Exploiting Symmetries in POMDPs for Point-Based Algorithms

Kee-Eung Kim
Department of Computer Science
Korea Advanced Institute of Science and Technology
383-1 Guseong-dong Yuseong-gu
Daejeon 305-701, Korea
kekim@cs.kaist.ac.kr

Abstract

We extend the model minimization technique for partially observable Markov decision processes (POMDPs) to handle symmetries in the joint space of states, actions, and observations. The POMDP symmetry we define in this paper cannot be handled by the model minimization techniques previously published in the literature. We formulate the problem of finding the symmetries as a graph automorphism (GA) problem, and although not yet known to be tractable, we experimentally show that the sparseness of the graph representing the POMDP allows us to quickly find symmetries. We show how the symmetries in POMDPs can be exploited for speeding up point-based algorithms. We experimentally demonstrate the effectiveness of our approach.

Partially observable Markov decision processes (POMDPs) are a natural model for stochastic sequential decision problems under observation uncertainty. However, due to intractability results in solving POMDPs, common algorithms are hindered by poor scalability on the size of the problem. One approach to address this issue is to exploit the redundancy present in the model: by aggregating equivalent states of the model, we can derive a minimized model which can be solved by traditional algorithms with a reduced computational complexity.

Such model minimization method has been explored in depth for Markov decision processes (MDPs) and POMDPs. Dean & Givan (1997) define the notion of state equivalence in MDPs and provide a model minimization algorithm that computes the partition of the state space so that the blocks of the partition can be regarded as abstract states. The result is a reduced MDP, while preserving the equivalence to the original MDP, which can be solved by traditional algorithms. Since the complexity of the algorithm depends on the size of the state space, we achieve savings in the computational cost compared to directly solving the original MDP. Pineau, Gordon, & Thrun (2003) extends the technique to POMDPs.

These model minimization methods concern with grouping behaviorally equivalent states only while leaving the original actions and observations unchanged. Hence, further reduction is possible by *recoding* actions and observations.

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Ravindran & Barto (2001) define this type of regularity as the *symmetry* in the model, and extend the model minimization method to cover the symmetries in MDPs. Later work by Wolfe (2006) extends the method to POMDPs.

In this paper, we study the symmetry in POMDPs that has not been previously explored in the literature. While previous approaches have been primarily focused on aggregating states in order to reduce the size of the model, we are interested in the POMDP symmetry that is *not* related to reducing the size of the POMDP, nonetheless can be exploited to speed up conventional POMDP algorithms.

Homomorphisms of POMDPs

A POMDP M is defined as a 7-tuple $\langle S, A, Z, T, O, R, b_0 \rangle$: S is a set of states; A is a set of actions; Z is a set of observations; T is a transition function where $T(s,a,s')$ denotes the probability $P(s'|s,a)$ of changing to state s' from state s by executing action a; O is an observation function where $O(s,a,z)$ denotes the probability $P(z|s,a)$ of making observation z when executing action a and arriving in state s; R is a reward function where $R(s,a)$ denotes the immediate reward of executing action a in state s; b_0 is the initial belief where $b_0(s)$ is the probability at we start at state s.

Model minimization methods for POMDPs search for a homomorphism ϕ that maps M to another equivalent POMDP $M' = \langle S', A', Z', T', O', R' \rangle$ with the minimal model size. Formally, homomorphism ϕ is defined as $\langle f, g, h \rangle$ where $f : S \to S'$ is the function that maps the states, $g : A \to A'$ maps the actions, and $h : Z \to Z'$ maps the observations. Note that M' is a *reduced model* of M if any of the mappings is *many-to-one*.

