Multiagent Decision by Partial Evaluation

Yang Xiang and Frank Hanshar

University of Guelph, Canada

Abstract. We consider multiagent cooperative decision in stochastic environments, and focus on online decision during which agents communicate. We generalize partial evaluation from a specific application to a class of collaborative decision networks (CDNs), and propose a distributed decision algorithm based on partial evaluation. We show that when agents have private decision variables, the new algorithm can significantly speed up decision in comparison with the earlier CDN algorithm.

1 Introduction

We consider a class of multiagent cooperative decision problems in partially observable and stochastic environments. One example is collaborative design in supply chain [5]. Another example is multiagent expedition (MAE) [7]. We assume that the decision problems can be modeled as CDNs [6, 7], a class of decision-theoretic, cooperative, multiagent, graphical models. Main assumptions of CDNs are the following: Application environment (*env*, a set of variables) is decomposed into overlapping sub-environments (subenvs), that can be organized into a hypertree with running intersection property. Each subenv is hosted by an agent A and consists of decision variables, effect variables, and utility variables. Dependency among variables in each subenv is modeled as a decision subnet (influence diagram). Overlaps of subenvs (agent interfaces) consist of decision variables only.

We focus on online decision making over short horizons rather than offline policy making, e.g., [4, 2]. That is, agents decide the best joint action based on what they know about the current env. Focus on current env state, rather than all possible trajectories as in offline policy making, allows much more efficient decision making.

An agent makes a *simultaneous decision* [10] when it decides its actions over multiple decision variables, all of which have the horizon length $h = 1$. A collection of decisions one from each agent is a *joint* decision. This work concerns joint, simultaneous decisions. An example is collaborative design in supply chain, where the component design at each agent is a simultaneous decision and the product design is a joint, simultaneous decision. This work concerns also *sequential decision*, where agents decide and act over time, e.g., in MAE. Agents plan at every time step with $h \geq 1$ and act according to joint action for time $t = 1$.

We term each choice of a decision variable as an *action*, e.g., choosing $2GB$ RAM for a device under design. For multiple decision variables (possibly for different time), we refer to a vector of choices, one for each variable, as a *plan*. If these variables are contained in a single subenv, the plan is *local*, otherwise, it is *joint*. Hence, a component design in collaborative design is a local plan, and so is a sequence of $h > 1$ movements by a MAE agent. A product design is a joint plan, and so is a collection of local plans of a team of MAE agents. The decision task is to obtain an optimal joint plan by distributed online computation.

The above decision problem is equivalent to a Dec-POMDP [1, 8], and hence its optimal solution is generally intractable. During online decision, our agents communicate, as in e.g. [3], but *not* local observations. For our decision algorithm, we analyze its communication cost, while our agents do not during decision making.

To improve efficiency in MAE applications, an idea of *partial evaluation* was explored in [9]. Significant speedup was achieved in centralized decision, while application to distributed decision was not productive. This work generalizes partial evaluation to a class of CDNs, where each directed path starts from a decision node, followed by an effect node, and ends by a utility node. We extend the earlier algorithm for CDNs [6] based on partial evaluation, and show that when subenvs contain private decisions the new algorithm can significantly speed up decision making.

Section 2 introduces CDNs. Sections 3 through 5 present partial evaluation in subnets of increasing sophistication. The proposed algorithm is presented in Sections 6 through 8, with performance demonstrated in Section 9.

2 Background

A set $\mathscr{A}$ of n agents populates an env. It is represented as a set $\mathscr{V}$ of variables, decomposed into overlapping subenvs V_1, ..., V_n, each hosted by an agent. The decomposition allows the construction of a *hypertree* $\mathscr{H}$ whose (hyper)nodes are labeled by subenvs such that intersection of every two nodes is contained in each node on the path between the two (running intersection).

Each subenv $V = D \cup E \cup U$, where D, E, U are disjoint, is modeled by an agent $A \in \mathscr{A}$ into a decision subnet $S = (D,E,U,G,P,T)$, where G is an acyclic directed graph whose nodes are labeled by elements of V. D is a set of local decision variables $D = \{d_1, d_2, ...\}$, and we denote $\rho = |D|$. Each d_i has a finite space of options or actions $Op_i = \{d_{i1}, d_{i2}, ...\}$. Denote $\sigma = \max_i |Op_i|$.

E is a set of effect variables $E = \{e_1, e_2, ...\}$, representing outcomes of actions. Each e_i has a finite space $Ef_i = \{e_{i1}, e_{i2}, ...\}$. Denote $\kappa = \max_j |Ef_j|$. Each e_i has a set $\delta_i \subseteq D$ of decision variables as its parents in G. The dependency of e_i on δ_i is quantified by a conditional probability table (CPT) $P(e_i|\delta_i)$. P is the set of CPTs, one for each $e_i \in E$.

U is a set of utility variables $U = \{u_1, u_2, ...\}$, encoding the subjective preference of A over effects, and we denote $\eta = |U|$. Each $u_i \in U$ has a set $\pi_i \subseteq E$ of effects as parents in G. Denote $m = \max_i |\pi_i|$. Preference over π_i is encoded in a utility function $u_i(\pi_i) \in [0,1]$, and T is a set of functions, one for each $u_i \in U$. Each u_i is associated with a weight $w_i > 0$ such that $\sum_i w_i = 1$.

If subenv V hosted by agent A is adjacent on the hypertree to subenv V' hosted by A', then the (hyper)link labeled $V \cap V'$ is the *interface* of A and A'. For A, the interface is *public*, and all other variables in V are *private* relative to A'.

