Tightly and Loosely Coupled Decision Paradigms in Multiagent Expedition

Yang Xiang and Franklin Hanshar
University of Guelph
Ontario, Canada

Abstract

Frameworks for multiagent decision making may be divided into those where each agent is assigned a single variable (SVFs) and those where each agent carries an internal model, which can be further divided into loosely coupled frameworks (LCFs) and tightly coupled frameworks (TCFs). In TCFs, agent communication interfaces render their subdomains conditionally independent. In LCFs, either agents do not communicate or their messages are semantically less restricted. SVFs do not address the privacy issue well. LCF agents cannot draw from collective knowledge as well as TCF agents. However, disproportional effort has been dedicated to SVFs and LCFs, which can be attributed partially to unawareness of the computational advantages of TCFs over performance, efficiency and privacy. This work aims to provide empirical evidence of such advantages by comparing *recursive modeling method* (RMM) from LCFs and *collaborative design network* (CDN) from TCFs, both of which are decision-theoretic and the latter of which is a graphical model. We apply both to *multiagent expedition* (MAE), resolve technical issues encountered, and report our experimental evaluation.

1 Introduction

We consider frameworks for online decision making (rather than offline policy making, e.g., (Becker et al., 2004)) in cooperative multiagent systems. They may be divided into SVFs where each agent is assigned a single variable in the domain and those where each agent carries an internal model over a subdomain. Frameworks using internal models can be further divided into LCFs and TCFs. In TCFs, agents communicate through messages over agent interfaces that are semantically rigorously defined to render subdomains conditionally independent. In LCFs, either agents do not communicate but rely on observing other agents' actions to coordinate, or their messages are semantically less restricted.

SVFs do not address the issue of private versus public variables well, as they do not have infrastructure to differentiate variables as such. LCFs are often motivated by tasks where agents cannot communicate. Given the proliferation of distributed and wireless computing, it is hard to find task domains where cooperative agents cannot communicate (except a few of military nature). Due to tightly controlled agent interface, joint belief of team agents is well defined and a TCF agent's belief is consistent with the joint belief. This is generally not true in LCFs even when agents do communicate (see Proposition 4.5 and Theorem 8.10 in (Xiang, 2002) for a formal treatment). In other words, a TCF agent draws from collective knowledge better than a LCF agent in general. However, significant research efforts have been dedicated to SVFs, e.g., (Modi et al., 2005; Petcu and Faltings, 2005), and LCFs, e.g., (Gmytrasiewicz et al., 1998; Gmytrasiewicz and Durfee, 2001; Maes et al., 2001; Shen and Lesser, 2006), in comparison with those to TCFs (Xiang, 2002; Xiang et al., 2005). This can be attributed at least partly to unawareness of the computational advantages of TCFs over performance, efficiency and privacy. Hence, empirical evidence of such advantages will contribute to the due adoption of TCFs. In this work, we select one representative, RMM (Gmytrasiewicz et al., 1998), from LCFs and one, CDN (Xiang et al., 2005), from TCFs, both of which are decision-theoretic and the latter of which is a graphical model. We apply both to the same multiagent decision problem, MAE (Xiang and Hanshar, 2007), resolve technical issues encountered, especially those related to RMM, and compare them experimentally.

Sec. 2 reviews background on MAE, CDN and RMM. Sec. 3 presents technical issues on applying RMM to MAE. Sec. 4 reports experimental results. We discuss the generality issues of this research in Sec. 5.

2 Background

2.1 Multiagent Expedition

We consider MAE in an area represented as a grid of cells. At any cell, an agent can move to an adjacent cell by actions *north*, *south*, *east*, *west* or remain there (*halt*). The effect of an action is uncertain. The desirability of an object (located at a cell) is indicated by a numerical *reward*. A cell that is neither interesting nor harmful has a reward of a *base value*. The reward at a harmful cell is lower than the base value. The reward at an interesting cell is higher than the base value and can be further increased through agent cooperation.

