Sequential Decision Making for Intelligent Agents
Papers from the AAAI 2015 Fall Symposium

Exploiting Anonymity in Approximate Linear Programming: Scaling to Large Multiagent MDPs

Philipp Robbel
MIT Media Lab
Cambridge, MA, USA

Frans A. Oliehoek
University of Amsterdam
Amsterdam, The Netherlands

Mykel J. Kochenderfer
Stanford University
Stanford, CA, USA

Abstract

The Markov Decision Process (MDP) framework is a versatile method for addressing single and multiagent sequential decision making problems. Many exact and approximate solution methods attempt to exploit structure in the problem and are based on value factorization. Especially multiagent settings (MAS), however, are known to suffer from an exponential increase in value component sizes as interactions become denser, meaning that approximation architectures are overly restricted in the problem sizes and types they can handle. We present an approach to mitigate this limitation for certain types of MASs, exploiting a property that can be thought of as 'anonymous influence' in the factored MDP. In particular, we show how anonymity can lead to representational and computational efficiencies, both for general variable elimination in a factor graph but also for the approximate linear programming solution to factored MDPs. The latter allows to scale linear programming to factored MDPs that were previously unsolvable. Our results are shown for a disease control domain over a graph with 50 nodes that are each connected with up to 15 neighbors.

1 Introduction

Cooperative multiagent systems (MASs) present an important framework for modeling the interaction between agents that collaborate to jointly solve a task. In the decision-theoretic community, models like the Markov Decision Process (MDP) and its partially observable extensions have both seen widespread use to model and solve such complex planning problems for single and multiple agents in stochastic worlds.

Given the well-known unfavorable complexity results associated with large action and state spaces, many problem representations and their solution methods attempt to exploit structure in the domain for efficiency gains. Factored (equivalently, "graph-based") MDPs (FMDPs) represent the problem in terms of a state space $\mathcal{S}$ that is spanned by a number of state variables, or factors, $X_1, \ldots, X_N$. Their multiagent extension (FMMDP) exploits a similar decomposition over the action space $\mathcal{A}$ and allows the direct representation of the "locality of interaction" that commonly arises in many multiagent settings. This paper uses the running example of a disease control problem over a large network consisting of both controlled an uncontrolled nodes along with the connections that define possible disease propagation paths (Ho et al. 2015; Cheng et al. 2013). The problem of guiding the network to a desired state has a natural formulation as a factored MMDP with factored states (individual nodes in the graph) and action variables (corresponding to the subset of controlled nodes in the graph).

These representational benefits, however, do not in general translate into gains for policy computation (Koller and Parr 1999). Still, many solution methods successfully exploit structure in the domain, both in exact and approximate settings, and have demonstrated scalability to large state spaces (Hoey et al. 1999; Raghavan et al. 2012; Cui et al. 2015). Approaches that address larger numbers of agents are frequently based on value factorization under the assumption that smaller, *localized* value function components can approximate the complete value function well (Guestrin et al. 2003; Kok and Vlassis 2006). The approximate linear programming (ALP) approach of Guestrin et al. (2003) is one of the few approaches in this class that has no exponential dependencies on $\mathcal{S}$ and $\mathcal{A}$ through the efficient computation of the constraints in the linear program based on a variable elimination method. The method retains an exponential dependency on the tree-width (the largest clique formed during variable elimination) meaning that the feasibility of the approximation architecture is based on the connectivity and scale of the underlying graph.

This paper presents an approach to mitigate this limitation for certain types of MASs, exploiting a property that can be thought of as "anonymous influence" in the graph. Anonymity refers to the reasoning over *joint effects* rather than identity of the neighbors in the graph. In the disease control example, the joint infection rates of the parent nodes rather than their individual identity can fully define the behavior of the propagation model. Based on this observation, we show how variable elimination—and the complete set of constraints in the ALP—can still be computed *exactly* for a larger class of graph-based problems than previously feasible.

The contributions of this paper are as follows: first, we define "anonymous influence" for representing aggregate effects in a graph. Second, we develop the concept in a general variable elimination setting and show how it supports compact representations of intermediate functions generated

during elimination. A key contribution is the insight into how a property referred to as "variable consistency" during VE admits particularly compact representations without the need to "shatter" function scopes into disjoint subsets. Third, based on the results for VE, we move to the planning problem and establish how all constraints in the ALP can be represented exactly (albeit more compactly) for factored MDPs that support "anonymous influence". Forth, we contrast the efficiency gains from exploiting anonymous influence on a set of random graphs that can still be solved with the normal VE and ALP methods. We demonstrate speed-ups of the ALP by an order of magnitude to arrive at the identical solution in a sampled set of random graphs with 30 nodes. Last, we address the disease control problem in graph sizes that were previously infeasible to solve with the ALP solution method. We show that the ALP policy outperforms a hand-crafted heuristic by a wide margin in a 50-node graph with 25 controlled agents.

The following section outlines the background on planning in factored domains and introduces the problem of controlling stochastic dynamics over graphs in more detail.

2 Background

Factored MDPs

Markov decision processes are a general framework for sequential decision making under uncertainty (Puterman 2005):

Definition 1. *A Markov decision process (MDP) is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, T, R, h \rangle$, where $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ and $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\}$ are the finite sets of states and actions, T the transition probability function specifying $P(s' \mid s, a)$, $R(s, a)$ the immediate reward function, and h the horizon of the problem.*

Factored MDPs (FMDPs) exploit structure in the state space $\mathcal{S}$ and define the system state by an assignment to the state variables $\mathbf{X} = \{X_1, \ldots, X_n\}$. Transition and reward function decompose into a two-slice dynamic Bayesian network (2TBN) consisting of independent factors, each described by their scope-restricted conditional probability distributions (CPDs). Following a notation similar to Guestrin (2003), under a particular action $a \in A$ the system transitions according to

$$P^a(\mathbf{x}' \mid \mathbf{x}) = \prod_i P^a(x'_i \mid \mathbf{x}[\mathrm{Pa}(X'_i)]) \tag{1}$$

where $\mathrm{Pa}(X'_i)$ denote the parent nodes of X'_i in the graphical model and the term $\mathbf{x}[\mathrm{Pa}(X'_i)]$ the value of the parent variables extracted from the current state $\mathbf{x}$. A similar (additive) decomposition holds for the reward function given state $\mathbf{x}$ and action a, i.e. $R^a(\mathbf{x}) = \sum_{i=1}^{r} R^a_i(\mathbf{x}[\mathbf{C}^a_i])$ for some subset of state factors $\mathbf{C}^a_i \subseteq \{X_1, \ldots, X_n\}$. Note that this yields one 2TBN per action in the single-agent case. An MDP utilizing factored transition and reward models is called a factored MDP (Boutilier, Dean, and Hanks 1999).