Depending on the definition of homomorphism ϕ, we obtain different definitions of the minimal model. Pineau, Gordon, & Thrun (2003) concern themselves with homomorphism ϕ defined only on the state spaces; the action and observation spaces remain the same as the original POMDP. Hence ϕ takes a form of $\langle f, 1, 1 \rangle$ where 1 denotes the identity mapping. f should satisfy the following constraints in order to hold equivalence between M and M':

$$\begin{aligned} T'(f(s), a, f(s')) &= \textstyle\sum_{s'' \in f^{-1}(s')} T(s, a, s'') \\ R'(f(s), a) &= R(s, a) \\ O'(f(s), a, z) &= O(s, a, z) \end{aligned}$$

$T(s, a_{\text{LISTEN}}, s')$	$s' = s_{\text{LEFT}}$	$s' = s_{\text{RIGHT}}$
$s = s_{\text{LEFT}}$	1.0	0.0
$s = s_{\text{RIGHT}}$	0.0	1.0
$T(s, a_{\text{LEFT}}, s')$	$s' = s_{\text{LEFT}}$	$s' = s_{\text{RIGHT}}$
$s = s_{\text{LEFT}}$	0.5	0.5
$s = s_{\text{RIGHT}}$	0.5	0.5
$T(s, a_{\text{RIGHT}}, s')$	$s' = s_{\text{LEFT}}$	$s' = s_{\text{RIGHT}}$
$s = s_{\text{LEFT}}$	0.5	0.5
$s = s_{\text{RIGHT}}$	0.5	0.5

Figure 1: Transition probabilities of the tiger problem

$O(s, a_{\text{LISTEN}}, z)$	$z = z_{\text{LEFT}}$	$z = z_{\text{RIGHT}}$
$s = s_{\text{LEFT}}$	0.85	0.15
$s = s_{\text{RIGHT}}$	0.15	0.85
$O(s, a_{\text{LEFT}}, z)$	$z = z_{\text{LEFT}}$	$z = z_{\text{RIGHT}}$
$s = s_{\text{LEFT}}$	0.5	0.5
$s = s_{\text{RIGHT}}$	0.5	0.5
$O(s, a_{\text{RIGHT}}, z)$	$z = z_{\text{LEFT}}$	$z = z_{\text{RIGHT}}$
$s = s_{\text{LEFT}}$	0.5	0.5
$s = s_{\text{RIGHT}}$	0.5	0.5

Figure 2: Observation probabilities of the tiger problem

$R(s, a)$	$a = a_{\text{LISTEN}}$	$a = a_{\text{LEFT}}$	$a = a_{\text{RIGHT}}$
$s = s_{\text{LEFT}}$	-1	-100	10
$s = s_{\text{RIGHT}}$	-1	10	-100

Figure 3: Reward function of the tiger problem

Wolfe (2006) extends the minimization method to compute homomorphism of a more general form $\langle f, g, h\rangle$ where the observation mapping h can change depending on the action. The constraints for the equivalence are given by:

$$T'(f(s), g(a), f(s')) = \textstyle\sum_{s'' \in f^{-1}(s')} T(s, a, s'')$$
$$R'(f(s), g(a)) = R(s, a)$$
$$O'(f(s), g(a), h_a(z)) = O(s, a, z)$$

Note that these methods are interested in finding many-to-one mappings in order to find a model with reduced size. Hence, they focus on computing *partitions* of the state, action, and observation spaces of which blocks represent aggregates of equivalent states, actions, and observations, respectively. Once the partitions are found, we can employ conventional POMDP algorithms on the abstract POMDP with reduced number of states, actions, or observations, which in effect reduce the computational complexities of algorithms.

Automorphisms of POMDPs

An automorphism is a special class of homomorphism. In the context of POMDPs, an automorphism ϕ is defined as $\langle f, g, h\rangle$ where the state mapping $f : S \rightarrow S$, the action mapping $g : A \rightarrow A$, and the observation mapping $h : Z \rightarrow Z$ are all one-to-one mappings. Hence, ϕ maps the original MDP to itself, and there is no assumption regarding the reduction in the size of the model.

