1^com}, π_a^{2^com}, x_1, y_1, bs^com, t, p_r)
25                  r^com ← r^com + r' , π_c^com ← π_c^com ∪ π_c'
26      if (r^com − COM_COST) > r then
27          π_c^1 ← {(t, x_1, y_1, Communicate)} ∪ π_c^com
            reward ← (r^com − COM_COST)
28      else
29          π_c^1 ← {(t, x_1, y_1, Don't Communicate)} ∪ π_c
            reward ← r
end
```

6. RESULTS

We ran experiments to compare $NOAC$ with the $no-communication$ and $greedy$ approach [1, 3]. Results presented have been averaged over 100 runs. The agents incur a cost of 5 for every joint action and communication. They receive a high reward of 10000 if they meet. The experiment we ran was over a 7×7 grid world and for a horizon of $T = 9$. The agents start at grid positions $(0, 0)$ and $(6, 6)$ respectively. We use $P_c = (1-p)^2$. Table 1 shows a summary of the results. Under complete uncertainty i.e $p = 0.5$, the average reward generated by $NOAC$ is almost 4 times that of the $greedy$ case thus revealing the myopic behavior of the latter approach in computing efficient communication policies. Table 2 gives the number of computations done by $NOAC$ off-line and on-line. On an average $NOAC$ does 0.75 times less computations than the case for computing complete optimal off-line policies which is a significant reduction. Table 3 presents the effect on the average reward generated and the computational overhead for increasing values of P_c. We set $p = 0.6$. P_c is varied to account for 1, 2 and 3 transitional errors while calculating the off-line policy. For increasing values of P_c, the average reward tends to increase as the agents compute better communication policies but at the cost of higher computational overhead. For $P_c = 0.064$, the

Algorithm 2: COMPUTE-SUB-POLICY

begin
  input : $\pi_a^1, \pi_a^2, x_1, y_1, bs, t, p_r$
  output: π_c, r
1 $\pi_c \leftarrow \phi, r \leftarrow 0$
2 $a_1 \leftarrow$ *compute action for policy* π_a^1 *and grid* (x_1, y_1)
3 $(x_1', y_1') \leftarrow$ *update grid position for action* a_1;
4 $bs \leftarrow UPDATE-BELIEF(a_1, bs)$
5 **if** $(x_1 = x_1')$ *and* $(y_1 = y_1')$ **then**
6 $< \pi_c^{1^{T-t}}, r^{T-t} > \leftarrow SOLVE-FOR-OPTIMAL-POLICY(\pi_a^1, \pi_a^2, x_1', y_1', bs, (t+1), p_r)$
7 $\pi_c \leftarrow \pi_c^{1^{T-t}}$
8 $r \leftarrow r + (prob \times (r^{T-t} - 2 \times STEP\_COST))$
9 **else**
10 $< \pi_c^{1^{T-t'}}, r^{T-t'} > \leftarrow SOLVE-FOR-OPTIMAL-POLICY(\pi_a^1, \pi_a^2, x_1', y_1', bs, (t+1), (p \times p_r))$
11 $< \pi_c^{1^{T-t''}}, r^{T-step''} > \leftarrow SOLVE-FOR-OPTIMAL-POLICY(\pi_a^1, \pi_a^2, x_1, y_1, bs, (t+1), ((1-p) \times p_r))$
12 $\pi_c \leftarrow \pi_c^{1^{T-t'}} \cup \pi_c^{1^{T-t''}}$
13 $r \leftarrow r + (prob \times (p \times r^{T-t'} + (1-p) \times r^{T-t''} - 2 \times STEP\_COST))$
end

Algorithm 3: UPDATE-BELIEF

begin
  input : a, bs
  output: $updated - bs$
1 $updated - bs \leftarrow \phi$
2 **for** $\forall < x_2, y_2, prob > \in bs$ **do**
3 $< x_2', y_2' > \leftarrow$ *update grid position for successful execution of action* a
4 $updated - bs \leftarrow updated - bs \cup ((x_2', y_2', prob \times p))$
5 *Normalize* $updated - bs$
end

average reward generated is about 1.4 times that of $P_c = 0.4$ but this gain comes at an expense of twice as many number of computations. P_c can be used to trade-off between the two metrics and thus should be tuned based on the system constraints.

7. CONCLUSION

We studied the problem of computing near optimal communication policies in a DEC_MDP_COM domain. We argued that due to the high complexity of computing complete off-line optimal communication strategies, there is a need to distribute some of this load over run-time and derive near-optimal partial off-line strategies. Our approach makes the evaluation process more efficient by pruning unlikely branches in the evaluation tree. We substantiated our approach by results from experiments on *the meeting under uncertainty* domain and showed that our algorithm generates higher average rewards in comparison to existing heuristic techniques. As future work, we would like to extend $NOAC$ for domains with uncertainty over domain-level actions.

	$No-com$	$Greedy$		$NOAC$	
p	$Reward$	$Reward$	$comms$	$Reward$	$comms$
0.65	2365.1	3951.75	2.82	4952.85	2.58
0.6	2221.45	3751.45	2.74	4152.3	2.44
0.55	713.2	949.5	2.4	1283.01	2.67
0.5	103.25	283.25	2.1	1050.41	2.4

Table 1: Comparison between the rewards generated and number of communications occurred for different values of p for a 7×7 grid and $T = 9$.

$NOAC$ $Computations$			
p	$Off-line$	$On-line$	$\%Savings$
0.65	31028357	2439153.8	75
0.6	31028357	4358660.5	74
0.55	28691045	1801007.9	77

Table 2: Comparison between the computational savings compared to the optimal method for different values of p for a 7×7 grid and $T = 9$.

$NOAC$ $Computations$				
P_c	$Reward$	$Off-line$	$On-line$	$\%Savings$
0.4	2619.41	637445	3300102.3	92
0.16	2717.92	31028357	3593723.5	75
0.064	3619.08	70743085	620512.56	47

Table 3: Comparison between the reward generated and the computational savings for different values of P_c for a 7×7 grid and $T = 9$.

Acknowledgment: US National Science Foundation award IIS- 0209208 partially supported this work.

8. REFERENCES

[1] C. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems, 2003.

[2] C. Goldman and S. Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis, 2004.

[3] P. Xuan, V. Lesser, and S. Zilberstein. Communication decisions in multi-agent cooperation: model and experiments. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 616–623, New York, NY, USA, 2001. ACM Press.