In the case of collaborative multiagent systems, the agent set $\mathbf{A} = \{A_1, \ldots, A_g\}$ additionally spans a joint action space $\mathcal{A}$ that is generally exponential in the number of agents. The factored multiagent MDP (FMMDP) is a tractable representation that introduces action variables into the 2TBN (formally a dynamic decision network). The global transition function factors as:

$$P(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) = \prod_i P(x'_i \mid \mathbf{x}[\mathrm{Pa}(X'_i)], \mathbf{a}[\mathrm{Pa}(X'_i)]) \tag{2}$$

where $\mathrm{Pa}(X'_i)$ now include state *and* action variables and each local CPD is only defined over the relevant subsets. Collaborative MASs further assume that each agent observes part of the global reward and is associated with (restricted scope) local reward function R_i, such that the global reward factors additively as $R(\mathbf{x}, \mathbf{a}) = \sum_{i=1}^{g} R_i(\mathbf{x}[\mathbf{C}_i], \mathbf{a}[\mathbf{D}_i])$ for some subsets of state and action factors $\mathbf{C}_i$ and $\mathbf{D}_i$, respectively. In general, factored reward and transitions do not imply a factored value function since scopes grow as the 2TBN is unrolled over time (Koller and Parr 1999).

Further representational efficiencies are possible by exploiting context-specific independence in the model, for example one may use decision trees or algebraic decision diagrams for encoding CPDs and reward functions (Hoey et al. 1999).

The Disease Control Domain

We use the domain of controlling a disease outbreak over a graph to serve as the running example in this paper. The control of stochastic dynamics over graphs has broad application ranging from management of electric power grids to network intrusion (Ho et al. 2015). Other domains aim to minimize collateral diffusion effects while actively targeting specific nodes in the graph (e.g. drugs in biological networks) (Srihari et al. 2014).

Underlying the formulation as a FMMDP is a (directed or undirected) graph $G = (V, E)$ with controlled and uncontrolled vertices $V = (V_c, V_u)$ and edge set $E \subseteq V \times V$. The state space $\mathcal{S}$ is spanned by state variables $X_1, \ldots, X_n$, one per associated vertex V_i, encoding the health of that node. The action set $A = \{A_1, \ldots, A_{|V_c|}\}$ factors similarly over the controlled vertices V_c in the graph and denote an active modulation of the flow out of node $V_i \in V_c$. Let $x_i = \mathbf{x}[X_i]$ and $a_i = \mathbf{a}[A_i]$ denote the state and action for a single node. The reward and transition model factor on a per-node basis

$$R(\mathbf{x}, \mathbf{a}) = \sum_i^n R_i(x_i, a_i) \tag{3}$$

$$P(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) = \prod_i^n T_i(x'_i \mid \mathbf{x}[\mathrm{Pa}(X'_i)], a_i) \tag{4}$$

where the set $\mathrm{Pa}(X'_i)$ includes variable X_i at the previous time step as well as all nodes that *flow into* X_i in G. The known infection transmission probabilities from node j to i, β_{ji}, and the recovery rate of node i, δ_i, define the transition function $T_i(x'_i \mid \mathbf{x}[\mathrm{Pa}(X'_i)], a_i)$ as follows:

$$T_i := \begin{cases} (1 - a_i)(1 - \prod_j (1 - \beta_{ji} x_j)) & \text{if } x_i = 0 \\ (1 - a_i)\,(1 - \delta_i) & \text{otherwise} \end{cases} \tag{5}$$

distinguishing the two cases that X_i was infected at the previous time step (bottom) or not (top). Note

that this model assumes binary state variables $X_i = \{0,1\} = \{\text{healthy}, \text{infected}\}$, and actions $A_i = \{0,1\} = \{\text{do not vaccinate}, \text{vaccinate}\}$ and that $A_u = \{0\}$ for all uncontrolled nodes V_u. The reward function factors as:

$$R(\mathbf{x}, \mathbf{a}) = -\lambda_1 \|\mathbf{a}\|_1 - \lambda_2 \|\mathbf{x}\|_1 \quad (6)$$

where the L_1 norm records a cost λ_2 per infected node X_i and an action cost λ_1 per vaccination action at a controlled node. All our experiments are for the infinite horizon case on an undirected graph G of varying size and structure but with consistent transmission and recovery rates β, δ.

Efficient Solution of Large FMMDPs

Coordinating a set of agents to maximize a shared performance measure (the long-term reward) is challenging because of exponential state and action spaces $\mathcal{S}$, $\mathcal{A}$. One successful approach that extends to *both* large $\mathcal{S}$ and $\mathcal{A}$ represents the joint value function as a linear combination of locally-scoped terms. Each of these addresses a part of the system and covers potentially multiple, even overlapping, state factors: $\mathcal{V}(\mathbf{x}) = \sum_i \mathcal{V}_i(\mathbf{x}[\mathbf{C}_i])$ for local state scopes $\mathbf{C}_i \subseteq \{X_1, \ldots, X_n\}$. Note that in the limit this representation is simply a single joint value function in the global state $\mathbf{x}$; still, one may hope that a set of lower-dimensional components may achieve an adequate approximation in a large structured system.