The classic tiger domain (Kaelbling, Littman, & Cassandra 1998) is perhaps one of the best examples to describe automorphisms in POMDPs. The state space S of the tiger domain is defined as $\{s_{\text{LEFT}}, s_{\text{RIGHT}}\}$, representing the state of the world when the tiger is behind the left door or the right door, respectively. The action space A is defined as $\{a_{\text{LEFT}}, a_{\text{RIGHT}}, a_{\text{LISTEN}}\}$, representing opening the left door, opening the right door, or listening, respectively. The observation space Z is defined as $\{z_{\text{LEFT}}, z_{\text{RIGHT}}\}$ representing hearing on the left or hearing on the right, respectively. The specifications of transition probabilities, observation probabilities, and the rewards are as given in Figure 1, Figure 2, and Figure 3. The initial belief is given as $b_0(s_{\text{LEFT}}) = b_0(s_{\text{RIGHT}}) = 0.5$.

Note that the tiger problem is already compact in the sense that minimization methods mentioned in the previous section cannot reduce the size of the model: examining the reward function alone, we cannot aggregate a_{LEFT} and a_{RIGHT} since the rewards are different depending on the current state being either s_{LEFT} or s_{RIGHT}. By a similar argument, we cannot reduce the state space nor the observation space. Thus if we were to compute partitions using minimization methods, we will end up with every block being a singleton set.

However, s_{LEFT} and s_{RIGHT} can be interchanged to yield an equivalent POMDP, simultaneously changing the corresponding actions and observations:

$$f(s) = \begin{cases} s_{\text{RIGHT}} & \text{if } s = s_{\text{LEFT}} \\ s_{\text{LEFT}} & \text{if } s = s_{\text{RIGHT}} \end{cases}$$

$$g(a) = \begin{cases} a_{\text{LISTEN}} & \text{if } a = a_{\text{LISTEN}} \\ a_{\text{RIGHT}} & \text{if } a = a_{\text{LEFT}} \\ a_{\text{LEFT}} & \text{if } a = a_{\text{RIGHT}} \end{cases}$$

$$h(z) = \begin{cases} z_{\text{RIGHT}} & \text{if } z = z_{\text{LEFT}} \\ z_{\text{LEFT}} & \text{if } z = z_{\text{RIGHT}} \end{cases}$$

The automorphism of POMDPs is the type of regularity we intend to exploit in this paper: the symmetry in the model that does not necessarily help the model minimization algorithm further reduce the size of the model. Hence, rather than computing partitions, we focus on computing all possible automorphisms of the original POMDP.

Note that if the original POMDP can be reduced in size, we can have exponentially many automorphisms in the number of blocks in the partition. For example, if the model minimization yields a state partition with K blocks of 2 states each, the number of automorphisms becomes 2^K. Hence, it is advisable to compute automorphism after we compute the minimal model of POMDP.

Graph Automorphism for POMDPs

In this section, we describe how we can cast the problem of finding automorphisms in POMDPs as a graph automor-

