Partial Evaluation for Planning in Multiagent Expedition

Y. Xiang and F. Hanshar

University of Guelph, Canada

Abstract. We consider how to plan optimally in a testbed, multiagent expedition (MAE), by centralized or distributed computation. As optimal planning in MAE is highly intractable, we investigate speedup through partial evaluation of a subset of plans whereby only the intended effect of a plan is evaluated when certain conditions hold. We apply this technique to centralized planning and demonstrate significant speedup in runtime while maintaining optimality. We investigate the technique in distributed planning and analyze the pitfalls.

1 Introduction

We consider a class of stochastic multiagent planning problems termed multiagent expedition (MAE) [8]. A typical instance consists of a large open area populated by objects as well as mobile agents. Agent activities include moving around the area, avoiding dangerous objects, locating objects of interest, and object manipulation depending on the nature of the application. Successful manipulation of an object may require proper actions of a single agent or may require cooperation of multiple agents coordinating through limited communication. Success of an agent team is evaluated based on the quantity of objects manipulated as well as the quality of each manipulation. MAE is an abstraction of practical problems such as planetary expedition or disaster rescue [3].

Planning in MAE may be achieved by centralized or distributed computation. Its centralized version can be shown to be a partially observable Markov decision process (POMDP) and its distributed version can be shown to be a decentralized POMDP (DEC-POMDP). A number of techniques have been proposed for solving POMDPs [4, 6]. The literature for DEC-POMDPs is growing rapidly, e.g., [1, 5]. Optimal planning is highly intractable in general for either POMDP or DEC-POMDP. Inspired by branch-and-bound techniques to improve planning efficiency [2], we propose a method *partial evaluation* that focuses on the intended effect of a plan and skips evaluation of unintended effects when certain conditions are met.

We focus on on-line planning. We experiment with partial evaluation for centralized planning in MAE and demonstrate a significant speedup in runtime while maintaining plan optimality. We also examine its feasibility in distributed planning. It is found to be limited by local optimality without guaranteed global optimality or intractable agent communication. This result yields insight into distributed planning that suggests future research on approximate planning.

The remainder of the paper is organized as follows: Section 2 reviews background on MAE. Sections 3-6 present partial evaluation for centralized planning with experimental results reported in Section 7. Section 8 first reviews background on collaborative design networks (CDNs), a multiagent graphical model for distributed decision making, and then investigates partial evaluation for distributed planning based on CDNs.

2 Background on Multiagent Expedition

In MAE, an open area is represented as a grid of cells (Figure 1 (a)). At any cell, an agent can move to an adjacent cell by actions *north*, *south*, *east*, *west* or remain there (*halt*). An action has an intended effect (e.g., north in Figure 1 (d)) and a number of unintended effects (other outcomes in (d)), quantified by transition probabilities.

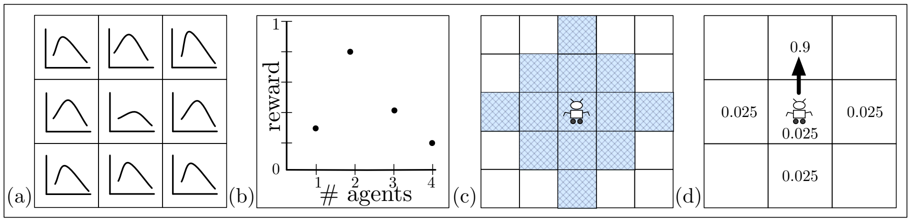

Fig. 1. a) Grid of cells and reward distribution in MAE. b) Cell reward distribution. c) Agent's perceivable area. d) Intended effect (arrow) of action *north*.

The desirability of a cell is indicated by a numerical *reward*. A neutral cell has a reward of a *base value* β. The reward at a harmful cell is lower than β. The reward at an interesting cell is higher than β and can be further increased through agent cooperation.

When a physical object is manipulated (e.g., by 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. We denote the most effective level by λ. Figure 1(b) shows the reward distribution of a single cell with $\lambda = 2$. At this cell, the reward collected by a single agent is 0.3, if two agents cooperate at the cell, each receives 0.8. Reward decreases with more than λ agents, promoting only effective cooperations.