A CDN is a tuple $(\mathscr{A}, \mathscr{V}, \mathscr{H}, \mathscr{S})$, where $\mathscr{S}$ is a set of decision subnets, one for each $A \in \mathscr{A}$. An example is in Fig. 1, where nodes in D, E, U are drawn as squares, ovals, diamonds, respectively. Defined as above, every directed path in a subnet has a length 2, e.g., Fig. 1 (a). This work focuses on such CDNs. More general subnets can all be converted equivalently to length-2 (with modification to relevant CPTs and utility functions), and we assume that such is done.

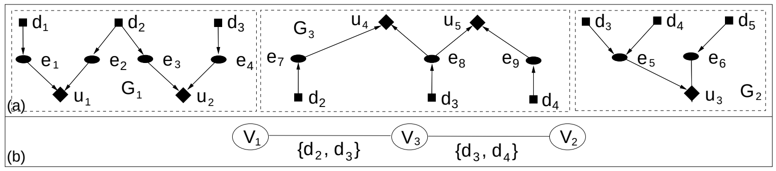

Fig. 1. (a) CDN subnets. (b) Hypertree.

3 Single Decision Variable

Consider a decision subnet where $\rho = 1$, $\eta = 1$, and $V = \{d_i, e_i, u_i\}$. Expected utility of taking an action $d_i = d_{ij}$ is

$$eu(d_{ij}) = P(e_{i1}|d_{ij})u_i(e_{i1}) + P(e_{i2}|d_{ij})u_i(e_{i2}) + ...,$$

which requires κ probability retrievals, κ utility retrievals, κ multiplications, and $\kappa - 1$ additions. Action d_{ij} is *fully evaluated* when $eu(d_{ij})$ is computed. Its complexity is $O(\kappa)$. Agent's objective is to compute the optimal decision pair $(d^*_{ik}, meuv)$ such that $eu(d^*_{ik}) = meuv = \max_j eu(d_{ij})$. This generally involves a full evaluation of each d_{ij}, with the complexity $O(\sigma\ \kappa)$.

For every action, we refer to an effect with the highest probability (indexed once and referenced repeatedly) as the *pivot effect*, and break ties arbitrarily. Let e_{ik} be the pivot effect of action d_{ik}. Then $eu(d_{ik}) \leq P(e_{ik}|d_{ik})u_i(e_{ik}) + (1 - P(e_{ik}|d_{ik}))u_i^{max} \equiv Q_{ik}$, where $u_i^{max} = \max_x u_i(e_{ix})$. We say that d_{ij} dominates d_{ik} iff $eu(d_{ij}) > eu(d_{ik})$. If $Q_{ik} < eu(d_{ij})$, then $eu(d_{ik}) < eu(d_{ij})$ and d_{ik} is dominated by d_{ij}. Computing Q_{ik} requires one probability retrieval, one utility retrieval, two multiplications, and two additions. Action d_{ik} is *partially evaluated* when Q_{ik} is computed.

Suppose probabilities of pivot effects for d_i are nearly identical. In MAE, a movement decision has alternatives *north, east*, etc. Moving north most likely lands on north location. Its probability is about the same as that of landing on east location if moving east. In many envs, each action most likely produces a given effect, with smaller probabilities to produce other effects. As another example, a message may be sent by email or post. Emails most likely arrive in seconds while post mails most likely in days. Formally, we assume

$$\forall j,k\ P(e_{ij}|d_{ij}) = P(e_{ik}|d_{ik}) = p. \tag{1}$$

Partial evaluation of action d_{ik} amounts to compute $p\ u_i(e_{ik}) + (1-p)u_i^{max}$. To determine whether

$$p\ u_i(e_{ik}) + (1-p)u_i^{max} < eu(d_{ij}) \tag{2}$$

holds, we check instead whether $u_i(e_{ik}) < \frac{eu(d_{ij})}{p} - \frac{1-p}{p}u_i^{max}$ holds. We assume that the right hand side (threshold) has been obtained before d_{ik} is evaluated. Then if the above inequality holds, d_{ik} can be rejected with just one utility retrieval and one comparison, and the threshold can be reused for evaluating the next action. In this case, partial evaluation of d_{ik} has a complexity of $O(1)$.

This leads to the following *partial evaluation based decision* to obtain $(d^*_{ik}, meuv)$: Apply full evaluation for the first action to establish a threshold. For each alternative action, apply partial evaluation to reject if warranted. Otherwise, apply full evaluation to it and update the threshold. The last option accepted is d^*_{ik}.

Using this method, efficiency is gained by evaluating a d_{ik} fully only if the above inequality fails. Let $\theta \in [0,1]$ be the percentage of d_{ik} fully evaluated. Then the complexity of partial evaluation based decision is $O(\theta\ \sigma\ \kappa + (1-\theta)\ \sigma)$.