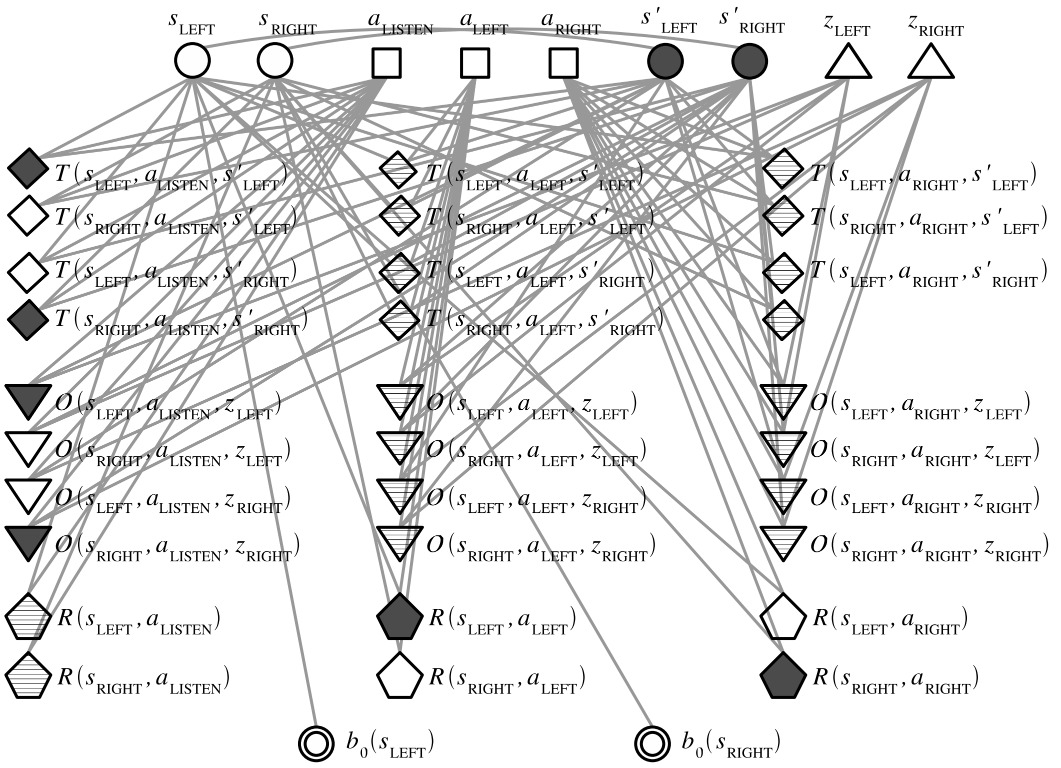
s_{LEFT}
s_{RIGHT}
a_{LISTEN}
a_{LEFT}
a_{RIGHT}
s'_{LEFT}
s'_{RIGHT}
z_{LEFT}
z_{RIGHT}
$T(s_{\text{LEFT}}, a_{\text{LISTEN}}, s'_{\text{LEFT}})$
$T(s_{\text{RIGHT}}, a_{\text{LISTEN}}, s'_{\text{LEFT}})$
$T(s_{\text{LEFT}}, a_{\text{LISTEN}}, s'_{\text{RIGHT}})$
$T(s_{\text{RIGHT}}, a_{\text{LISTEN}}, s'_{\text{RIGHT}})$
$T(s_{\text{LEFT}}, a_{\text{LEFT}}, s'_{\text{LEFT}})$
$T(s_{\text{RIGHT}}, a_{\text{LEFT}}, s'_{\text{LEFT}})$
$T(s_{\text{LEFT}}, a_{\text{LEFT}}, s'_{\text{RIGHT}})$
$T(s_{\text{RIGHT}}, a_{\text{LEFT}}, s'_{\text{RIGHT}})$
$T(s_{\text{LEFT}}, a_{\text{RIGHT}}, s'_{\text{LEFT}})$
$T(s_{\text{RIGHT}}, a_{\text{RIGHT}}, s'_{\text{LEFT}})$
$T(s_{\text{LEFT}}, a_{\text{RIGHT}}, s'_{\text{RIGHT}})$
$O(s_{\text{LEFT}}, a_{\text{LISTEN}}, z_{\text{LEFT}})$
$O(s_{\text{RIGHT}}, a_{\text{LISTEN}}, z_{\text{LEFT}})$
$O(s_{\text{LEFT}}, a_{\text{LISTEN}}, z_{\text{RIGHT}})$
$O(s_{\text{RIGHT}}, a_{\text{LISTEN}}, z_{\text{RIGHT}})$
$O(s_{\text{LEFT}}, a_{\text{LEFT}}, z_{\text{LEFT}})$
$O(s_{\text{RIGHT}}, a_{\text{LEFT}}, z_{\text{LEFT}})$
$O(s_{\text{LEFT}}, a_{\text{LEFT}}, z_{\text{RIGHT}})$
$O(s_{\text{RIGHT}}, a_{\text{LEFT}}, z_{\text{RIGHT}})$
$O(s_{\text{LEFT}}, a_{\text{RIGHT}}, z_{\text{LEFT}})$
$O(s_{\text{RIGHT}}, a_{\text{RIGHT}}, z_{\text{LEFT}})$
$O(s_{\text{LEFT}}, a_{\text{RIGHT}}, z_{\text{RIGHT}})$
$O(s_{\text{RIGHT}}, a_{\text{RIGHT}}, z_{\text{RIGHT}})$
$R(s_{\text{LEFT}}, a_{\text{LISTEN}})$
$R(s_{\text{RIGHT}}, a_{\text{LISTEN}})$
$R(s_{\text{LEFT}}, a_{\text{LEFT}})$
$R(s_{\text{RIGHT}}, a_{\text{LEFT}})$
$R(s_{\text{LEFT}}, a_{\text{RIGHT}})$
$R(s_{\text{RIGHT}}, a_{\text{RIGHT}})$
$b_0(s_{\text{LEFT}})$
$b_0(s_{\text{RIGHT}})$