After a cell has been visited by any agent, its reward is decreased to β. As a result, wandering within a neighbourhood is unproductive. Agents have no prior knowledge how rewards are distributed in the area. Instead, at any cell, an agent can reliably perceive its location and reward distribution within a small radius (e.g. shaded cells in Figure 1(c)). An agent can also perceive the location of another agent and communicate if the latter is within a given radius.

Each agent's objective is to move around the area, cooperate as needed, and maximize the team reward over a finite horizon based on local observations and limited communication. For a team of n agents and horizon h, there are 5^{nh}

joint plans each of which has 5^{nh} possible outcomes. With $n = 6$ and $h = 2$, a total of $5^{24} \approx 6 \times 10^{16}$ uncertain outcomes need evaluated. Hence, solving MAE optimally is highly intractable.

In the following, we refer to maximization of *reward* and *utility* interchangeably with the following assumption: A utility is always in $[0, 1]$, no matter if it is the utility of an action, or a plan (a sequence of actions), or an joint action (simultaneous actions by multiple agents), or a joint plan (a sequence of joint actions). In each case, the utility is mapped linearly from $[min\ reward, max\ reward]$, with $min\ reward$ and $max\ reward$ properly defined accordingly.

3 Partial Evaluation

We study how to speedup planning in the context of MAE, based on an idea: partial evaluation. Let a be an action with two possible outcomes: an *intended* and an unintended. The intended outcome has the probability p_1 and utility u_1, and the unintended $p_2 = 1 - p_1$ and u_2, respectively. Its expected utility is evaluated as

$$eu = p_1 u_1 + p_2 u_2. \tag{1}$$

Let a' be an alternative action with the same outcome probabilities p_1 (for intended) and p_2, and utilities u_3 and u_4, respectively. Its expected utility is $eu' = p_1 u_3 + p_2 u_4$. The alternative action a' is *dominated* by a if

$$eu - eu' = eu - p_1 u_3 - p_2 u_4 > 0. \tag{2}$$

From Eqn (2), the following holds:

$$u_3 < \frac{eu}{p_1} - \frac{p_2}{p_1} u_4 \tag{3}$$

Letting u_{max} denote the maximum utility achievable, we have

$$\frac{eu}{p_1} - \frac{p_2}{p_1} u_{max} \leq \frac{eu}{p_1} - \frac{p_2}{p_1} u_4. \tag{4}$$

Eqn (3) is guaranteed to hold if we maintain

$$u_3 < \frac{eu}{p_1} - \frac{1 - p_1}{p_1} u_{max} \equiv t. \tag{5}$$

When the number of alternative actions is large, the above idea can be used to speed up search for best action: For an unevaluated action a', if u_3 satisfies Eqn (5), discard a'. We say that a' is *partially* evaluated. Otherwise, eu' will be *fully* evaluated. If eu' exceeds eu, then a will be updated as a', eu updated as eu', and u_1 will be updated as u_3.

Eqn (5) allows more efficient search without losing optimality, and is an *exact* criterion for partial evaluation. The actual speed-up depends on the threshold t for u_3. The larger the value of t, the less actions that must be fully evaluated, and the more efficient the search.

Consider the value of u_{max}. When utility is bounded by $[0, 1]$, we have the obvious option $u_{max} = 1$. That is, we derive u_{max} from the *global* utility distribution over all outcomes of actions. Threshold t increases as u_{max} decreases. Hence, it is desirable to use a smaller u_{max} while maintaining Eqn (4). One option to achieve this is to use u_{max} from the *local* utility distribution over only outcomes of current alternative actions. The trade-off is the following: With $u_{max} = 1$, it is a constant. With the localized u_{max}, it must be updated before each planning.

4 Single-Agent Expedition

In single-agent expedition, an action a has an intended outcome and four unintended ones. We assume that the intended outcome of all actions have the same probability p_1, and unintended outcomes have the same probability $(1 - p_1)/4$. Hence, we have

$$eu = p_1 u_1 + \sum_{i=1}^{4} u_{2,i}\ (1 - p_1)/4, \tag{6}$$

where $u_{2,i}$ is the utility of the ith unintended outcome. Comparing Eqn (1) and Eqn (6) , we have

$$p_2\ u_2 = \sum_{i=1}^{4} u_{2,i}\ \frac{1 - p_1}{4} = (1 - p_1)\ (\frac{1}{4} \sum_{i=1}^{4} u_{2,i}).$$

If we aggregate the four unintended outcomes as an equivalent single unintended outcome, then this outcome has probability $p_2 = 1 - p_1$ and utility $u_2 = \frac{1}{4} \sum_{i=1}^{4} u_{2,i}$.