For *factored linear value functions* given basis function choice $H = \{h_1, \ldots, h_k\}$, each local $\mathcal{V}_i$ can be written as

$$\mathcal{V}_i(\mathbf{x}[\mathbf{C}_i]) = \sum_{j=1}^{k} w_j h_j(\mathbf{x}) \quad (7)$$

where $\mathbf{C}_i$ is a subset of state factors, h_j similarly defined over some distinct subset of variables $\mathbf{C}_{h_j}$ (omitted for clarity), and w_j the weight associated with basis h_j.

Factored Q-value Functions A factored linear state-value function $\mathcal{V}(\mathbf{x}) = \sum_i \mathcal{V}_i(\mathbf{x}[\mathbf{C}_i])$ induces local Q-functions Q_i:

$$\begin{aligned} Q(\mathbf{x}, \mathbf{a}) &= R(\mathbf{x}, \mathbf{a}) + \gamma \sum_{\mathbf{x}'} P(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) \sum_j w_j h_j(\mathbf{x}') \\ &= \sum_{i=1}^{r} R_i(\mathbf{x}, \mathbf{a}) + \gamma \sum_{j=1}^{k} w_i g_i(\mathbf{x}, \mathbf{a}) \end{aligned} \quad (8)$$

where all functions R_i, g_i are again locally-scoped (omitted for clarity) and $g_i(\mathbf{x}, \mathbf{a})$ is the *expectation* of an *individual basis function* h_i, which is computed efficiently via backprojection of h_i through the 2TBN. Local Q-functions follow by associating disjoint subsets of local reward and backprojection functions with each Q_i. Local payoff functions Q_i and agents $A_1, \ldots, A_g$ then span a *factor graph*, i.e. a factorization of the (global) Q-function into locally-scoped terms (see Figure 1).

The globally maximizing *joint action* in a given state $\mathbf{x}$, i.e. $\mathbf{a}^* = \arg\max_{\mathbf{a}} \sum_i Q_i(\mathbf{x}, \mathbf{a})$, is computed efficiently with a distributed max operation in the factor graph via variable elimination (VE) (Koller and Friedman 2009). An important secondary effect is that agents only need to observe the state factors associated with the components in which they participate.

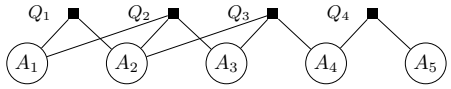

Figure 1: An example factor graph with five agents $(A_1, \ldots, A_5)$ and local payoff functions $Q_1, \ldots, Q_4$. Edges indicate which agent participates in which factor.

VE Variable elimination is a fundamental step for computing the maximizing joint action $\mathbf{a}$ in a factor graph where the enumeration of all joint actions is infeasible. Similar to MAP inference in a Bayesian network, the algorithm eliminates the agents one-by-one and performs only *maximizations* and *summations* over local terms. Let $Q_{\mathbf{x}}$ denote the instantiation of the Q-function in a particular state $\mathbf{x}$ and consider the elimination of agent A_1. Then

$$\max_{\mathbf{a}} Q_{\mathbf{x}}(\mathbf{a}) \equiv \max_{\mathbf{a} \setminus \{a_1\}} \left\{ \Gamma_{\mathbf{x}}(\mathbf{a}) + \max_{a_1} \Gamma^{a_1}_{\mathbf{x}}(\mathbf{a}) \right\} \quad (9)$$

where $\Gamma^{a_1}_{\mathbf{x}}$ collects the sum of all local Q-functions that depend on A_1, and $\Gamma_{\mathbf{x}} = \{Q_{i,\mathbf{x}}\} \setminus \Gamma^{a_1}_{\mathbf{x}}$ those that have no such dependency. The result is a local payoff function $e(\mathbf{a} \setminus \{a_1\}) = \max_{a_1} \Gamma^{a_1}_{\mathbf{x}}$ and A_1 is removed from the factor graph. The execution time is exponential in the size of the largest intermediate term formed which depends on the chosen elimination order.

ALP The approximate linear programming approach computes the best approximation (in a weighted L_1 norm sense) to the optimal value function in the space spanned by the basis functions H (Puterman 2005). The ALP formulation for an infinite horizon discounted MDP given basis functions $h_1, \ldots, h_k$ is given by:

$$\begin{array}{ll} \min\limits_{\mathbf{w}} & \sum_{\mathbf{x}} \alpha(\mathbf{x}) \sum_i w_i h_i(\mathbf{x}) \\ \text{s.t.} & \sum_i w_i h_i(\mathbf{x}) \geq \\ & [R(\mathbf{x}, \mathbf{a}) + \gamma \sum_{\mathbf{x}'} P(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) \sum_i w_i h_i(\mathbf{x}')] \; \forall \mathbf{x}, \mathbf{a} \end{array} \quad (10)$$

for state relevance weights $\alpha(\mathbf{x})$ (assumed uniform here) and variables w_i unbounded. The ALP yields a solution in time polynomial in the sizes of $\mathcal{S}$ and $\mathcal{A}$ but both are exponential for general MASs.