4 Multiple Decision Variables

Consider the case where $\rho > 1$ and $\eta = 1$, e.g., subnet G_2 in Fig. 1 (a), where each local plan is to be evaluated. Let $\overline{d}$ be a local plan over D, $\overline{e}$ be its (compound) pivot effect (made of pivot effect of each action in $\overline{d}$), $\overline{e''}$ be any (compound) effect, and $eu(\overline{d})$ be the expected utility of $\overline{d}$. Then

$$eu(\overline{d}) = P(\overline{e}|\overline{d})u_i(\overline{e}) + \sum_{\overline{e''}} P(\overline{e''}|\overline{d})u_i(\overline{e''}). \tag{3}$$

Let β be a subset of D and $\gamma = D \setminus \beta$. We consider the decision problem to obtain a pair of functions $(meu(\beta), peer(\beta))$, where $meu : \beta \to [0,1]$ and $peer : \beta \to \gamma$, such that for each plan $\overline{b}$ over β, $meu(\overline{b}) = \max_{\overline{y}} eu(\overline{b},\overline{y})$ *and* $eu(\overline{b}, peer(\overline{b})) = meu(\overline{b})$, where $\overline{y}$ is a plan over γ, $(\overline{b},\overline{y})$ denotes a *join* of plans, maximization is over each plan $\overline{y}$, and $peer(\overline{b})$ equals the optimal plan $\overline{y^*}$. We refer to β as *constraint scope* and γ as *optimization scope*. In other words, $(meu(\beta), peer(\beta))$ specifies, for each constraint $\overline{b}$, the MEU $meu(\overline{b})$ and the corresponding optimal plan $\overline{d^*} = (\overline{b}, peer(\overline{b}) = \overline{y^*})$.
The task requires evaluating $eu(\overline{b},\overline{y})$. Denote plan *projection* by $\overline{b} = proj(\overline{d}, \beta)$, $\overline{y} = proj(\overline{d}, \gamma)$, and write $\overline{d} = (\overline{b},\overline{y})$. To evaluate $eu(\overline{d})$ by Eqn. (3), $P(\overline{e''}|\overline{d})$ must be computed for each $\overline{e''}$. For G_2 in Fig. 1 (a), $P(\overline{e''}|\overline{d}) = P(e_5, e_6|d_3, d_4, d_5) = P(e_5|d_3, d_4)P(e_6|d_5)$. In general, computation of $P(\overline{e''}|\overline{d})$ for each $\overline{e''}$ involves m probability retrievals and $m-1$ multiplications. Hence, a full evaluation of $\overline{d}$ by Eqn. (3) takes $m\ \kappa^m$ probability retrievals, κ^m utility retrievals, $m\ \kappa^m$ multiplications, and $\kappa^m - 1$ additions. Complexity of full evaluation of $\overline{d}$ is thus $O(m\ \kappa^m)$. To obtain $meu(\beta)$ by full evaluation, a total of σ^ρ alternative $\overline{d}$ must be evaluated, and the complexity is thus $O(\sigma^\rho\ m\ \kappa^m)$.
For more efficient decision, we observe

$$\begin{aligned} eu(\overline{d}) &= P(\overline{e}|\overline{d})u_i(\overline{e}) + \sum_{\overline{e''}} P(\overline{e''}|\overline{d})u_i(\overline{e''}) \le P(\overline{e}|\overline{d})u_i(\overline{e}) + \sum_{\overline{e''}} P(\overline{e''}|\overline{d})u_i^{max} \quad (4)\\ &= P(\overline{e}|\overline{d})u_i(\overline{e}) + (1 - P(\overline{e}|\overline{d}))u_i^{max}, \quad \textit{where}\ \ u_i^{max} = \max_{\overline{e''}} u_i(\overline{e''}). \end{aligned}$$

Let e_i be an effect variable with parent set δ_i. Let $\overline{\delta_{ik}}$ be kth configuration of δ_i and e_{ik} be its pivot effect. Assume

$$\forall\ i, j, k\ \ P(e_{ij}|\overline{\delta_{ij}}) = P(e_{ik}|\overline{\delta_{ik}}). \tag{5}$$

That is, for each effect variable and its influencing decision variables, probabilities of pivot effects are approximately identical. It then follows that $\forall \overline{d}, \overline{d'}\ P(\overline{e}|\overline{d}) = P(\overline{e'}|\overline{d'}) = p$, where $\overline{e'}$ is the pivot effect of local plan $\overline{d'}$, and p can be computed once for all pivot effects. For the example on G_2 in Fig. 1, suppose probabilities of pivot effects for $P(e_5|d_3, d_4)$ is 0.7 and those for $P(e_6|d_5)$ is 0.9. Then we have $p = 0.63$. Now, inequation (4) becomes $eu(\overline{d}) \le p\ u_i(\overline{e}) + (1-p)u_i^{max}$.
In general, given $eu(\overline{d'})$ and an alternative plan $\overline{d}$, if

$$p\ u_i(\overline{e}) + (1-p)u_i^{max} < eu(\overline{d'}), \tag{6}$$

it follows that $eu(\overline{d}) < eu(\overline{d'})$ and $\overline{d}$ is dominated by $\overline{d'}$. Partial evaluation of $\overline{d}$ by Eqn. (6) takes only one utility retrieval, with complexity $O(1)$.
Extending partial evaluation based decision in Section 3 with the above, the decision problem to obtain $(meu(\beta), peer(\beta))$ can be solved by PEUtilDec1:

Algorithm 1 *PEUtilDec1*
Input: subnet over $(D,E,U=\{u_i\})$*; max utility* u_i^{max}*; pivot probability* p*;*
constraint scope $\beta \subset D$ *and optimization scope* $\gamma = D \setminus \beta$*;*

for each constraint $\overline{b}$ *over* β*,*
  pick a plan $\overline{y'}$ *over* γ *to fully evaluate by Eqn. (3); get* $eu(\overline{b},\overline{y'})$*;*
  set $meu(\overline{b}) = eu(\overline{b},\overline{y'})$ *and* $peer(\overline{b}) = \overline{y'}$*;*
  set threshold $th = (eu(\overline{b},\overline{y'}) - (1-p)u_i^{max})/p$*;*
  for each other plan $\overline{y}$ *over* γ*,*
    retrieve $u_i(\overline{e})$ *for pivot effect* $\overline{e}$ *of* $\overline{d} = (\overline{b},\overline{y})$*;*
    if $u_i(\overline{e}) \geq th$*,*
      fully evaluate $\overline{d}$ *by Eqn. (3) to get* $eu(\overline{d})$*;*
      $meu(\overline{b}) = eu(\overline{d})$*,* $peer(\overline{b}) = \overline{y}$*;* $th = (eu(\overline{d}) - (1-p)u_i^{max})/p$*;*
return $(meu(\beta), peer(\beta))$*;*