When a physical object at a given location is to be manipulated (e.g., digging), cooperation is often most effective when a certain number of agents are involved, and the per-agent productivity is reduced with more or less agents. Suppose that the most effective level is 2. The reward that can be collected by a single agent from a given cell may be 0.3, and we denote this as a *unilateral reward*. If two agents cooperate at the cell, each receives 0.4, and we denote this as a *bilateral reward*. If three or more agents meet at the cell, two of them each receives 0.4 reward and others receive the base value. This feature promotes effective cooperations and discourages unproductive ones.

After a cell has been visited by any agent, its reward is decreased to the base value. As a result, wandering within a neighborhood is unproductive. Agents have no prior knowledge how rewards are distributed in the area. Instead, at any cell, an agent can reliably perceive the cell's absolute location (e.g., through GPS or triangulation) and reward distribution within a small radius. An agent can perceive the location and communicate with another agent if the latter is within a small radius.

The objective of the agents is to move around the area, cooperate as needed, and maximize the team reward over a finite horizon. They must do so based on local observations and limited communication.

2.2 Collaborative Design Networks

CDN is motivated by collaborative industrial design in supply chains. An agent responsible for a component encodes design knowledge and preference into a *design network* (DN) $S = (V, G, P)$. The *domain* is a set of discrete variables $V = D \cup T \cup M \cup U$. D is a set of *design parameters*. T is a set of *environmental factors* of the product under design. M is a set of objective *performance measures* and U is a set of subjective *utility functions* of the agent.

The dependence *structure* $G = (V, E)$ is a directed acyclic graph whose nodes are mapped to elements of V and whose set E of arcs encode design constraints, dependency of performance on design and environment, and dependency of utility on performance.

P is a set of potentials, one for each node x, formulated as a probability distribution $P(x|\pi(x))$, where $\pi(x)$ are parent nodes of x. $P(d|\pi(d))$, where $d \in D$, encodes a design constraint. $P(t|\pi(t))$ and $P(m|\pi(m))$, where $t \in T, m \in M$, are typical probability distributions. Each utility variable has a space $\{y, n\}$. $P(u = y|\pi(u))$ is a utility function $u(\pi(u)) \in [0, 1]$. Each node u is assigned a weight $k \in [0, 1]$ where $\sum_U k = 1$. With P thus defined, $\prod_{x \in V \setminus U} P(x|\pi(x))$ is a joint probability distribution (JPD) over $D \cup T \cup M$. With the assumption of additive independence among utility variables, the expected utility of a design $\mathbf{d}$ is $EU(\mathbf{d}) = \sum_i k_i(\sum_{\mathbf{m}} u_i(\mathbf{m})P(\mathbf{m}|\mathbf{d}))$, where $\mathbf{d}$ (bold) is a configuration of D, i indexes utility nodes in U, $\mathbf{m}$ (bold) is a configuration of parents of u_i, and k_i is the weight of u_i.

Each supplier is a designer of the supplied component. Agents, one per supplier, form a collaborative design system. Each agent embodies a design network called a design *subnet* and agents are organized into a *hypertree*: Each hypernode corresponds to an agent and its subnet. Each hyperlink (called *agent interface*) corresponds to design parameters shared by the two subnets, which renders them conditionally independent. They are *public* variables and remaining variables in each subnet are *private*. The hypertree specifies whom an agent can communicate directly. Each subnet is assigned a weight w_i, representing a compromise of preferences among agents, where $\sum_i w_i = 1$. The collection of subnets $\{S_i = (V_i, G_i, P_i)\}$ forms a CDN.

The product $\prod_{x \in V \setminus \cup_i U_i} P(x|\pi(x))$ is a JPD over $\cup_i(D_i \cup T_i \cup M_i)$, where $P(x|\pi(x))$ is associated with node x in a subnet. The expected utility of a design $\mathbf{d}$ is $EU(\mathbf{d}) = \sum_i w_i \left(\sum_j k_{ij} \left(\sum_{\mathbf{m}} u_{ij}(\mathbf{m}) P(\mathbf{m}|\mathbf{d})\right)\right)$, where $\mathbf{d}$ is a configuration of $\cup_i D_i$, i indexes subnets, j indexes utility nodes $\{u_{ij}\}$ in ith subnet, $\mathbf{m}$ is a configuration of parents of u_{ij}, and k_{ij} is the weight associated with u_{ij}. Hence, given a CDN, a decision-theoretical optimal design is well defined. Optimal design (Xiang et al., 2005) is obtained by communicating messages over agent interfaces along the hypertree. After communication, all agents have local designs that are globally optimal (collectively maximizing $EU(\mathbf{d})$). Computation (incl. communication) is linear on the number of agents (Xiang et al., 2005) and is efficient for a non-trivial class of CDNs (Xiang, 2007).