Figure 4: Encoding the tiger problem as a vertex-colored graph. Two vertices have the same color if and only if their shapes and fillings are the same.

phism (GA) problem.

A vertex-colored graph G is specified by $\langle V, E, C, \psi \rangle$, where V denotes the set of vertices, E denotes the set of edges $\langle v_i, v_j \rangle$, C is the set of colors, and $\psi : V \to C$ denotes the color associated with each vertex. An automorphism $\phi : G \to G$ is a permutation π of V with the property that for any edge $\langle v_i, v_j \rangle$ in G, $\langle \pi(v_i), \pi(v_j) \rangle$ is also an edge in G, and for any vertex v_i in G, $\psi(v_i) = \psi(\pi(v_i))$. GA is the problem of finding an automorphism of G (Bondy & Murty 1976). The computational complexity of GA is known to be in NP, but neither known to be in P nor NP-complete (Garey & Johnson 1979). However, GA can be solved quite efficiently in practice.

We can encode a POMDP as a vertex-colored graph in order to apply graph automorphism algorithms[1]. First, for every state s in the POMDP, we prepare vertex v_s (called state vertex) and make every vertex share the same color, *i.e.*, $\forall s \in S$, $\psi(v_s) = c_{\text{state}}$. We proceed with the same manner for every action (action vertex), next state (next state vertex), and observation (observation vertex), *i.e.*, $\forall a \in A$, $\psi(v_a) = c_{\text{action}}$, $\forall s' \in S$, $\psi(v_{s'}) = c_{\text{state'}}$, and $\forall z \in Z$, $\psi(v_z) = c_{\text{obs}}$. We choose distinct colors for $c_{\text{state}}, c_{\text{action}}$, $c_{\text{state'}}$ and c_{obs} so that we can prevent mapping a state vertex to an action vertex, *etc.* Next, for every triplet (s, a, s') in the POMDP, we prepare vertex $v_{T(s,a,s')}$ that represents the transition probability $T(s, a, s')$ (transition probability vertex), and assign colors so that two vertices share the same color if and only if the transition probabilities are the same, *i.e.*, $\forall(s, a, s'), \forall(s'', a', s''')$, $\psi(v_{T(s,a,s')}) = \psi(v_{T(s'',a',s'')})$ if and only if $T(s, a, s') = T(s'', a', s''')$. We connect the the transition probability vertex $v_{T(s,a,s')}$ to the corresponding state and action vertices, $v_s, v_{s'}$ and v_a. We proceed with the same manner for observation probabilities, rewards, and initial belief. Finally, we prepare edges between the state vertex and the next state vertex that correspond to the same state. Figure 4 shows the result of the graph encoding process for the tiger problem.