One of the key contributions of Guestrin (2003) is an efficient scheme to represent exponentially many constraints that applies if the basis functions have local scope and transitions and rewards are factored. Reconsider the constraints using the result for backprojections from Equation 8 and let $\hat{\mathcal{V}}(\mathbf{x}) = \sum_i w_i h_i(\mathbf{x})$:

$$\begin{array}{lrcl} \forall \mathbf{x}, \mathbf{a} & \hat{\mathcal{V}}(\mathbf{x}) & \geq & [R(\mathbf{x}, \mathbf{a}) + \gamma \sum_i w_i \, g_i(\mathbf{x}, \mathbf{a})] \\ \forall \mathbf{x}, \mathbf{a} & 0 & \geq & [R(\mathbf{x}, \mathbf{a}) + \sum_i w_i [\gamma g_i(\mathbf{x}, \mathbf{a}) - h_i(\mathbf{x})] \\ \Rightarrow & 0 & \geq & \max_{\mathbf{x},\mathbf{a}} [\sum_r R_r(\mathbf{x}[\mathbf{C}_r], \mathbf{a}[\mathbf{D}_r]) + \\ & & & \sum_i w_i [\gamma g_i(\mathbf{x}, \mathbf{a}) - h_i(\mathbf{x})] \end{array} \quad (11)$$

Note that the exponential set of linear constraints has been replaced by a *single* non-linear constraint and that the replacement is exact. Using a procedure similar to VE (over state and action variables), the max constraint in Equation 11

can be implemented with a small set of linear constraints, avoiding the enumeration of the exponential state and action spaces. Consider an intermediate term $e'(\mathbf{x}[\mathbf{C}])$ obtained after eliminating a state variable X_k from $e(\mathbf{x}[\mathbf{C} \cup \{X_k\})$. Enforcing that e' is maximal over its domain can be implemented with $|Dom(e')|$ new variables and $|Dom(e)|$ new linear constraints in the ALP (Guestrin 2003):

$$e'(\mathbf{x}[\mathbf{C}]) \geq e(\mathbf{x}[\mathbf{C} \cup \{X_k\}]) \quad \forall \mathbf{x}[\mathbf{C} \cup \{X_k\}] \in Dom(e) \tag{12}$$

The total number of resulting linear constraints is only exponential in the size of the largest clique formed during VE.

3 Anonymous Influence

The FMMDPs described in the previous section may not, in general, impose strong constraints on the connectivity of the underlying graph. In fact, many interesting disease propagation settings contain nodes with large in- or out-degrees yielding dense connectivity in some regions of the graph, rendering the ALP solution method intractable (see, for example, the graphs in Figure 3). In this section we develop a novel approach to deal with larger scope sizes than addressed previously.

At the core lies the assumption that in the graph-based problems above, only the *joint effects* of the parents $\text{Pa}(X_i)$—rather than their identity—may determine the outcome at an individual node X_i. We show how under this assumption variable elimination can be run *exactly* in graphs with higher node and degree counts. The key insight is that the exponential representation of intermediate functions e may be reduced to some *subscope* when only the joint effects, rather than the identity, of some variables in the domain $Dom(e)$ need to be considered. In the following, we first address the representation of "joint effects" before turning to how it can be exploited at a computational level during VE (Section 4) and in the ALP (Section 5). In our exposition we assume binary variables but the results carry over to the more general, discrete variable setting.

Count Aggregator Functions

We define count aggregator functions to summarize the "anonymous influence" of a set of variables. In the disease propagation scenario for example, the number of active parents uniquely defines the transition model T_i while the identity of the parent nodes is irrelevant (for *representing* T_i).

Definition 2 (Count Aggregator Function)**.** *Let $\#\{\mathbf{Z}\} : Z_1 \times \ldots \times Z_N \mapsto \mathbb{R}$, $Z_i \in \{0, 1\}$, define a* count aggregator function *(CAF) that takes on $N + 1$ distinct values, one for each setting of k 'enabled' factors Z_i (including the case that no factor is 'enabled'). Note that all permutations of k 'enabled' factors map to the same value.*

Consider, e.g., a monotonically increasing CAF, $\#_i$, that summarizes the number of infected parents of a node X_i in a disease propagation graph. Here, the codomain of $\#_i\{\mathbf{Z}\}$ directly corresponds to $\{0, \ldots, N\}$, i.e. the *number* of 'enabled' factors in $\mathbf{z} \in \mathbf{Z}$.

We delay a discussion of conceptual similarities with generalized (or 'lifted') counters in first-order inference to the comparison with related work in Section 7.

Mixed-mode Functions

We now contrast 'proper' variables with those variables that appear in a counter scope.

Definition 3 (Mixed-Mode Function)**.** *Let $f(\mathbf{X}, \#\{\mathbf{Z}\})$ denote a* mixed-mode function *over domain $\mathbf{X} \times \mathbf{Z} = X_1 \times \ldots \times X_M \times Z_1 \times \ldots \times Z_N$ if,* for every instantiation $\mathbf{x} \in \mathbf{X}$, *$f(\mathbf{x}, \#\{\mathbf{Z}\})$ is a count aggregator function. We refer to $X_i \in \mathbf{X}$ as* proper variables *and $Z_j \in \mathbf{Z}$ as* count variables *in the scope of f.*

A mixed-mode function can be described with (at most) $K^M(N + 1)$ parameters where K is an upper bound on $|Dom(X_i)|$. A CAF is a mixed-mode function where $\mathbf{X} = \emptyset$.

The definition of mixed-mode functions can be extended to allow for multiple count scopes $\#_i$:

Definition 4. *Let $f(\mathbf{X}, \#_i\{\mathbf{Z_i}\}, \ldots, \#_k\{\mathbf{Z_k}\})$ denote a* mixed-mode function *and assume (for now) that $\mathbf{Z}_i \cap \mathbf{Z}_j = \emptyset$ for $i \neq j$. Then, for each of the K^M instantiations $\mathbf{x} \in \mathbf{X}$, there is the induced CAF $f(\mathbf{x}, \#_i\{\mathbf{Z_i}\}, \ldots, \#_k\{\mathbf{Z_k}\})$ which is fully defined by the values assigned to $\#_i\{\mathbf{Z_i}\}, \ldots, \#_k\{\mathbf{Z_k}\}$.*

Consider, e.g., $g(X_1, X_2, \#_1\{\text{Pa}(X_1)\}, \#_2\{\text{Pa}(X_2)\})$ and let $|\text{Pa}(X_1)| = |\text{Pa}(X_2)| = 6$ and $\text{Pa}(X_1) \cap \text{Pa}(X_2) = \emptyset$. While the naive representation has 2^{14} values, the mixed-mode function g can be represented with $2^2 \cdot 7 \cdot 7$ parameters.