The general problem of MAE is exponentially complex on the number of agents and the length of horizon. A more efficient solution of MAE for limited horizon can be devised based on CDN (Xiang and Hanshar, 2007) (see Fig. 2 in Sec 4). An agent team is divided into groups. It allows group members to cooperate at the most effective level. At the same time, different groups can stay apart so that the area is explored more effectively and planning computation is made more efficient with less group interaction.

Within group, a hypertree organization is imposed to support tightly-coupled communication and reduce agent interaction. For instance, if the most productive level of cooperation is two, a group size of three and an organization $A - B - C$ for agents A, B and C can be used. In each agent subnet, movement actions form design nodes, agent locations form performance nodes, and rewards form utility nodes.

As mentioned above, planning is made more efficient by ignoring inter-group interaction and some inner-group interaction. The computation is sound only if unconsidered interactions do not exist. Such desirable behavior of agents is promoted through modifying the distribution of each utility node. The reward of a location is initialized to the perceived value. If the location is part of a group configuration where intended agent interactions are negatively affected (e.g., group members are too far apart) or unintended interactions are possible (e.g., members of different groups are too close), the reward value will be reduced. With grouping, planning complexity is unchanged as the team size grows.

2.3 Recursive Modeling Method

RMM (Gmytrasiewicz et al., 1998) uses a payoff matrix to encode an agent's preference over consequences of joint actions of team agents. With a total of n agents, a payoff matrix for an agent A has n dimensions, with one corresponding to each agent. The width of each dimension is equal to the number of alternative actions of the agent. Each cell of the matrix corresponds to the outcome of a joint action by all agents and is filled with the sum of rewards.

Agents do not communicate and reason about each other through a hierarchy of models (see Fig. 1 in Sec. 3.2). For instance, in a system with agents A and B, the top level model in A is its own payoff matrix. Each model in the second level represents what A believes to be the payoff matrix of B, assuming a specific state of B. The state is associated with a prior probability of A, and is updated through Bayesian learning (Gmytrasiewicz et al., 1998) based on observed actions of B.

3 Recursive Modeling for MAE

3.1 Payoff Matrix

Grouping, as proposed in CDN-based solution of MAE, was not a component in the original RMM. To allow a fair comparison, we apply grouping to RMM as well (otherwise, its computational cost would be much worse). This implies that additional measures in the CDN-based solution should also be applied to ensure soundness of group-based planning, e.g., reward adjustment. Let the group size be g and the length of planning horizon be k. The payoff matrix for each RMM agent has a dimension g, the width of each dimension is 5^k, and the total number of cells in the matrix is 5^{kg}.

Each cell is the payoff of the corresponding joint plan mv with k actions for each agent in the group G. Let the sequence of joint actions of agents in G be $mv = (mv^1,...,mv^k)$. Notation mv^i denotes the joint action at the ith step and consists of the ith action of each agent, i.e., $mv^i = \{mv_x^i | x \in G\}$. Let a resultant group trajectory be $t = (c^1, ..., c^k)$, where c^i is the group configuration after joint action mv^i. Configuration c^i consists of the position of each agent, i.e., $c^i = \{ps_x^i | x \in G\}$. The payoff can be computed as the expected group accumulative reward $erw_G(mv) = \sum_{y \in G}(\sum_t(P(t|mv)\sum_{i=1}^k rw_y(c^i)))$, where the second summation is over all possible group trajectories, and $rw_y(c^i)$ is the reward y receives at the group configuration c^i. Since the group configuration c^i is dependent only on the previous configuration c^{i-1} and joint action mv^i, we have $P(t|mv) = \prod_{i=1}^k P(c^i|c^{i-1}, mv^i)$, where $c^0 = null$. Furthermore, since the position of each agent x in group configuration c^i is dependent only on its own action mv_x^i and its own previous position ps_x^{i-1}, we have $P(c^i|c^{i-1}, mv^i) = \prod_{x \in G} P(ps_x^i|ps_x^{i-1}, mv_x^i)$. Combining the above, we have $erw_G(mv) =$