The encoded graph is sparse: the number of vertices in the graph is $2|S|+|A|+|Z|+|S|^2|A|+|S||A||Z|+|S||A|+|S|$, and the number of edges is $3|S|^2|A|+3|S||A||Z|+2|S||A|+|S|$. Hence, the number of edges is linear in the number of vertices, and typical GA algorithms quickly find automorphisms in benchmark POMDP domains. In our study, we used nauty (McKay 2007) for experiments.

Application to Point-Based Algorithms

Point-based value iteration (PBVI) is an approximate POMDP algorithm using finite samples of belief points for value updates (Pineau, Gordon, & Thrun 2006). The belief point at time t is the probability distribution over the states given a history of actions and observations, which is updated from the belief point at time $t-1$ incorporating action at time $t-1$ and observation at time t

$$b_t(s) = (1/C) O(s, a_{t-1}, z_t) \sum_{s' \in S} T(s', a_{t-1}, s) b_{t-1}(s')$$

[1]We can also formulate artibrary GA as a POMDP symmetry finding problem, although such hardness proof is omitted here.

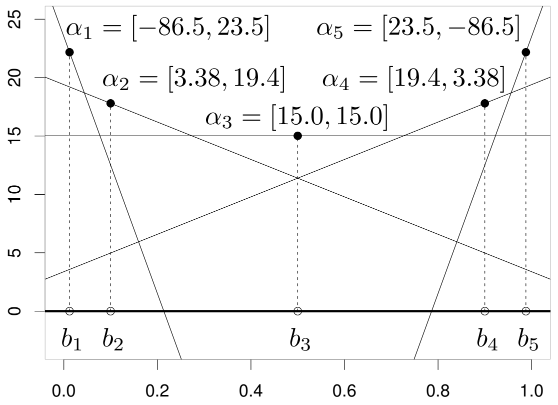

Figure 5: Value function of tiger problem obtained by PBVI with 5 belief points. b_1 and b_5 are symmetric, hence the corresponding α-vectors α_1 and α_5 are symmetric. The same argument applies to b_2 and b_4. Although the illustration uses an approximate value function from PBVI, the value function from exact methods will show the same phenomenon.

where C is the normalizing constant. We will use the notation $b_t = \tau(b_{t-1}, a_{t-1}, z_t)$ to denote the update equation. In order to compute the optimal value at belief point b, we apply dynamic programming (referred as backup operation) to compute t-step value function from $(t-1)$-step value function

$$V_t(b) = \max_{a \in A} \left[R(b,a) + \gamma \sum_{z \in Z} P(z|a,b) V_{t-1}(\tau(b,a,z)) \right]$$

where $R(b,a) = \sum_{s \in S} R(s,a)b(s)$. The value function at each time step is represented by a set of α-vectors so that $\Gamma_t = \{\alpha_0, \ldots, \alpha_m\}$, and the value at a particular belief point b is calculated as $V_t(b) = \max_{\alpha \in \Gamma_t} \sum_{s \in S} \alpha(s)b(s)$. Hence, the backup operation can be formulated as

$$V_t(b) = \max_{a \in A} \Big[\sum_{s \in S} R(s,a)b(s) + \gamma \sum_{z \in Z} \max_{\alpha' \in \Gamma_{t-1}} \sum_{s,s' \in S} T(s,a,s')O(s',a,z)\alpha'(s')b(s) \Big]$$

In finding Γ_t for V_t, PBVI generates intermediate sets $\Gamma_t^{a,z}, \forall a \in A, \forall z \in Z$, whose elements are calculated by

$$\alpha_i^{a,z}(s) = \gamma \sum_{s' \in S} T(s,a,s')O(s',a,z)\alpha_i(s'), \forall \alpha_i \in \Gamma_{t-1}$$