We now show how mixed-mode functions can be exploited during variable elimination and during constraint generation in the ALP.

4 Efficient Variable Elimination

Variable elimination (VE) removes variables iteratively from a factor graph given an elimination ordering to implement specific operations (such as marginalization or maximization over a variable). As part of this, VE performs maximizations and summations over local terms. This section shows how the max operation is implemented efficiently when factors are mixed-mode functions (results for summation follow analogously). We begin with the special case that counter scopes in a mixed-mode function are disjoint and then address the general setting.

Consider maximization over proper variable A in the factor graph in Figure 2. Denote the result by the intermediate function $f'(\#\{B_1, B_2, \ldots, B_k\})$ computed as:

$$f'(v) := \max \Big[f(a, v), f(\bar{a}, v) \Big] \tag{13}$$

for all $v \in \{0, \ldots, k\}$ that can be assigned to counter $\#$. This operation is implemented with $k + 1$ operations and has no exponential dependency on k like normal VE when computing f'.

Now consider the elimination of count variable B_i from f. The result is again a mixed-mode function $f''(A, \#\{B_1, \ldots, B_{i-1}, B_{i+1}, \ldots, B_k\})$ where:

$$f''(a, v) := \max \Big[f(a, v), f(a, \mathbf{v} + \mathbf{1}) \Big] \tag{14}$$

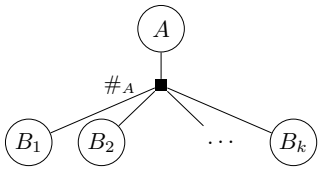

Figure 2: A factor graph with one factor defined by *mixed-mode function* $f(A, \#\{B_1, B_2, \ldots, B_k\})$ where variables B_i only occur in counter scope $\#$ (not the general case).

for all $a \in A, v \in \{0, \ldots, \mathbf{k-1}\}$, since the eliminated count variable *may increase the count by at most 1*. Note that for monotonically increasing (or decreasing) CAFs the max in Equation 14 can be avoided.

Sums of mixed-mode functions can again be written compactly in mixed-mode form. Consider N additional factors $\phi_j(A, \#\{\mathbf{Z}_j\})$ in the factor graph and assume (for now) that all counter scopes are mutually disjoint. Then $l = f + \sum_{j=1}^{N} \phi_j$ can be represented as:

$$l(A, \#\{B_1, \ldots, B_k\}, \#_1\{\mathbf{Z}_1\}, \ldots, \#_N\{\mathbf{Z}_N\}) \qquad (15)$$

which is computed without exponential expansion of any of the variables appearing in a counter scope as $l(a, v, z_1, \ldots, z_N) := f(a, v) + \sum_{j=1}^{N} \phi_j(a, z_i)$ for all valid assignments a, v, z_i.

General Case

Both proper and count variables are in general not uniquely associated with a single factor during VE. For example, in a disease propagation graph, variables may appear as both proper and count variables in different factors. We can distinguish two cases:

Shared proper and count variables Consider factor $e(A, \#\{A, B_1, \ldots, B_k\})$ where A appears as both proper and count variable. Elimination of A requires full instantiation (i.e., it can be considered proper) and it is removed from the counter scope: $e'(a, v) := e(a, \mathbf{a} + \mathbf{v})$ for all $a \in A, v \in \{0, \ldots, k\}$ in a variant of Equation 14 that *enforces consistency* with the choice of (binary) proper variable A, thereby avoiding the max operation in the same Equation. The resulting e' has a representation that is strictly smaller than that of e.

Non-disjoint counter scopes Consider two non-disjoint count variable sets $\#_i, \#_j$ in a function f, i.e. $\mathbf{Z}_i \cap \mathbf{Z}_j \neq \emptyset$. Trivially, there always exists a partition of $\mathbf{Z}_i \cup \mathbf{Z}_j$ of mutually disjoint sets $\{\mathbf{Y}\}, \{\mathbf{W}_i\}, \{\mathbf{W}_j\}$ where $\mathbf{W}_i, \mathbf{W}_j$ denote the variables unique to $\#_i$ and $\#_j$, respectively, and $\mathbf{Y}$ are shared. Associate the counts $\#\{\mathbf{Y}\}$, $\#\{\mathbf{W}_i\}$, and $\#\{\mathbf{W}_j\}$. Then the mixed-mode function f can be written as $f(\mathbf{X}, \#\{\mathbf{Y}\}, \#\{\mathbf{W}_i\}, \#\{\mathbf{W}_j\})$. This observation extends to more than two non-disjoint count variable sets.

In the worst case, a partition of $\bigcup_{i=1}^{k} \mathbf{Z}_i$, $k \geq 2$ requires $p = 3 \cdot 2^{k-2}$ splits into mutually disjoint sets and the resulting representation of f is exponential in p. The next section shows that this 'shattering' into mutually disjoint sets can be avoided and representations be kept compact if *variable consistency is enforced during VE*.

Example 1. *Consider* $f(\#\{A, B, C, D, E\}, \#\{A, B, X, Y, Z\}, \#\{A, C, W, X\})$ *with non-disjoint counter scopes. The direct tabular encoding of* f *requires* $6 \cdot 6 \cdot 5 = 180$ *parameters but contains* invalid entries *due to overlapping counter scopes. The representation of the identical function,* $f'(\#\{A\}, \#\{B\}, \#\{C\}, \#\{D, E\}, \#\{X\}, \#\{W\}, \#\{Y, Z\})$ *with no invalid entries requires* 288 *parameters.*

Note that, in general, the difference in size between shattered and un-shattered representations can be made arbitrarily large.

Compact Representation

As shown above, a representation that avoids invalid entries due to overlapping variable scopes is, in general, less compact than one that does not. We now state a key result that functions do not need to avoid overlapping counter scopes to guarantee valid solutions during variable elimination.