$$\sum_{y \in G}[\sum_t((\prod_{i=1}^k \prod_{x \in G} P(ps_x^i|ps_x^{i-1}, mv_x^i)) \cdot \sum_{i=1}^k rw_y(c^i))].$$

3.2 Recursive Model Structure

For an agent to use a payoff matrix to plan its actions, it needs the probability of each joint plan, determined by the likelihood of other agents' taking corresponding actions. We identify the key issue for agent B to predict actions of agent A as whether A will move closer to B for cooperation, which is determined by reward distribution around A. Since the reward distribution in the other side of A may be unobservable to B, and RMM agents do not communicate, the above probability must be computed by considering all possible cases of A's neighbourhood. We use recursive modeling as follows:

We characterize the unobservable neighborhood of A by whether it contains high unilateral reward cells. If so, A is more likely to move away from B. Otherwise, A is more likely to move towards B for the benefit of a cooperation. In particular, let nbp_x^y summarize unilateral rewards in neighborhood of agent y that is unobservable to agent x, where $nbp_x^y \in \{$**allLow**,¬**allLow**$\}$. If the unobservable area has at least one high reward, it is labeled ¬**allLow**. In general, there are $g-1$ unobservable neighborhoods one per group member, and they form 2^{g-1} possible cases. Each case forms a model at the second level of RMM tree, and is associated with x's belief $P(nbp_x^1, nbp_x^2, ..., nbp_x^{g-1})$. A two-level RMM tree is used in this work as knowledge at deeper levels cannot be reasonably assumed and deeper models have little effect on performance (Gmytrasiewicz et al., 1998). Fig. 1 shows a RMM tree with $g = 3$ and $k = 2$.

3.3 Bayesian Belief Update

As agents move around, agent x's belief $P(nbp_x^1, nbp_x^2, ..., nbp_x^{g-1})$ needs to be updated based on observations of other agents' last actions. Let lmv_x^y be the last move of agent y observed by x, where $lmv_x^y \in \{$**towards**, ¬**towards**$\}$ and **towards** means that y moved closer to x. To simplify discussion, we assume that $g = 3$, the group consists of agents A, B and C, and $x = B$. Hence, B needs

$$P(nbp^A, nbp^C|lmv^A, lmv^C), \quad (1)$$

we have omitted subscript B to aid readability.

It is difficult, if not impossible, to specify (1) directly as a joint probability of unobserved areas. What can be practically specified is the following as it refers to local dependencies:

$$P(nbp^A|lmv^A) \cdot P(nbp^C|lmv^C). \quad (2)$$

In general, (1) is not equivalent to (2). We show below assumptions needed to obtain (1)