Let ua_{max} (where 'a' in 'ua' refers to 'agent') denote the maximum utility of outcomes. Substituting u_2 in Eqn (2) by $\frac{1}{4} \sum_{i=1}^{4} u_{2,i}$, repeating the analysis after Eqn (2), and noting that $\frac{1}{4} \sum_{i=1}^{4} u_{2,i}$ is upper-bounded by ua_{max}, we have an exact criterion for partial evaluation:

$$u_3 < t = \frac{1}{p_1} eu - \frac{1 - p_1}{p_1} ua_{max} \tag{7}$$

As discussed in the last section, the smaller the value of ua_{max}, the more efficient the search. Since ua_{max} was replacing $\frac{1}{4} \sum_{i=1}^{4} u_{4,i}$ (compare Eqns (3) and (5)), we can alternatively replace $\frac{1}{4} \sum_{i=1}^{4} u_{4,i}$ with an upper bound tighter than ua_{max}. Since $\frac{1}{4} \sum_{i=1}^{4} u_{4,i}$ is essentially the average utility over unintended outcomes, we can replace ua_{max} by $\alpha\ ua_{avg}$, where ua_{avg} is the average (local) utility of outcomes and $\alpha \geq 1$ is a scaling factor. This yields the following:

$$u_3 < t = \frac{1}{p_1} eu - \frac{1 - p_1}{p_1} \alpha\ ua_{avg} \tag{8}$$

According to Chebyshev's inequality, the smaller the variance of utilities over outcomes, the closer to 1 the α value can be without losing planning optimality.

5 Single Step MAE by Centralized Planning

Next, we consider multiagent expedition with n agents. Each agent action has k alternative outcomes o_1, ..., o_k, where o_1 is the intended with probability p. A joint action by n agents consists of a tuple of n individual actions and is denoted by $\underline{a}$. The *intended outcome* of $\underline{a}$ is the tuple made of the intended outcomes of individual actions, and is unique. We denote the utility of the intended outcome of $\underline{a}$ by u. Outcomes of individual agent actions are independent of each other given the joint action plan. Hence, the intended outcome of $\underline{a}$ has probability p^n. The expected utility of $\underline{a}$ is

$$eu = p^n \ u + \sum_i p_i \ u_i, \tag{9}$$

where i indexes unintended outcomes, u_i is the utility of an unintended outcome, and p_i is its probability. Note that $p_i \neq p_j$ in general for $i \neq j$, and $p^n + \sum_i p_i = 1$.

Let $\underline{a'}$ be an alternative joint action whose intended outcome has utility u'. Denote the expected utility of $\underline{a'}$ by eu'. The joint action $\underline{a'}$ is *dominated* by joint action $\underline{a}$ if

$$eu - eu' = eu - p^n u' - \sum_i p_i \ u'_i > 0. \tag{10}$$

Eqn (10) can be rewritten as follows:

$$u' < p^{-n} \ (eu - \textstyle\sum_i p_i \ u'_i)$$

Let uts_{avg} (where 't' in 'uts' refers to 'team' and 's' refers to 'single step') denote the average utility of outcomes of joint actions. From

$$0 < p_i < 1 - p^n, 0 < \tfrac{p_i}{1-p^n} < 1, \textstyle\sum_i \tfrac{p_i}{1-p^n} = 1,$$

$$\sum_i p_i \ u'_i = (1 - p^n) \sum_i \frac{p_i}{1 - p^n} \ u'_i,$$

we have the expected value of $\sum_i \frac{p_i}{1-p^n} \ u'_i$ (weighted mean with normalized weights) to be uts_{avg}, and the expected value of $\sum_i p_i \ u'_i$ to be $(1 - p^n) \ uts_{avg}$. We can choose $\alpha \geq 1$ (e.g. based on Chebyshev's inequality) so that it is highly probable $\sum_i p_i \ u'_i \leq (1 - p^n) \ \alpha \ uts_{avg}$ and hence $eu - \sum_i p_i \ u'_i \geq eu - (1 - p^n) \ \alpha \ uts_{avg}$. It then follows from Eqn (10) that the joint action $\underline{a'}$ is dominated by $\underline{a}$ with high probability if the following holds,

$$u' < t = \frac{eu}{p^n} - \frac{1 - p^n}{p^n} \ \alpha \ uts_{avg}, \tag{11}$$

in which case $\underline{a'}$ can be discarded without full evaluation. Note that the condition is independent of k.