When $\beta = \emptyset$, in the return value of PEUtilDec1, $meu(\beta)$ becomes a single value and $peer(\beta)$ becomes a single optimal plan $\overline{d^*}$ over D. Let $\theta \in [0,1]$ be the percentage of $\overline{d}$ fully evaluated by PEUtilDec1. Its complexity is then $O(\theta\ \sigma^{\rho}\ m\ \kappa^m + (1-\theta)\ \sigma^{\rho}) = O(\theta\ \sigma^{\rho}\ m\ \kappa^m)$. Proposition 1 summarizes key properties of PEUtilDec1.

Proposition 1 *PEUtilDec1 satisfies the following: (1) For each constraint* $\overline{b}$ *over* β*,* $meu(\overline{b})$ *is the MEU. (2) For each* $\overline{b}$*,* $peer(\overline{b})$ *is the optimal plan over* γ*. (3) Its complexity is* $O(\theta\ \sigma^{\rho}\ m\ \kappa^m)$*.*

5 Multiple Utility Variables

Next, consider a decision subnet where $\rho > 1$ and $\eta > 1$. An example is G_1 in Fig. 1 (a), where $\eta = 2$ and a weight w_i is associated with each u_i ($i = 1,2$). The optimal local plan cannot be obtained by solving independent sub-problems over $\{d_1,d_2\}$ and over $\{d_2,d_3\}$. Hence, fully evaluating a local plan amounts to compute

$$eu(d_1,d_2,d_3) = \\ w_1 \sum_{e_1,e_2} P(e_1|d_1)P(e_2|d_2)u_1(e_1,e_2) + w_2 \sum_{e_3,e_4} P(e_3|d_2)P(e_4|d_3)u_2(e_3,e_4).$$

In general, let $\overline{d}$ be a local plan over D, $\overline{e}$ be its pivot effect, $\overline{e''}$ be any alternative effect, and $eu(\overline{d})$ be expected utility of $\overline{d}$. For each u_i ($i = 1,...,\eta$) with parents π_i, let α_i be the set of decision ancestors of u_i. In G_1 of Fig. 1 (a), α_1 for u_1 is $\{d_1,d_2\}$. Define $\overline{e_i} = proj(\overline{e},\pi_i)$, and $\overline{d_i} = proj(\overline{d},\alpha_i)$. Then, a full evaluation of $\overline{d}$ computes

$$eu(\overline{d}) = \sum_{i=1}^{\eta} w_i [P(\overline{e_i}|\overline{d_i})u_i(\overline{e_i}) + \sum_{e''_i} P(\overline{e''_i}|\overline{d_i})u_i(\overline{e''_i})]. \quad (7)$$

We consider the decision problem to obtain $(meu(\beta), peer(\beta))$ such that for each plan $\overline{b}$ over β, $meu(\overline{b}) = \max_{\overline{y}} eu(\overline{b},\overline{y})$ and $eu(\overline{b}, peer(\overline{b})) = meu(\overline{b})$, where $\overline{y}$ is a plan over γ, and $peer(\overline{b})$ equals optimal plan $\overline{y^*}$.

Extending result from Section 4, the complexity to fully evaluate $\overline{d}$ is $O(\eta\ m\ \kappa^m)$. We consider partial evaluation with the above example: If e_j is the child node of d_i, we denote its pivot effect corresponding to d_{ik} by e_{jk}. We have

$$\begin{aligned} & eu(d_{1x}, d_{2y}, d_{3z}) \leq \\ & \quad w_1 P(e_{1x}|d_{1x}) P(e_{2y}|d_{2y}) u_1(e_{1x}, e_{2y}) + w_2 P(e_{3y}|d_{2y}) P(e_{4z}|d_{3z}) u_2(e_{3y}, e_{4z}) \\ & \quad + w_1(1 - P(e_{1x}|d_{1x}) P(e_{2y}|d_{2y})) u_1^{max} + w_2(1 - P(e_{3y}|d_{2y}) P(e_{4z}|d_{3z})) u_2^{max}. \end{aligned}$$