A	B	C	u
(N,N)	(N,N)	(N,N)	0.85
(N,N)	(N,N)	(N,S)	0.50
	⋮		⋮
(H,H)	(H,H)	(H,W)	0.41
(H,H)	(H,H)	(H,H)	0.72

$P(nbp_B^A = \textbf{allLow}, nbp_B^C = \textbf{allLow})$

A	B	C	u
(N,N)	(N,N)	(N,N)	0.85
(N,N)	(N,N)	(N,S)	0.50
	⋮		⋮
(H,H)	(H,H)	(H,W)	0.41
(H,H)	(H,H)	(H,H)	0.72

$P(nbp_B^A = \textbf{allLow}, nbp_B^C = \neg\textbf{allLow})$

A	B	C	u
(N,N)	(N,N)	(N,N)	0.85
(N,N)	(N,N)	(N,S)	0.40
	⋮		⋮
(H,H)	(H,H)	(H,W)	0.41
(H,H)	(H,H)	(H,H)	0.72

$P(nbp_B^A = \neg\textbf{allLow}, nbp_B^C = \textbf{allLow})$

A	B	C	u
(N,N)	(N,N)	(N,N)	0.85
(N,N)	(N,N)	(N,S)	0.40
	⋮		⋮
(H,H)	(H,H)	(H,W)	0.41
(H,H)	(H,H)	(H,H)	0.72

$P(nbp_B^A = \neg\textbf{allLow}, nbp_B^C = \neg\textbf{allLow})$

A	B	C	u
(N,N)	(N,N)	(N,N)	0.85
(N,N)	(N,N)	(N,S)	0.70
	⋮		⋮
(H,H)	(H,H)	(H,W)	0.41
(H,H)	(H,H)	(H,H)	0.72

Figure 1: RMM tree of agent B. Payoff matrices shown in table format.

by computing (2). (1) can be rewritten as $P(nbp^A|nbp^C, lmv^A, lmv^C)P(nbp^C|lmv^A, lmv^C)$. If we assume that unobservable neighborhoods of A and C are conditionally independent given their movements, denoted $I(nbp^C, \{lmv^A, lmv^C\}, nbp^A)$, the above probability can be expressed as $P(nbp^A|lmv^A, lmv^C) \cdot P(nbp^C|lmv^A, lmv^C)$. With the additional assumptions $I(nbp^C, lmv^C, lmv^A)$ and $I(nbp^A, lmv^A, lmv^C)$, we obtain (2). Each factor in (2), say, $P(nbp^A|lmv^A)$, can be computed as $\frac{P(lmv^A|nbp^A)P(nbp^A)}{P(lmv^A),}$ where $P(nbp^A)$ is from the last belief update and $P(lmv^A)$ is a normalizing constant. $P(lmv^A|nbp^A)$ can be obtained by reasoning by case based on how rewards in the area between A and B are distributed. Let $m_B^A \in \{\textbf{allLow}, \neg\textbf{allLow}\}$ summarize rewards in this area. We have $P(lmv^A|nbp^A) = \sum_{m^A} P(lmv^A, m^A|nbp^A) = \sum_{m^A} P(lmv^A|m^A, nbp^A)P(m^A|nbp^A)$, where the first factor can be directly specified and the second factor can be estimated based on observed dependence between nearby rewards.

The above relies on the assumptions:

- $I(nbp^A, \{lmv^A, lmv^C\}, nbp^C)$,
- $I(nbp^A, lmv^A, lmv^C)$ and $I(nbp^C, lmv^C, lmv^A)$.

They often do not hold. For the first, when unobservable neighborhoods of A and C overlap, we have $nbp^A = nbp^C$ and the independence no longer holds. The second also fails in this situation since lmv^C is directly dependent on nbp^A. Requirement of these strong assumptions to make (1) computable in practice appears to be a limitation of the RMM framework.

4 Experimental Evaluation

We setup the environment such that the most productive level of cooperation is at two agents. The radius of agent perception and communication is 10 cells. Three types of environments of distinctive natures are simulated. In *Barren* type, each high reward cluster is less than 6×6 in size and is at least 20 cells away from any other high reward cluster. This type is useful to evaluate how well agents can avoid wandering in a low reward area and can migrate to locations with high reward. In *Dense* type, at least every 10×10 square of cells has a high reward cell. In *Path* type, high reward cells form a path and each high reward cell on the path has at least one other high reward cell within a distance of 2 cells.

We set up the CDN-based agent team with group size three, two groups per team, and planning horizon two. Agents, A, B and C, in a group are organized into a chain $A - B - C$. Subnets for A and B are shown in Fig. 2, where design, performance and utility nodes are shown as squares, ovals and diamonds, respectively.

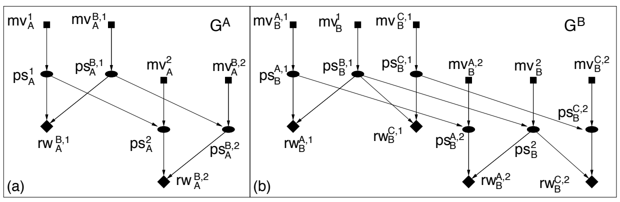


Figure 2: Subnets for group members A and B.