In order to compute u' by any agent Ag, it needs to know the intended outcome of the action in $\underline{a'}$ for each other agent, and use this information to determine if any cooperation occurs in the intended outcome of $\underline{a'}$. To do so, it suffices for Ag to know the current location of each agent as well as $\underline{a'}$. Ag also needs to know the unilateral or cooperative reward associated with the intended outcome to calculate u'. When other agents are outside of the observable area of Ag, this information must be communicated to Ag. Similarly, in order to compute uts_{avg}, Ag needs to collect from other agents the average rewards in their local areas.

Alternatively, following a similar analysis, we could base threshold t on uts_{max}, the maximum utility achievable by the outcome of any joint action, and test u' by the following condition:

$$u' < t = \frac{eu}{p^n} - \frac{1-p^n}{p^n}\, uts_{max} \tag{12}$$

Since $uts_{max} > \alpha\ uts_{avg}$, the search is less efficient, but its probability to get the optimal plan is 1. To compute uts_{max}, Ag needs to collect from other agents the maximum rewards in their local areas, instead of average rewards as in the case of uts_{avg}.

6 Multi-Step MAE by Centralized Planning

Consider multiagent expedition with horizon $h \geq 2$ (single step is equivalent to $h = 1$). Each agent selects a sequence $\underline{a}$ of h actions. The n agents collectively select a joint plan $\underline{A}$ (an $n \times h$ array). The intended outcome of joint plan $\underline{A}$ is made of the intended outcomes of all individual actions of all agents. Assume that the outcome of each individual action of each agent is independent of outcomes of its own past actions and is independent of outcomes of actions of other agents (as is the case in MAE). Then the probability of the intended outcome of joint plan $\underline{A}$ is p^{hn}.

We denote the utility of the intended outcome of $\underline{A}$ by u. The expected utility of $\underline{A}$ is then

$$eu = p^{hn}\ u + \sum_i p_i\ u_i, \tag{13}$$

where i indexes unintended outcomes, u_i is the utility of an unintended outcome, and p_i is its probability. Note that $p^{hn} + \sum_i p_i = 1$.

Let $\underline{A'}$ be an alternative joint plan whose intended outcome has utility u'. Denote the expected utility of $\underline{A'}$ by eu'. The joint plan $\underline{A'}$ is *dominated* by $\underline{A}$ if

$$eu - eu' = eu - p^{hn}\ u' - \sum_i p_i\ u'_i > 0. \tag{14}$$

Through an analysis similar to that in the last section, and from the similarity of Eqns (14) and (10), we can conclude the following: Let utm_{avg} (where 'm' in 'utm' refers to 'multi-step') denote the average utility of outcomes of joint plans. Let $\alpha \geq 1$ to be a scaling factor. With a large enough α value, the joint plan $\underline{A'}$ is dominated with high probability by plan $\underline{A}$ if the following inequation holds,

$$u' < t = \frac{eu}{p^{hn}} - \frac{1-p^{hn}}{p^{hn}}\ \alpha\ utm_{avg}, \tag{15}$$

in which case $\underline{A'}$ can be discarded without full evaluation.

In order to compute u' by any agent Ag, it needs to know $\underline{A'}$, the current location of each agent, and unilateral or cooperative reward associated with the intended outcomes. In order to compute utm_{avg}, Ag needs to collect from other agents average rewards in their local areas.