Theorem 1. *Given a mixed-mode function* f *with overlapping counter scopes, variable elimination will never include an invalid value in any of its operations* Op *(e.g., max) on* f*. In particular, denote by* f' *the result of eliminating one variable from* f *and consider a valid entry* v *in* f'*. Then, the operation* Op *used to compute* v *only involves feasible (i.e., consistent) values from* f*. This holds for any elimination ordering* $\mathcal{O}$*.*

Proof. The result is immediate for the case of eliminating proper and *non-shared* count variables. Let f' be the result of such elimination. In case of a proper variable, it follows from Equation 13 that any valid assignment to the variables in f' yields a max operation over valid entries in f. For non-shared count variables, the corresponding counter in f' is reduced by 1 and the max is over assignments $v + 0, v + 1$ to the *reduced counter scope* (Equation 14). Both count assignments are therefore valid in the *extended counter scope* of f. We defer the result for *shared* count variables to the extended version of the paper.

Corollary 1. *The representation of mixed-mode function* $f(\mathbf{X}, \#_1\{\mathbf{Z}_1\}, \ldots, \#_N\{\mathbf{Z}_N\})$ *for proper variable set* $\mathbf{X}$ *and* N *counters is of size* $O(K^{|X|} \cdot L^N)$ *where* K *is an upper bound on* $|Dom(X_i)|$ *and* L *the largest counter value.*

The results above are for general variable elimination but can similarly be exploited in the ALP for efficient constraint generation in both single- and multiagent FMDPs.

5 Exploiting Anonymity in the ALP

For the ALP, the functions $c_i := \gamma g_i - h_i \; \forall h_i \in H$, along with reward functions $R_j, j = 1, \ldots, r$ define a factor graph corresponding to the max constraint in Equation 11. Note that the scopes of c_i are generally larger than those of h_i since parent variables in the 2TBN are added during back-projection. Factors are over state and action variables in the multiagent case.

A key insight is that for a class of factored (M)MDPs defined with count aggregator functions in the 2TBN, the same

intuition as in the previous section applies to implement the non-linear max constraint in the ALP exactly.

We first establish that basis functions $h_i \in H$, when back-projected through the 2TBN (which now includes mixed-mode functions), retain the counters in the resulting back-projections g_i. The backprojection operator is the expectation of basis function h_i defined as $g_i(\mathbf{x}, \mathbf{a}) = \sum_{\mathbf{x}'} P(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) h_i(\mathbf{x})$ and involves summation and product operations only (Guestrin 2003). We have established previously that summation of mixed-mode functions preserves counters (see Equation 15). The same result holds for multiplication when replacing the sum operation in Equation 15 with a multiplication. It follows that g_i (and c_i) preserve counters present in the 2TBN and share the results for compact representations derived in the previous section.

ALP Constraint Generation

The exact implementation of the max constraint via VE in Equation 11 proceeds as before with compact representations. Note that the domain $Dom(e)$ of an intermediate (mixed-mode) term $e(\mathbf{X}, \mathbf{Z})$ with proper and count variable sets $\mathbf{X}$, $\mathbf{Z}$, is reduced as established for general variable elimination in Corollary 1. In particular, the number of variables and constraints in the ALP is exponential only in the *size of the representation* of the largest mixed-mode function formed during VE. Further, the reduction is exact and the ALP computes the identical value-function approximation $\hat{\mathcal{V}} = \mathbf{Hw}$.

6 Experimental Evaluation

We evaluate the method on random disease propagation graphs with 30 and 50 nodes. For the first round of random graph experiments, we obtain the value-function for an uncontrolled disease propagation process and contrast runtimes of the normal VE/ALP method (where possible) with those that exploit "anonymous influence" in the graph. We then move to a controlled disease propagation process with 25 agents in a 50-node graph and compare the results of the obtained policy to two heuristics.

All examples implement the disease control domain from Section 2. For the regular VE/ALP, the parent scope in T_i includes only 'proper' variables as usual. The alternative implementation utilizes count aggregator functions $\#\{\text{Pa}(X_i')\}$ in every T_i. We use identical transmission and node recovery rates throughout the graph, $\beta = 0.6$, $\delta = 0.3$. Action costs are set to $\lambda_1 = 1$ and infection costs to $\lambda_2 = 50$. All experiments implement the same greedy elimination heuristic for VE that minimizes the scope size at the next iteration.

Random Graphs

We use graph-tool (Peixoto 2014) to generate 10 random graphs with an out-degree k sampled from $P(k) \propto 1/k$, $k \in [1, 10]$. Table 1 summarizes the graphs and Figure 3 illustrates a subset. 60 indicator basis functions I_{X_i}, $I_{\bar{X}_i}$ (covering instantiations x_i, $\bar{x}_i$ for all 30 variables X_i), along with 30 reward functions R_i, are utilized in the ALP. The *factor graph* consists of functions c_i that additionally span the parent scope of h_i. Table 1 shows minimum and maximum node degrees, which correspond to lower bounds on parent scope sizes since action and state factors from the previous time step are added in Equation 4.

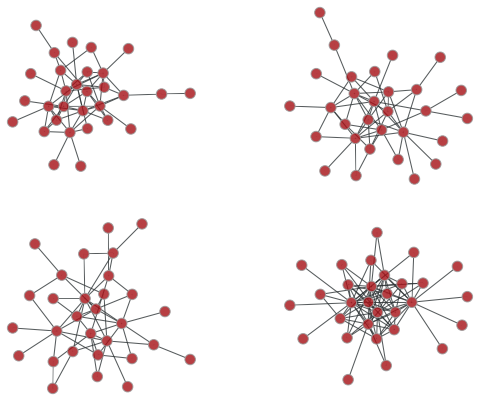

Figure 3: From top left to bottom right: sample of three random graphs in the test set with 30 nodes and a maximum out-degree of 10 (first three). Bottom right: test graph with an increased out-degree sampled from $[1, 20]$.

Mean/min/max degree:				
4.2/1/10	3.4/1/10	3.7/1/10	3.7/1/10	3.7/1/10
2.8/1/10	3.5/1/10	3.1/1/9	3.2/1/8	3.9/1/9

Table 1: Properties of the 10 random 30-node graphs.