Each movement variable mv_x^i or $mv_x^{y,i}$ generally has 5 possible values denoting the five actions. Each position variable ps_x^1 or $ps_x^{y,1}$ has 5 possible values and each position variable ps_x^2 or $ps_x^{y,2}$ has 13 possible values. The conditional probability table (CPT) associated with a position node encodes uncertain dependency of the position on movement action. The node ps_x^2 is also dependent on the previous location ps_x^1. Utility node $rw_B^{A,i}$ represents rewards that agent

Table 1: Experimental results. Highest means bolded.

	Barren		Dense		Path	
	μ	σ	μ	σ	μ	σ
CDN	**55.84**	4.21	**25.14**	3.27	**20.41**	3.39
GRDU	48.56	0.56	12.32	0.20	12.20	0.15
GRDB	48.64	0.62	18.57	1.10	16.80	2.39
RMM	50.35	5.95	18.50	3.39	18.71	2.79

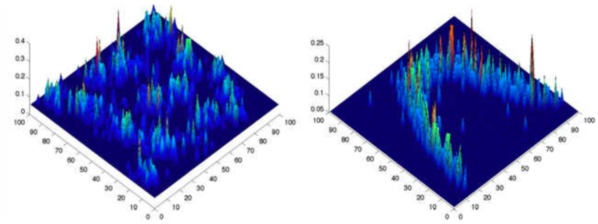
B receives due to cooperation (or lack of) with A. Its associated CPT encodes the reward as a utility distribution. We set up the RMM-based agent team with the same group size, team size and horizon. Both CDN and RMM teams use sophisticated reasoning. To evaluate its benefit, we also implemented two versions of simple greedy agents. One version (GRDU) is based on unilateral reward rwu and selects actions for agent x that maximize $\sum_{i=1}^{k} rwu(ps_x^i)$, where $rwu(ps_x^i)$ is the unilateral reward at the intended position of i'th action. Another version (GRDB) considers bilateral reward rwb as well and maximizes $\sum_{i=1}^{k}(rwu(ps_x^i)+rwb(ps_x^i))$. Each agent acts independently without communication. No group formation is applied as in RMM and CDN. For each version, we set the team size to six.

4.1 Performance Comparison

Tbl. 1 shows experimental performance of each agent team in different environments. For *Barren* type (base value 0.1), each team executes 40 time-steps (80 actions planned) in each run. For *Dense* (Fig. 3(a)) and *Path* (Fig. 3(b)) types (base value 0.05), each team executes 20 time-steps (40 actions planned) in each run. Each team performs 30 runs in each type of environment. The table gives the mean μ and standard deviation σ of the accumulative team reward.

CDN agents outperform other agents. The difference is significant at the 1% significance level when two-tailed t-test is used for all instances except *Path*, where CDN is better than RMM at the 5% significance level.

The *Path* type represents environments where all agents can perform well easily due to an abundance of clues. The *Barren* type represents those where all agents would perform poorly because of the lack of opportunities. The *Dense* type represents those where sound planning would best utilize the existing opportunities. Here the CDN shows the most gain in performance compared with alternatives.



(a) (b)

Figure 3: Environment types. High rewards are shown as taller peaks.

CDN agents outperform greedy agents since they coordinate actions to meet at high bilateral reward cells, whereas greedy agents have no coordination. CDN agents also outperform RMM agents, which can be attributed to two limitations of the latter. Firstly, estimation of neighborhood rewards of other agents through behavior observation and Bayesian update is inaccurate, which hinders effective cooperation. The second limitation is due to existence of multiple optimal joint plans. These joint plans promise the same maximal expected reward, but each agent must choose one in the plan. Without communication, each agent may commit to a different plan such that the resultant joint plan is sub-optimal. Note that this problem cannot be solved by social convention in a LCF as we show in the next section. CDN agents do not suffer from this problem as the interface between agents is composed of movement nodes, which explicitly communicates agent actions.