To increase the probability of plan optimality to 1, Ag can use the following test, with the price of less efficient search:

$$u' < t = \frac{eu}{p^{hn}} - \frac{1 - p^{hn}}{p^{hn}}\, utm_{max} \tag{16}$$

7 Centralized Planning Experiment

The experiment aims to provide empirical evidence on efficiency gain and optimality of partial evaluation in multi-step MAE by centralized planning. Two MAE environments are used that differ in transition probability p_t (0.8 or 0.9) for intended outcomes. Agent teams of size $n = 3$, 4 or 5 are run. The base reward $\beta = 0.05$. The most effective level of cooperation is set at $\lambda = 2$. Planning horizon is $h = 2$.

Several threshold values from Section 6 are tested. The first, $utm_{max,1} = 1$, corresponds to the global maximum reward. The second, utm_{max}, corresponds to the local maximum reward for each agent. The third, $utm_{avg,\alpha} = \alpha\ utm_{avg}$, corresponds to average reward over outcomes, scaled up by α. We report result for $\alpha = 1$ as well as for a lower bound that yields an optimal plan by increasing α in 0.25 increments.

Tables 1 and 2 show the result for different values of p_t. Each row corresponds to an experiment run. *Full%* refers to the percentage of plans fully evaluated. *BFR* denotes the team reward of the best joint plan found, and an asterisk indicates if the plan is optimal. *BFR%* denotes ratio of *BFR* over reward of optimal plan. *Time* denotes runtime in seconds.

Table 1. Experiments with $p_t = 0.9$.

n	Threshold	Full%.	BFR	BFR%	Time
3	$utm_{max,1}$	48.87	3.192*	100	3.3
	utm_{max}	0.780	3.192*	100	0.3
	$utm_{avg,1}$	0.172	3.102	97.18	0.1
	$utm_{avg,3}$	0.812	3.192*	100	0.3
4	$utm_{max,1}$	83.51	4.940*	100	142.6
	utm_{max}	0.053	4.940*	100	2.5
	$utm_{avg,1}$	0.046	4.940*	100	1.9
5	$utm_{max,1}$	100	5.262*	100	4671.2
	utm_{max}	0.002	5.046	95.89	52.2
	$utm_{avg,1}$	0.001	5.046	95.89	52.1
	$utm_{avg,5}$	0.19	5.262*	100	62.4

Table 2. Experiments with $p_t = 0.8$.

n	Threshold	Full%.	BFR	BFR%	Time
3	$utm_{max,1}$	100	2.407*	100	6.2
	utm_{max}	2.0	2.407*	100	0.2
	$utm_{avg,1}$	0.16	2.327	96.67	0.1
	$utm_{avg,3}$	2.25	2.407*	100	0.2
4	$utm_{max,1}$	100	3.630*	100	167.3
	utm_{max}	0.068	3.630*	100	19.6
	$utm_{avg,1}$	0.051	3.630*	100	19.0
5	$utm_{max,1}$	100	3.902*	100	6479.5
	utm_{max}	0.002	3.745	95.97	53.5
	$utm_{avg,1}$	0.001	3.745	95.97	52.3
	$utm_{avg,4.5}$	1.704	3.902*	100	136.0

The results show that partial evaluation based on $utm_{max,1}$ is conservative: all plans are fully evaluated in 4 out of 6 runs. Second, utm_{max} finds an optimal plan in 4 out of 6 runs, and $utm_{avg,1}$ in 2 out of 6 runs. Third, partial evaluation based on utm_{max} and $utm_{avg,\alpha}$ shows significant speedup on all runs. For example, with $p_t = 0.8$, $n = 5$ and $utm_{avg,\alpha}$, an optimal plan is found when $\alpha = 4.5$ and only 1.7% of joint plans are fully evaluated. The planning takes 136 seconds or 2% of the runtime (108min) by $utm_{max,1}$ which evaluates all plans fully.

Table 3. Mean (μ) and standard deviation (σ) of team rewards over all plans

n	# Plans	p_t	μ	σ	p_t	μ	σ
3	15,625	0.9	0.558	0.342	0.8	0.542	0.260
4	390,625		0.738	0.462		0.713	0.352
5	9,765,625		0.914	0.514		0.882	0.342

Table 3 shows the mean and standard deviation of team rewards over all joint plans for $n = 3$, 4 and 5, and $p_t = 0.8$ and 0.9. The mean team reward in each case is no more than 23% of the corresponding optimal reward in Tables 1 and 2. For example, consider $n = 5$ and $p_t = 0.8$, the optimal reward from Table 2 is 3.902 whereas the mean reward is 0.882, approximately 23% of the magnitude of the optimal plan. This signifies that the search space is full of low reward plans with very few good plans. Searching such a plan space is generally harder than a space full of high reward plans. The result demonstrates that partial evaluation is able to traverse the search space, skip full evaluation of many low reward plans, and find high reward plans. This is true even for relatively aggressive threshold $utm_{avg,1}$, achieving at least 95% of the optimal reward (see Table 2).