The results are summarized in Table 2. Recorded are the number of resulting constraints, the wall-clock times for VE to generate the constraints, and the ALP runtime to solve the value-function after constraint generation on the identical machine. The last three columns record the gains in efficiency per graph.

Lastly, we test with a graph with a larger out-degree (k sampled from the interval $[1, 20]$, shown at the bottom right of Figure 3). The disease propagation problem over this graph cannot be solved with the normal VE/ALP because of exponential blow-up of intermediate terms. The version exploiting anonymous influence can perform constraint generation using VE in 124.7s generating 5816731 constraints.

Disease Control

In this section we show results of policy simulation for three distinct policies in the disease control task over two random graphs (30 nodes with 15 agents, and 50 nodes with 25 agents with a maximum out-degree per node of 15 neighbors: $|\mathcal{S}| = 2^{50}$, $|\mathcal{A}| = 2^{25}$). Besides a random policy, we consider a heuristic (referred to as "copystate" policy) that applies a vaccination action at X_i if X_i is infected in the current state. It is reactive and does not provide anticipatory vaccinations if some of its parent nodes are infected. The "copystate" heuristic serves as our main comparison metric for these large scale graphs where optimal solutions are not available. We are not aware of other MDP solution methods that scale to these state sizes *and* agent numbers in a densely

|C1|, VE1, ALP1	|C2|, VE2, ALP2	|C2|/|C1|	VE2/VE1	ALP2/ALP1
131475, 6.2s, **1085.8s**	94023, 1.5s, **25.37s**	0.72	0.24	**0.02**
24595, 1.1s, 3.59s	12515, 0.17s, 1.2s	0.51	0.15	0.33
55145, 3.5s, 30.43s	27309, 0.4s, 8.63s	0.5	0.11	0.28
74735, 3.0s, 115.83s	41711, 0.69s, 12.49s	0.56	0.23	0.11
71067, 4.16s, 57.1s	23619, 0.36s, 8.86s	0.33	0.08	0.16
24615, **1.6s**, 1.15s	**4539**, **0.07s**, 0.35s	**0.18**	**0.04**	0.30
63307, 2.2s, 141.44s	34523, 0.39s, 4.03s	0.55	0.18	0.03
57113, 0.91s, **123.16s**	40497, 0.49s, **2.68s**	0.71	0.54	**0.02**
28755, 0.54s, 17.16	24819, 0.36s, 3.86s	0.86	0.67	0.22
100465, 2.47s, 284.75s	38229, 0.62s, 36.76s	0.38	0.25	0.13
	Average reduction:	0.53	0.25	0.16

Table 2: Results of random graph experiment. Shown are constraint set sizes, VE and ALP solution times for both normal implementation (column 1) and the one exploiting anonymous influence (column 2). Highlighted in bold are the maximal reductions for each of the three criteria.

connected graph (see related work in Section 7).

The ALP is solved with the exact max constraint by exploiting anonymous influence in the graph. It is not possible to solve this problem with the normal ALP due to infeasibly large intermediate terms being formed during VE. All nodes are covered with two indicator basis functions I_{X_i} and $I_{\bar{X}_i}$ as in the previous experiment. Results for the 30 and 50-node control tasks are shown in Figure 4 with 95% confidence intervals. The "copystate" heuristic appears to work reasonably well in the first problem domain but is consistently outperformed by the ALP solution which can administer anticipatory vaccinations. This effect actually becomes more pronounced with *fewer* agents: we experimented with 6 agents in the identical graph and the results (not shown) indicate that the "copystate" heuristic performs significantly worse than the random policy with returns averaging up to -20000 in a subset of the trials. This is presumably because blocking out disease paths early becomes more important with fewer agents since the lack of agents in other regions of the graph cannot make up for omissions later. Similarly, in the 50-node scenario the reactive "copystate" policy does not provide a statistically significant improvement over a random policy (Figure 4).

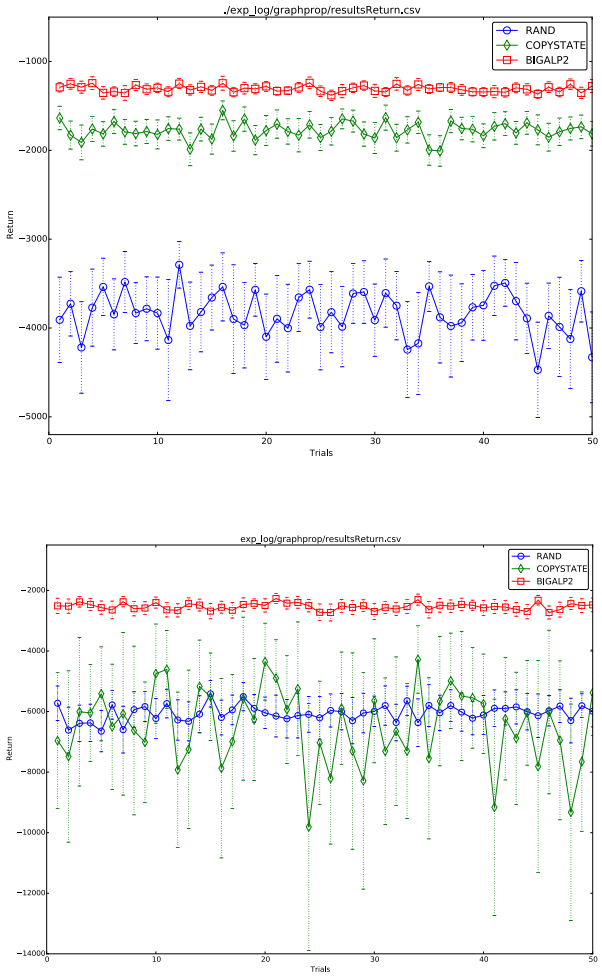

Figure 4: Mean return for 50 trials of 200 steps each in the 30-node disease control domain with 15 agents (top) and 50-nodes with 25 agents (bottom). All results are averaged over 50 runs and shown with 95% confidence intervals for each of *random*, *"copystate" heuristic*, and *ALP policy* (see text).