4.2 On Social Convention

A social convention defines, for each agent, without resorting to communication, the action to take when multiple optimal actions exist. We show that no such convention exists that guarantees collectively optimal actions in MAE. Consider S_1 in Fig. 4, where each cell is labelled with its coordinates, the occupying agent A, B or C, the cooperative per-agent reward b and unilateral reward $u < b$, and no agent can perceive beyond two cells. Let the convention be lexicographical, i.e., $goto(v, z) \succ goto(w, z)$

($\succ$ reads *is-preferred-over*) whenever rewards in cells (v, z) and (w, z) are identical but $v < w$. Hence, B would prefer $goto(2, 0)$ to meet A over $goto(4, 0)$ to meet C, because both actions have the same reward. A would prefer $goto(2, 0)$ to meet B and C would prefer $goto(4, 0)$ to meet B, since $b > u$. The joint action is optimal, though C is unable to meet B.

S_1 :	u	▶A	b, u	◀B	b, u	◀C	u
	$(0,0)$	$(1,0)$	$(2,0)$	$(3,0)$	$(4,0)$	$(5,0)$	$(6,0)$
S_2 :	u	▶A	b, u	◀B	b, u	◀C ▷	$\frac{b+u}{2}$

Figure 4: Two scenarios S_1 and S_2 for planning

Next consider S_2 in Fig. 4, where reward for $(6, 0)$ is slightly increased. Preferences of A and B do not change. C still prefers $goto(4, 0)$ to meet B since $b > \frac{b+u}{2}$. The joint action is suboptimal, as C would be better off with $goto(6, 0)$ as shown by the hollow arrow. Without communication, C has no way of knowing the reward at $(2, 0)$ and predicting B's action. Hence, social convention is incapable of coordinating agents with partial observations.

4.3 Efficiency Comparison

CDN, RMM and GRD teams (team size six) use 57, 16, and 0.8 seconds, respectively, for each round of planning. Below, we consider their scalability. Since greedy agents act independently, their efficiency is unaffected by the team size.

With grouping, each payoff matrix in a RMM agent has the size 5^{kg}, where g is the group size and k is the length of horizon. Hence, the space and time complexity of RMM planning grows exponentially with group size.

In comparison, each CDN-based agent group is organized into a hypertree. The necessity of the hypertree organization for exact multiagent probabilistic reasoning is formally established in (Xiang, 2002). The hypertree, together with required agent interfaces, are essential components of tight coupling and ensure optimal decision making in CDN. Hypertree organization also contributes to efficiency. It guarantees that the computational complexity of a CDN-based group is no worse than that of a RMM-based group in the worst case, and is more efficient when the CDN dependency structure is sparse. Fig. 5 shows a possible hypertree organization for MAE with g agents, where the subnet for agent A_2 is similar to that in Fig. 2 (b). The degree of any agent on the hypertree determines the number of agents whose interaction must be modeled and critically determines planning complexity of the agent. As long as this degree is bounded, complexity of computation at each group member does not increase with group size and complexity of planning only grows linearly with group size.

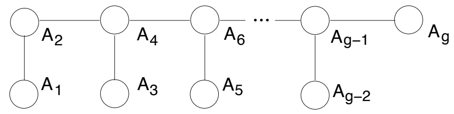


Figure 5: Possible hypertree structure for MAE

5 Discussion

This work is motivated by the disproportional research effort allocated among SVFs, LCFs and TCFs, which forms an odd contrast with the proliferation of distributed/wireless computing, societal emphasis on privacy, and theoretically established advantage of TCFs in utilizing collective knowledge. To improve awareness, we implemented CDN and RMM as representatives of TCFs and LCFs, respectively, in MAE, to allow experimental comparison. The application of RMM to MAE is a novel and nontrivial attempt. The study provided empirical evidence of advantages of CDN over RMM on performance and efficiency. Below we generalize this comparison to other domains and the advantage of CDN over SVFs on privacy.

At the modeling level, RMM and LCFs are limited by the need to model agent interactions without sufficient information. This is evidenced by the need for strong and often invalid assumptions in order to update belief on possible states of team agents (Sec. 3.3). This limitation also applies to communicative LCFs, e.g., (Gmytrasiewicz and Durfee, 2001). Agents in noncommunicative LCFs coordinate by observing other agents' actions. Since messages are speech acts, communicative LCFs are not fundamentally different. In contrast, CDN-based agents and TCFs in general do not suffer from this problem as agent interfaces are required to render agent subdomains conditionally independent. RMM is also limited by its matrix-based representation of exponential complexity. This can be remedied by adopting a graphical model in each agent as in MAID (Koller and Milch, 2001), although the above limitation stands.