8 Partial Evaluation in Distributed Planning

8.1 Collaborative Design Networks

Distributed planning in MAE can be performed based on multiagent graphical models, known as collaborative design networks (CDNs) [8], whose background is reviewed in this subsection. CDN is motivated by industrial design in supply chains. An agent responsible for a component encodes design knowledge 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.

Dependence structure $G = (V, E)$ is a directed acyclic graph (DAG) whose nodes are mapped to elements of V and whose set E of arcs is from the following legal types: Arc (d, d') $(d, d' \in D)$ signifies a design constraint. Arc (d, m) $(m \in M)$ represents dependency of performance on design. Arc (t, t') $(t, t' \in T)$ represents dependency between environmental factors. Arc (t, m) signifies dependency of performance on environment. Arc (m, m') defines a composite performance measure. Arc (m, u) $(u \in U)$ signifies 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 $util(\pi(u)) \in [0, 1]$. 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 $D \cup T \cup M$. Assuming 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 a supplied component. Agents, one per supplier, form a collaborative design system. Each agent embodies a DN called a *subnet* and agents are organized into a *hypertree*: Each hypernode corresponds to an agent and its subnet. Each hyperlink (called an *agent interface*) corresponds to design parameters shared by the two subnets, which renders them conditionally independent. They are *public* and other subnet variables are *private.* The hypertree specifies to whom an agent communicates 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. Figure 2 shows a trivial CDN for agents A_0, A_1, A_2.

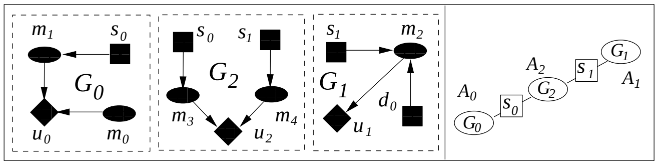

Fig. 2. Subnets G_0, G_1, G_2 (left) and hypertree (right) of a CDN. Design nodes are denoted by s if public and d if private, performance nodes by m, and utility nodes by u.

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 $EU(\mathbf{d}) = \sum_i w_i\ (\sum_j k_{ij}\ (\sum_{\mathbf{m}} u_{ij}(\mathbf{m})\ P(\mathbf{m}|\mathbf{d})))$, 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}. Given a CDN, decision-theoretical optimal design is well defined.

Agents evaluate local designs in batch before communicating over agent interfaces. An arbitrary agent is chosen as communication root. Communication is divided into *collect* and *distribute* stages. *Collect messages* propagate expected utility evaluation of local designs inwards along hypertree towards root. A receiving agent knows the best utility of every local configuration when extended by partial designs in downstream agents. At end of collect stage, the root agent knows the expected utility of the optimal design. *Distribute messages* propagate outwards along hypertree from root. After distribute stage, each agent has identified its local design that is globally optimal (collectively maximize $EU(\mathbf{d})$). Computation (incl. communication) is linear on the number of agents [7] and is efficient for sparse CDNs.

8.2 Distributed Per-Plan Evaluation

We consider partial evaluation in distributed planning based on CDN. Each MAE agent uses a DN to encode its actions (moves) as design nodes, outcomes of actions as performance nodes, and rewards as utility nodes. The hypertree for a team of agents (A, B, C) and DN for agent B are shown in Figure 3. An agent only models and communicates with adjacent agents on hypertree. Movement nodes are labelled mv, performance nodes are labelled ps, and utility nodes are labelled rw.

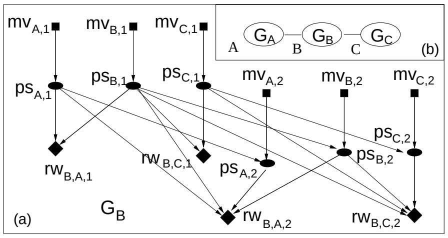

Fig. 3. (a) DN for MAE agent B. (b) Hypertree.