Assuming probabilities of pivot effects for each effect variable e_i is identical, denoted by p_i, we have $eu(d_{1x}, d_{2y}, d_{3z}) \leq w_1 p_1 p_2 u_1(e_{1x}, e_{2y}) + w_2 p_3 p_4 u_2(e_{3y}, e_{4z}) + w_1(1 - p_1 p_2) u_1^{max} + w_2(1 - p_3 p_4) u_2^{max}$.
Given $eu(d'_{1x}, d'_{2y}, d'_{3z})$ and another plan (d_{1x}, d_{2y}, d_{3z}), if

$$\begin{aligned} & w_1 p_1 p_2 u_1(e_{1x}, e_{2y}) + w_2 p_3 p_4 u_2(e_{3y}, e_{4z}) + w_1(1 - p_1 p_2) u_1^{max} \\ & \qquad + w_2(1 - p_3 p_4) u_2^{max} < eu(d'_{1x}, d'_{2y}, d'_{3z}), \end{aligned} \tag{8}$$

holds, it follows that $eu(d_{1x}, d_{2y}, d_{3z}) < eu(d'_{1x}, d'_{2y}, d'_{3z})$, and (d_{1x}, d_{2y}, d_{3z}) is dominated by $(d'_{1x}, d'_{2y}, d'_{3z})$. Partial evaluation of (d_{1x}, d_{2y}, d_{3z}) by Eqn. (8) takes only two utility retrievals.
In general, each utility variable u_i has π_i effect parents, each of which is associated with a pivot probability p_{ij} indexed by j. Given $eu(\overline{d'})$ from a full evaluation, and an alternative plan $\overline{d}$, if

$$\sum_{i=1}^{\eta} w_i \left((\prod_j p_{ij})\ u_i(\overline{e_i}) + (1 - (\prod_j p_{ij}))\ u_i^{max} \right) < eu(\overline{d'}),$$

then $\overline{d}$ is dominated by $\overline{d'}$. This leads to threshold and dominance comparison below:

$$th = eu(\overline{d'}) - \sum_{i=1}^{\eta} w_i (1 - (\prod_j p_{ij}))\ u_i^{max}, \tag{9}$$

$$\sum_{i=1}^{\eta} w_i (\prod_j p_{ij})\ u_i(\overline{e_i}) < th. \tag{10}$$

By pre-computing $w_i(\prod_j p_{ij})$ for each i, partial evaluation of $\overline{d}$ has a complexity of $O(\eta)$. Extending PEUtilDec1, PEUtilDec2 makes partial evaluation based decision with multiple decision and utility variables, whose properties are stated below.

Algorithm 2 *PEUtilDec2*
Input: subnet over (D,E,U)*; max utility* u_i^{max} *for each* u_i*;*
for each u_i *and each of its parent, a pivot probability* p_{ij}*;* $\beta \subset D$ *and* $\gamma = D \setminus \beta$*;*
for each constraint $\overline{b}$ *over* β*,*
  pick a plan $\overline{y'}$ *over* γ *and denote* $\overline{d'} = (\overline{b}, \overline{y'})$*;*
  fully evaluate $\overline{d'}$ *by Eqn. (7) to get* $eu(\overline{d'})$*;*
  set $meu(\overline{b}) = eu(\overline{d'})$ *and* $peer(\overline{b}) = \overline{y'}$*; set threshold* th *by Eqn. (9);*
  for each other plan $\overline{y}$ *over* γ *and* $\overline{d} = (\overline{b}, \overline{y})$*,*
    perform comparison by Eqn. (10);
    if comparison fails,
      fully evaluate $\overline{d}$ *by Eqn. (7) to get* $eu(\overline{d})$*;*
      $meu(\overline{b}) = eu(\overline{d})$*,* $peer(\overline{b}) = \overline{y}$*, and update* th *by Eqn. (9);*
return $(meu(\beta), peer(\beta))$*;*

Proposition 2 *PEUtilDec2 satisfies (1) and (2) of Proposition 1, as well as the following: Its complexity is* $O(\theta\ \sigma^{\rho}\ \eta\ m\ \kappa^{m})$.

Note that, when $\beta = \emptyset$, the return value of PEUtilDec2, $meu(\beta)$, becomes a single value, and $peer(\beta)$ becomes a single optimal plan $\overline{d^*}$ over D.

6 Utility Message by Leaf Agent

We obtain an optimal joint plan with two rounds of message passing, using PEUtilDec1 and PEUtilDec2. Hypertree is directed from an arbitrary root and agents become parent-child according to the direction. In first round, utility messages flow from leaf agents towards root. Message from a leaf agent is computed as follows.

Let the subnet of a leaf agent B be over (D,E,U), and its interface with the adjacent agent C on hypertree be $SD \subset D$. Message $utm_0(SD)$ that B sends to C is a MEU function that, for each local plan $\overline{sd}$ over SD, specifies

$$utm_0(\overline{sd}) = \max_{\overline{rd}} eu(\overline{sd}, \overline{rd}), \tag{11}$$

where $\overline{rd}$ is a local plan over $RD = D \setminus SD$. From Proposition 2, if B applies PEUtilDec2 with $\beta = SD$, the return value $meu(\beta)$ satisfies $utm_0(SD) = meu(\beta)$. Since this computation is exponential on ρ, we seek to improve its efficiency below.

For each utility u_i and its α_i, we define $\beta_i = \alpha_i \cap SD$ *and* $\gamma_i = \alpha_i \setminus \beta_i$. That is, β_i is public and $\gamma_i \subset RD$ is private relative to C. Utility variables can then be classified into four exhaustive and mutually exclusive cases:

1. $\gamma_i = \emptyset$. That is, $\alpha_i = \beta_i$ is entirely public.
2. $\gamma_i \neq \emptyset$, $\beta_i \neq \emptyset$, and $\gamma_i \cap \gamma_j = \emptyset$ for all $j \neq i$. That is, there exists no other utility variable u_j, such that u_i and u_j share a private decision ancestor.
3. $\gamma_i \neq \emptyset$, $\beta_i \neq \emptyset$, and $\gamma_i \cap \gamma_j \neq \emptyset$ for some $j \neq i$.
4. $\beta_i = \emptyset$. That is, $\alpha_i = \gamma_i$ is entirely private.