At the decision making level, RMM and LCFs are limited by having to guess about the states and decisions of other agents based on observations. The inaccuracy in estimation can degrade agent performance through two distinct mechanisms: Firstly through misjudgement of other agents' states, which in turn leads to misjudgement of the optimal joint plan. Secondly, multiple optimal joint plans can degrade agent performance due to independent choice of agents. Social conventions cannot solve this problem as we have shown through a counterexample.

On the other hand, conditional independence rendering interfaces in TCFs convey sufficient states and decisions, resulting in improved coordination and superior performance. Compared to SVFs, an infra-structure exists within each CDN agent to differentiate variables into public and private. Only information on public variables are communicated through an agent interface. Private internal representations and preferences are *not* disclosed. Therefore, TCFs such as CDN provide superior performance, more efficient computation and a higher degree of privacy. Although a cost of communication must be paid (relative to non-communicative LCFs), since the communication is efficient (when the CDN is sparse), the price will be worthwhile for many applications. If communication is noisy/lost the performance degrades gracefully as agents can continue to work in smaller groups (see Sec. 8.9 in (Xiang, 2002)).

Regarding the generality of this work, we draw attention to key features of CDN and RMM. Both are decision-theoretic. RMM is proposed as a general framework for cooperative multiagent decision making. CDN is proposed in the context of collaborative design, but is in fact a general framework, whose applicability to MAE, a domain very different from design, is a clear indication. The generality of CDN and RMM and their common decision-theoretic foundation point to the source of difference in their experimental evaluation, i.e., their difference in agent coupling, and promise that our empirical results in MAE are generalizable.

References

R. Becker, S. Zilberstein, V. Lesser, and C.V. Goldman. 2004. Solving transition independent decentralized Markov decision processes. *J. Artificial Intelligence Research*, 22:423–455.

P.J. Gmytrasiewicz and E.H. Durfee. 2001. Rational communication in multi-agent enviroments. *Auto. Agents and Multi-Agent Systems*, 4(3):233–272.

P.J. Gmytrasiewicz, S. Noh, and T. Kellogg. 1998. Bayesian update of recursive agent models. *User Modeling and User-Adapted Interaction*, 8(1):49–69.

D. Koller and B. Milch. 2001. Multi-Agent Influence Diagrams for Representing and Solving Games. In *Proc. 17th Inter. Joint Conf. on Artificial Intelligence*, pages 1027–1034.

S. Maes, K. Tuyls, and B. Manderick. 2001. Modeling a multi-agent enviroment combining influence diagrams. In M. Mohammadian, editor, *Proc. of IAWTIC*, pages 379–384.

P.J. Modi, W. Shen, M. Tambe, and M. Yokoo. 2005. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149–180.

A. Petcu and B. Faltings. 2005. A scalable method for multiagent constraint optimization. In *Proc. 19th Inter. Joint Conf. on Artificial Intelligence*, pages 266–271.

J. Shen and V. Lesser. 2006. Communication management using abstraction in distributed Bayesian networks. In *Proc. 5th Inter. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 622–629.

Y. Xiang and F. Hanshar, 2007. *Advances in Artificial Intelligence, LNCS*, volume 4509, chapter Planning in Multiagent Expedition with Collaborative Design Networks, pages 526–538. Springer-Verlag.

Y. Xiang, J. Chen, and W.S. Havens. 2005. Optimal design in collaborative design network. In *Proc. 4th Int. Joint Conf. on Auto. Agents and Multiagent Systems*, pages 241–248. ACM.

Y. Xiang. 2002. *Probabilistic Reasoning in Multiagent Systems: A graphical models approach*. Cambridge University Press.

Y. Xiang. 2007. Tractable optimal multiagent collaborative design. In *Proc. IEEE/WIC/ACM Inter. Conf. on Intelligent Agent Technology*, pages 257–260.