7 Related Work

Many recent algorithms tackle domains with large (structured) state spaces. For exact planning in factored domains, SPUDD exploits an efficient, decision diagram-based representation (Hoey et al. 1999). Monte Carlo tree search (MCTS) has been a popular online approximate planning method to scale to large (not necessarily factored) domains (Silver, Sutton, and Müller 2008). These methods do not apply to exponential action spaces without further approximations. Ho et al. (2015), for example, evaluated MCTS with 3 agents for a targeted version of the graph control problem. Recent variants that exploit factorization (Amato and Oliehoek 2015) may be applicable.

Our work is based on earlier contributions of Guestrin (2003) on exploiting factored value functions to scale to large factored action spaces. Similar assumptions can be exploited by inference-based approaches to planning which have been introduced for MASs where policies are represented as finite state controllers (Kumar, Zilberstein, and Toussaint 2011). There are no assumptions about the policy in our approach. The variational framework of Cheng et al. (2013) uses belief propagation (BP) and is exponential in the cluster size of the graph. Results are shown for 20-node graphs with out-degree 3 and a restricted class of chain graphs. The results here remain exponential in tree-width but exploit anonymous influence in the graph to scale to random graphs with denser connectivity. A more detailed comparison with (approximate) loopy BP is future work.

First-order (FO) methods (Sanner and Boutilier 2009; Milch et al. 2008) solve planning problems in lifted domains without resorting to grounded representations. Our ideas share a similarity with "generalized counts" in FO models that can eliminate indistiguishable variables in the same predicate in a single operation. Our contributions are distinct from FO methods. Anonymous influence applies in propositional models and to node sets that are not necessarily indistiguishable in the problem. Even if nodes appear in

a count aggregator scope of some X_i in the network, they are further *uniquely connected in the graph* and are unique instances. We also show that shattering into disjoint counter scopes is not required during VE and show how this results in efficiency gains during VE.

Lastly, decentralized and partially-observable frameworks exist to model a larger class of MASs (Oliehoek, Spaan, and Vlassis 2008). The issue of scalability in these models due to negative complexity results is an active field of research.

8 Conclusions and Future Work

This paper introduces the concept of "anonymous influence" in large factored multiagent MDPs and shows how it can be exploited to scale variable elimination and approximate linear programming beyond what has been previously solvable. The key idea is that both representational and computational benefits follow from reasoning about influence of variable sets rather than variable identity in the factor graph. These results hold for both single and multiagent factored MDPs and are exact reductions, yielding the identical result to the normal VE/ALP, while greatly extending the class of graphs that can be solved. Potential future directions include approximate methods (such as loopy BP) in the factor graph to scale the ALP to even larger problems and to support increased basis function coverage in more complex graphs.

Acknowledgments

F.O. is supported by NWO Innovational Research Incentives Scheme Veni #639.021.336.

References

Amato, C., and Oliehoek, F. A. 2015. Scalable planning and learning for multiagent POMDPs. In *Twenty-Ninth Conference on Artificial Intelligence (AAAI)*, 1995–2002.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11:1–94.

Cheng, Q.; Liu, Q.; Chen, F.; and Ihler, A. 2013. Variational planning for graph-based MDPs. In Burges, C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 26*, 2976–2984.

Cui, H.; Khardon, R.; Fern, A.; and Tadepalli, P. 2015. Factored MCTS for large scale stochastic planning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 3261–3267.

Guestrin, C.; Koller, D.; Parr, R.; and Venkataraman, S. 2003. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research* 19:399–468.

Guestrin, C. 2003. *Planning Under Uncertainty in Complex Structured Environments*. Ph.D. Dissertation, Computer Science Department, Stanford University.

Ho, C.; Kochenderfer, M. J.; Mehta, V.; and Caceres, R. S. 2015. Control of epidemics on graphs. In *54th IEEE Conference on Decision and Control (CDC)*.

Hoey, J.; St-Aubin, R.; Hu, A. J.; and Boutilier, C. 1999. Spudd: Stochastic planning using decision diagrams. In *Proceedings of Uncertainty in Artificial Intelligence*.

Kok, J. R., and Vlassis, N. A. 2006. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research* 7:1789–1828.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Koller, D., and Parr, R. 1999. Computing factored value functions for policies in structured MDPs. In *Proc. Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 1332–1339.

Kumar, A.; Zilberstein, S.; and Toussaint, M. 2011. Scalable multiagent planning using probabilistic inference. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2140–2146.

Milch, B.; Zettlemoyer, L. S.; Kersting, K.; Haimes, M.; and Kaelbling, L. P. 2008. Lifted probabilistic inference with counting formulas. In *Twenty Third Conference on Artificial Intelligence (AAAI)*, 1062–1068.

Oliehoek, F. A.; Spaan, M. T. J.; and Vlassis, N. 2008. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research* 32:289–353.

Peixoto, T. P. 2014. The graph-tool python library. *figshare*.

Puterman, M. L. 2005. *Markov decision processes: discrete stochastic dynamic programming*. New York: John Wiley & Sons. A Wiley-Interscience publication.

Raghavan, A.; Joshi, S.; Fern, A.; Tadepalli, P.; and Khardon, R. 2012. Planning in factored action spaces with symbolic dynamic programming. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

Sanner, S., and Boutilier, C. 2009. Practical solution techniques for first-order MDPs. *Artificial Intelligence* 173(5-6):748–788.

Silver, D.; Sutton, R. S.; and Müller, M. 2008. Sample-based learning and search with permanent and transient memories. In *Twenty-Fifth International Conference on Machine Learning (ICML)*, 968–975.

Srihari, S.; Raman, V.; Leong, H. W.; and Ragan, M. A. 2014. Evolution and controllability of cancer networks: A Boolean perspective. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* 11(1):83–94.