For example, suppose $SD = \{d_2, d_3\}$ in Fig. 2. Then u_2 is under case 1, u_1 is under case 2, u_3 is under case 3, and u_4 is under case 4.

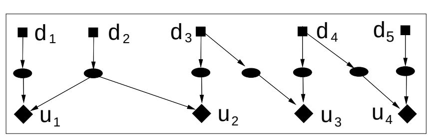

Fig. 2. Illustration of correlated cluster

[Case 1] For this case, $\overline{b_i} = proj(\overline{sd}, \beta_i) = proj(\overline{sd}, \alpha_i)$. We refer to the subnet segment that contains u_i and all its ancestors (including π_i and α_i) as the *subnet segment of* u_i. For instance, subnet segment of u_2 in Fig. 2 includes nodes d_2, d_3, u_2, the two parent nodes of u_2 and links among them.

We observe that contribution of u_i to Eqn. (11) can be evaluated using its subnet segment, independently of other $u_j \in U$, and the contribution is additive. That is, if $meu(\overline{sd}) = v$ is obtained using B's subnet, $meu(\overline{sd}) = v'$ is obtained using

B's subnet with node u_i removed, and $eu(\overline{b_i}) = v''$ is obtained using the subnet segment of u_i, then $v = v' + v''$. We therefore compute $eu(\alpha_i)$ using the subnet segment of u_i according to Eqn. (3). As it involves full evaluation for each $\overline{b_i}$, the complexity is

$$O(\sigma^{|\alpha_i|}\ m\ \kappa^m). \tag{12}$$

[Case 2] For this case, the contribution of u_i to Eqn. (11) is also additive and obtainable from its subnet segment. Therefore, we can apply PEUtilDec1 to the subnet segment. Set the parameters of PEUtilDec1 to $D = \alpha_i$, $\beta = \beta_i$, and $\gamma = \gamma_i$. The return value $meu(\beta_i)$ is the contribution of u_i to Eqn. (11). Note that the return value $peer(\beta_i)$ will be used later and needs to be saved. Applying Proposition 1 to this case, the complexity to obtain $meu(\beta_i)$ is then

$$O(\theta\ \sigma^{|\alpha_i|}\ m\ \kappa^m). \tag{13}$$

[Cases 3 and 4] These utility variables can be grouped into *correlated clusters* (possibly overlapping): Each cluster is obtained by starting with one variable u_i under case 3. Note that whenever cases 1 and 2 do not cover all utility variables, there exist some under case 3. Be definition, there exists u_j such that u_i and u_j share a private decision ancestor. Note that u_j may be under case 3 or case 4. Add u_j to the cluster, and continue until no such utility variable can be found. In Fig. 2, u_3 and u_4 form a correlated cluster. Formally, a correlated cluster can be defined as follows.

Definition 1 *Let $SU = \{u_1, ..., u_{\eta'}\} \subseteq U$ be a subset of utility variables. SU is a correlated cluster if for each u_i ($i = 2, ..., \eta'$), there exists $j < i$ with $\gamma_j \cap \gamma_i \neq \emptyset$, and no proper superset of SU has such property.*

For each correlated cluster with its utility variables indexed as $u_1, ..., u_{\eta'}$, we apply PEUtilDec2 to its subnet segment. Denote $\alpha' = \cup_{i=1}^{\eta'} \alpha_i$, $\beta' = \cup_{i=1}^{\eta'} \beta_i$, $\gamma' = \cup_{i=1}^{\eta'} \gamma_i$. Set parameters of PEUtilDec2 to $D = \alpha'$, $\beta = \beta'$, and $\gamma = \gamma'$. Return value $meu(\beta')$ is contribution of the cluster to Eqn. (11). Return value $peer(\beta')$ is needed later and is to be saved. Applying Proposition 2, complexity to obtain $meu(\beta')$ is

$$O(\theta\ \sigma^{|\alpha'|}\ \eta'\ m\ \kappa^m). \tag{14}$$

Let $SU_1, SU_2, ...$ be subsets of U, where $\cup_i SU_i = U$, and each SU_i is either a singleton under case 1, or a singleton under case 2, or a correlated cluster from cases 3 or 4. Let α_i', β_i' and γ_i' denote the sets of decision ancestor variables for SU_i. Then Eqn. (11) can be computed as

$$utm_0(\overline{sd}) = (\sum_i eu_i(proj(\overline{sd}, \beta_i'))) + \sum_j meu_j(proj(\overline{sd}, \beta_j')), \tag{15}$$

where each $eu_i()$ is the contribution from a SU_i under case 1, and each $meu_j()$ is the contribution either from a SU_j under case 2 or from a SU_j under case 3 or 4. From Eqns. (12), (13) and (14), assuming cases 2, 3, 4 dominate the computation, the complexity to obtain $utm_0(\overline{sd})$ is the following, where $\alpha^* = \max_i \alpha_i'$,

$$O(\theta\ \sigma^{|\alpha^*|}\ \eta\ m\ \kappa^m). \tag{16}$$

It is significantly more efficient than $O(\theta\ \sigma^{\rho}\ \eta\ m\ \kappa^m)$ (Proposition 2) as it would be if PEUtilDec2 is directly applied to agent B's subnet. It is only exponential on cardinality of the largest cluster ancestor set, while the latter is exponential on $|D|$.