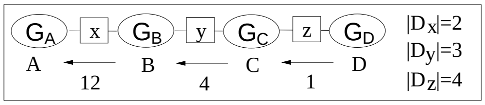

Fig. 4. Message collection where D_x is the domain of x.

As shown earlier, partial evaluation relies on sequentially evaluating (fully or partially) individual joint plans. A distributed per-plan evaluation involves four technical issues: (1) How can a joint plan be evaluated fully? (2) How can it be evaluated partially? (3) As the root agent drives sequential per-plan evaluations, how can it know the total number of joint plans when it does not know other agents' private variables? (4) When a given joint plan is being evaluated, how does each agent know which local plan should be evaluated when it does not know the joint plan as a whole?

First, the existing distributed MAE planning by CDN [8] processes all plans in one batch. At the end of collect stage, the root agent knows the utility of the optimal plan. If we reduce the batch to a single joint plan, at the end of collect stage, root would know the expected utility of that plan.

Second, to evaluate a joint plan partially, instead of passing expected utility, collect messages should contain utility based only on intended outcome.

Third, we propose a method for root to determine the total number of joint plans. Consider the hypertree in Figure 4 over agents A, B, C and D with root A. Assume that x, y and z are the only action variables and are public (no private action variables in MAE). Each agent i maintains a counting variable d_i: the number of joint plans over agents downstream from i. Root A initiates message collection along hypertree (Figure 4). Leaf agent D passes to C message $d_D = 1$ (no downstream agent). C passes $d_C = d_D * |D_z| = 4$ to B, and B passes $d_B = d_C * |D_y| = 12$ to A. In the end, A computes the total number of joint plans as $d_A = d_B * |D_x| = 24$.

Fourth, as any joint plan is evaluated, each agent needs to know how to instantiate their local (public) variables accordingly. For instance, B needs to know the values of x and y, but not z. We assume that the order of domain values of each public variable, e.g., $x \in (x_0, x_1)$, is known to corresponding agents. Joint plans are lexicographically ordered based on domains of public variables. Hence, 0^{th} joint plan corresponds to (x_0, y_0, z_0), and 22^{nd} to (x_1, y_2, z_2).

We propose a message distribution for each agent to determine values of local variables according to current joint plan. Each agent i maintains a working index wr_i. Root A sets wr_A to the index of current joint plan. Each other agent receives wr_i in message. The index of a variable, say x, is denoted by x_{inx}.

Suppose A initiates message distribution with $wr_A = 22$. A computes $x_{inx} = \lfloor \frac{wr_A \ \% \ d_A}{d_B} \rfloor = 1$, where $\%$ and $\lfloor \ \rfloor$ are *mod* and *floor* operations. A passes the index $wr_B = \lfloor wr_A \ \% \ d_A \rfloor = 22$ to B. B computes $x_{inx} = \lfloor \frac{wr_B}{d_B} \rfloor = 1$ and $y_{inx} = \lfloor \frac{wr_B \ \% \ d_B}{d_C} \rfloor = 2$. B passes to C the index $wr_C = \lfloor wr_B \ \% \ d_B \rfloor = 10$. Similar computations at C and D determine $z_{inx} = 2$.

The above can be combined for distributed planning with partial evaluation. It consists of a sequence of message collection followed by one message distribution. The first collection fully evaluates the first joint plan. Local maximum and average utilities from agents are also collected and aggregated for use in all subsequent evaluations (Section 6).

The second collection calls for a partial evaluation (Section 3) of the next joint plan. Upon receiving the response, A determines if the second joint plan needs full evaluation or can be discarded. If full evaluation is needed, A issues the next collection as a full evaluation of the second plan. Otherwise, a call of partial evaluation on the third joint plan is issued. This process continues until all joint plans are evaluated.

One distribution is used after all plans are evaluated to communicate the optimal plan. If 22^{nd} joint plan is optimal, a message distribution as described earlier suffices for each agent to determine its optimal local plan.