7 Utility Message from Non-leaf

Next, we consider non-leaf agents in the first round of message passing. Let D be the set of decision variables of a non-leaf agent B. B receives utility messages from child agents $A_1,...,A_k$, over interfaces $SD_1,...,SD_k$, respectively, and then computes and sends a utility message over interface SD with parent agent C.
We denote message from A_j by $utm_j(SD_j)$, which specifies $utm_j(\overline{sd_j})$ for each local plan $\overline{sd_j}$ over SD_j. To count $utm_j(SD_j)$ in computing message $utm_0(SD)$ to C, we modify subnet as follows: For each decision variable $d_i \in SD_j$ with space Op_i, add a new child node e_i with space $Ef_i = Op_i$. Associate CPT $P(e_i|d_i)$ such that $P(e_i|d_i) = 1$ whenever $e_i = d_i$ and $P(e_i|d_i) = 0$ otherwise. Hence, e_i is deterministically dependent on d_i. Denote the set of new child nodes added relative to SD_j as SE_j. Add a new utility node utm_j with SE_j as its parents, associate it with the function $utm_j(SE_j)$, such that $utm_j(SE_j) = utm_j(SD_j)$, and assign it weight $w_j = 1$.
After conversion for each SD_j, message $utm_0(SD)$ to C is computed by the method in Section 6 and Eqn. (15). The effect is that for each $\overline{sd}$ over SD, $utm_0(\overline{sd})$ is the MEU based on all subnets on the hypertree rooted at the subnet of agent B.
For root agent, after conversion for each SD_j, it performs PEUtilDec2 with $\beta = \emptyset$ to get the optimal local plan $\overline{d^*}$. The first round of message passing ends. Operation by non-leaf agent B is summarized in Algorithm 3. Its main property is established in Proposition 3 with proof omitted for space.

Algorithm 3 *CollectUtilPE*
Input: decision subnet over (D,E,U)*;*

if B is not a leaf agent,
  for each child agent A_i*,*
    receive $utm_j(SD_j)$*; add new utility node* utm_j *and its segment to subnet;*
if B is root agent, call PEUtilDec2 with $\beta = \emptyset$ *and get return value* $\overline{d^*}$*; return;*
$U' = U$*; classify utility variables in* U' *into 4 cases;*
while $U' \neq \emptyset$*,*
  remove $u_i \in U'$ *from* U'*;*
  if u_i *is under case 1, compute eu* (α_i) *using subnet segment of* u_i *and Eqn. (3);*
  else if u_i *is under case 2,*
    call PEUtilDec1 with subnet segment of u_i *and parameters* α_i, β_i, γ_i*;*
    get return value $(meu(\beta_i), peer(\beta_i))$ *and save peer* (β_i)*;*
  else
    for each $u_j \in U'$ *in the same correlated cluster with* u_i*, remove* u_j *from* U'*;*
    call PEUtilDec2 with subnet segment of the cluster and parameters α', β', γ'*;*
    get return value $(meu(\beta'), peer(\beta'))$ *and save peer* (β')*;*
compute $utm_0(SD)$ *from* $eu(\alpha_i)$*,* $meu(\beta_i)$ *and* $meu(\beta')$ *obtained above by Eqn. (15);*
send $utm_0(SD)$ *to agent C;*

Proposition 3 *For each non-root agent B, after completing CollectUtilPE, its message* $utm_0(SD)$ *is the MEU function with respect to joint plans over the union of subenvs on the sub-hypertree rooted at B.*

8 Decision Message Distribution

The second round of message passing starts at root agent. It projects the optimal local plan to interface with each adjacent agent on hypertree, and sends the restricted plan to the agent. When a non-root agent B receives the local plan $\overline{sd^*}$ over its interface SD with the parent agent C, it uses the message to compute its optimal local plan. The computation is organized based on the partition of its utility variables into the four cases in Section 6.
If u_i is under case 1, then $\alpha_i \subset SD$, and the optimal local plan over α_i is

$$\overline{sd_i^*} = proj(\overline{sd^*}, \alpha_i). \tag{17}$$

If u_i is under case 2, B obtains $\overline{b_i^*} = proj(\overline{sd^*}, \beta_i)$, and retrieves $\overline{y_i^*} = peer(\overline{b_i^*})$ using the peer function saved during CollectUtilPE. The optimal local plan over α_i is

$$\overline{sd_i^*} = (\overline{b_i^*}, \overline{y_i^*}). \tag{18}$$

If u_i is under cases 3 or 4, the optimal plan over its correlated cluster with decision ancestor sets α' and β' is obtained in one operation (although we still index the result by i). B obtains $\overline{b'^*} = proj(\overline{sd^*}, \beta')$, and retrieves $\overline{y'^*} = peer(\overline{b'^*})$ using the peer function saved during CollectUtilPE. The optimal local plan over α' is

$$\overline{sd_i^*} = (\overline{b'^*}, \overline{y'^*}). \tag{19}$$

After the optimal local plan over each α_i (case 1 or 2) or α' (case 3 or 4) is defined, the optimal local plan over D (decision nodes in B) is the join

$$\overline{d^*} = (\overline{sd_1^*}, \overline{sd_2^*}, ...). \tag{20}$$

The operation by B is summarized in DistributePEPlan.

Algorithm 4 *DistributePEPlan*
Input: decision subnet over (D, E, U)*;*

if B is root with $\overline{d^*}$ *derived by CollectUtilPE,*
  send $proj(\overline{d^*}, SD_j)$ *to each child agent* A_j*; return;*
receive message $\overline{sd^*}$ *over interface SD with agent C;*
$U' = U$*; classify utility variables in* U' *into 4 cases;*
while $U' \neq \emptyset$*,*
  remove $u_i \in U'$ *from* U'*;*
  if u_i *is under case 1, define* $\overline{sd_i^*}$ *over* α_i *by Eqn. (17);*
  else if u_i *is under case 2, define* $\overline{sd_i^*}$ *over* α_i *by Eqn. (18);*
  else for each $u_j \in U'$ *in the same correlated cluster with* u_i*, remove* u_j *from* U'*;*
    define $\overline{sd_i^*}$ *over* α' *by Eqn. (19);*
compute $\overline{d^*}$ *by Eqn. (20);*
for each child agent A_j*, send message* $proj(\overline{d^*}, SD_j)$ *to* A_j*;*

System coordinator executes DecisionPE, which combines above algorithms. Its optimality is established in Theorem 1, with proof omitted for space.

Algorithm 5 *DecisionPE*

select an agent C arbitrarily;
call CollectUtilPE in C;
call DistributePEPlan in C;

Theorem 1 *After DecisionPE, the joint plan made from joining the local plan $\overline{d^*}$ at each agent is optimal.*

As computation of root agent is dominated by the rest, from Eqn. (16), the *total* complexity of DecisionPE is $O(n\ \theta\ \sigma^{|\alpha^*|}\ \eta\ m\ \kappa^m)$, only exponential on size of the correlated cluster $|\alpha^*|$. This represents an exponential reduction from σ^ρ to $\sigma^{|\alpha^*|}$ as well as a factor θ reduction, relative to the algorithm in [6].

9 Case Study in MAE

Three agent teams are formed to plan for horizon $h = 1$. Two of them (referred to as $3A3D$ and $3A5D$) each has 3 agents, and one ($5A5D$) has 5 agents. $3A3D$ and $3A5D$ are organized into hypertree $A-B-C$, and $5A5D$ into $A-B-C-D-E$. Each decision has $\sigma = 5$. Each subnet has between 1 and 3 public decision variables. Numbers of private decision variables per subnet in $3A3D$, $3A5D$, $5A5D$ are 3, 5, 5, respectively. Hence, the maximum numbers of decision variables per agent for the three teams are 6, 8, 8, respectively. Maximum numbers of local plans per agent are 15625, 390625, 390625, respectively. Numbers of joint plans for the teams are $6.1 \times 10^9, 9.5 \times 10^{13}, 1.4 \times 10^{25}$, respectively.
Three MAE envs are simulated of different reward distributions. Each team is placed at 6 distinct positions in each env, creating 18 distinct decision scenarios. For each scenario, each team is run using the method in [6] (denoted CDNCD) and that in this work (denoted CDNPE). For all teams and all decision scenarios, CDNPE runs obtained identical plans as CDNCD runs, confirming CDNPE optimality.

Table 1. Mean μ and standard deviation std for runtime (in seconds)

Team	CDNPE Time		CDNCD Time	
	μ	std	μ	std
3A3D	3.41	0.05	13.9	0.92
3A5D	4.84	0.24	124	0.58
5A5D	4.78	0.25	127	0.63

Table 1 summarizes runtime per team and method. CDNPE runs significantly faster than CDNCD. For 3A3D (less expensive agents), CDNPE takes 25% of the time used by CDNCD. For 5A5D (more expensive), CDNPE takes 3.8% of the time.

10 Conclusion

We extend multiagent decision algorithm in [6] and generalize partial evaluation for MAE in [9] to propose a new algorithm for length-2 CDNs under pivot probability assumption, reducing computation complexity exponentially compounded with a factor. In relation to MAIDs, ours is tightly coupled while MAIDs are loosely coupled [7]. In relation to DCOP methods such as DPOP, ours is decision theoretic while DPOP is not. Its generality rests on allowing pivot probability beyond a single value. Decision optimality is expected to degrade gracefully as pivot probability assumption is relaxed, and more experimental study is underway to confirm this.

Acknowledgements

Financial support from NSERC, Canada is acknowledged.

References

1. D.S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.
2. F. Oliehoek, Matthijs Spaan, S. Whiteson, and N. Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *Proc. 7th Inter. Conf. on Autonomous Agents and Multiagent Systems*, pages 517–524, 2008.
3. D.V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *J. Artificial Intelligence Research*, 16:389–423, 2002.
4. M. Roth, R. Simmons, and M. Veloso. Exploiting factored representations for decentralized execution in multi-agent teams. In *Proc. 6th Inter. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 469–475, 2007.
5. Y. Xiang, J. Chen, and A. Deshmukh. A decision-theoretic graphical model for collaborative design on supply chains. In A.Y. Tawfik and S.D. Goodwin, editors, *Advances in Artificial Intelligence, LNAI 3060*, pages 355–369. Springer, 2004.
6. Y. Xiang, J. Chen, and W.S. Havens. Optimal design in collaborative design network. In *Proc. 4th Inter. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'05)*, pages 241–248, 2005.
7. Y. Xiang and F. Hanshar. Comparison of tightly and loosely coupled decision paradigms in multiagent expedition. *International Journal of Approximate Reasoning*, 51:600–613, 2010.
8. Y. Xiang and F. Hanshar. Multiagent expedition with graphical models. *Inter. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, 19(6):939–976, 2011.
9. Y. Xiang and F. Hanshar. Partial evaluation for planning in multiagent expedition. In C. Butz and P. Lingras, editors, *Advances in Artificial Intelligence, LNAI 6657*, pages 420–432. Springer, 2011.
10. W. Zhang and Q. Ji. A factorization approach to evaluating simultaneous influence diagrams. *IEEE SMC-A*, 36(4):746–757, 2006.