It can be shown that the above protocol achieves the same level of optimality as centralized planning. However, for each joint plan, one round of communication is required, resulting in a communication amount exponential on the number of agents and horizon length. This differs from the existing method for planning in CDN (Section 8.1), where two rounds of communication are sufficient.

8.3 Aggregation of Local Evaluation

Given the above analysis, we consider an alternative that attempts to avoid intractable communication: Each agent applies partial evaluation to evaluate all local plans in a single batch. The results are then assembled through message passing in order to obtain the optimal joint plan. After local evaluation, agent i has a set E_i of fully evaluated local plans and a set L_i from partial evaluation. From analysis in Section 3, E_i contains the local optimal plan at i.

Table 4. Utilities for MAE team

Joint Plan	U_A U_B U_C	U_{ABC}	Joint Plan	U_A U_B U_C	U_{ABC}
P_1	0.3 0.3 0.3	0.9	P_3	0.1 0.6 0.1	0.8
P_2	0.6 0.1 0.1	0.8	P_4	0.1 0.1 0.6	0.8

Consider some selected joint plans in Table 4 for agents A, B and C. Each row corresponds to an evaluated joint plan. Each agent i evaluates expected utilities locally as shown in the U_i column. Overall expected utilities are given in the U_{ABC} column as sum of local values. Joint plan P_2 is the best according to evaluation by agent A. P_3 and P_4 are the best according to B and C, respectively. All of them are inferior to P_1.

From the above illustration, the following can be concluded. Optimal planning cannot be obtained from independent local partial evaluations in general. It cannot be obtained based on E_i, nor L_i or their combination.

9 Conclusion

The main contribution of this work is the method of partial evaluation for centralized planning in uncertain environments such as MAE. The key assumption on the environment is that each agent action has a distinguished *intended* outcome whose probability given the action is independent of (or approximately so) the chosen action. This assumption seems to be valid for many problem domains where actions normally achieve some intended consequences where failures are rare occurrences. We devised simple criteria to divide planning computation into *full* and *partial* evaluations to allow only a small subset of alternative plans to be fully evaluated while maintaining optimal or approximate optimal planning. Significant efficiency gains are obtained with our experiments.

Alternatively, extending the method to distributed planning has resulted in unexpected outcomes. Two very different schemes are analyzed. One evaluates individual plans distributively, which demands an intractable amount of agent communication. Another evaluates local plans in batch and assembles the joint plan distributively, but is unable to guarantee a globally optimal joint plan. These analyses discover pitfalls in distributed planning and facilitate development of more effective methods. As such, we are currently exploring other schemes of distributed planning that can benefit from partial evaluation.

Acknowledgements

We acknowledge financial support from Discovery Grant, NSERC, Canada, to the first author, and from NSERC Postgraduate Scholarship, to the second author.

References

1. Besse, C., Chaib-draa, B.: Parallel rollout for online solution of Dec-POMDPs. In: Proc. 21st Inter. Florida AI Research Society Conf. pp. 619–624 (2008)
2. Corona, G., Charpillet, F.: Distribution over beliefs for memory bounded Dec-POMDP planning. In: Proc. 26th. Conf. on Uncertainty in AI (UAI 2010) (2010)
3. Kitano, H.: Robocup rescue: a grand challenge for multi-agent systems. In: Proc. 4th Int. Conf. on MultiAgent Systems. pp. 5–12 (2000)
4. Murphy, K.: A survey of POMDP solution techniques. Tech. rep., U.C. Berkeley (2000)
5. Oliehoek, F., Spaan, M., Whiteson, S., Vlassis, N.: Exploiting locality of interaction in factored Dec-POMDPs. In: Proc. 7th Inter. Conf. on Autonomous Agents and Multiagent Systems. pp. 517–524 (2008)
6. Ross, S., Pineau, J., Chaib-draa, B., Paquet, S.: Online planning algorithms for POMDPs. J. of AI Research pp. 663–704 (2008)
7. Xiang, Y., Chen, J., Havens, W.: Optimal design in collaborative design network. In: Proc. 4th Inter. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'05). pp. 241–248 (2005)
8. Xiang, Y., Hanshar, F.: Planning in multiagent expedition with collaborative design networks. In: Advances in Artificial Intelligence, LNAI 4509. pp. 526–538. Springer-Verlag (2007)