PBVI then constructs Γ_t^a, $\forall a \in A$, whose elements are calculated by

$$\alpha_b^a(s) = R(s,a) + \sum_{z \in Z} \left[\underset{\alpha \in \Gamma_t^{a,z}}{\operatorname{argmax}} (\alpha \cdot b) \right](s), \ \forall b \in B$$

where B is the finite set of belief points. In principle, the backup operation has to be done on all possible belief points in order to obtain exact optimal value function. However, PBVI performs the backup operation only on the selected

```
Procedure: $\Gamma_t = \text{backup}(B, \Gamma_{t-1}, \Phi)$
  for each $a \in A, z \in Z, \alpha_i \in \Gamma_{t-1}$ do
    for each $s \in S$ do
      $\alpha_i^{a,z}(s) = \gamma \sum_{s' \in S} T(s,a,s')O(s',a,z)\alpha_i(s')$
    end for
    $\Gamma_t^{a,z} = \cup_i \alpha_i^{a,z}$
  end for
  $\Gamma_t = \{\}$
  for each $b \in B$ do
    for each $a \in A, s \in S$ do
      $\alpha_b^a(s) = R(s,a) + \sum_{z \in Z} \text{argmax}_{\alpha \in \Gamma_t^{a,z}} (\alpha \cdot b)$
    end for
    $a^* = \text{argmax}_a (\alpha_b^a \cdot b)$
    $\alpha_b = \alpha_b^{a^*}$
    if $\alpha_b \notin \Gamma_t$ then
      $\Gamma_t = \Gamma_t \cup \alpha_b$
      for each $\phi = \langle f, g, h \rangle \in \Phi$ do
        if $f(\alpha_b) \notin \Gamma_t$ then
          $\Gamma_t = \Gamma_t \cup f(\alpha_b)$
        end if
      end for
    end if
  end for
```

Figure 6: The backup operation of PBVI taking into account Φ, the set of all symmetries.

belief points B. The set of α-vectors for V_t is finally obtained by $\Gamma_t = \cup_{a \in A} \Gamma_t^a$.

The automorphisms of POMDPs reveal the symmetries in belief points and α-vectors; Given a POMDP M with automorphism $\phi = \langle f, g, h \rangle$, let Γ^* be the set of α-vectors for the optimal value function. If b is a reachable belief point, then $f(b)$ is also a reachable belief point (by a slight abuse of notation, $f(b)$ is the transformed vector of b whose elements are permutated by f). Also, if $\alpha \in \Gamma^*$, then $f(\alpha) \in \Gamma^*$ (Proofs are provided in the appendix). Figure 5 shows the symmetric relationships of belief points and α-vectors in the tiger problem.

We can modify PBVI to take advantage of the symmetries in belief points and α-vectors: First, when we sample the set of belief points, one of the heuristics used by PBVI is to select the belief point with the farthest L_1 distance from any belief point already in B. However, the belief point selected by such heuristic may have a symmetric belief point already in B. Thus, we can modify the heuristic to exclude the *redundant* belief points by calculating the distances between all possible symmetric images of belief points. Second, since B will exclude redundant belief points, we can modify the backup operation to include symmetric images of α-vectors into Γ_t^a. Figure 6 shows the pseudo-code for performing the symmetric backup operation.

There is a small but important issue regarding symmetric backup of α-vectors; some of the belief points will have the same symmetric image, *i.e.*, $b = f(b)$. For these belief points, it is often unnecessary to add $f(\alpha_b)$ into Γ_t, since $f(\alpha_b)$ is relevant to the belief point $f(b)$ but b and $f(b)$ are the same! Hence, it is advisable to identify which au-

Problem	$\|S\|$	Min $\|S\|$	$\|V\|$	nauty exec time	$\|\Phi\|$
Tiger	2	2	39	0.004 s	2
Tiger-grid	36	35	9814	0.061 s	4
2-city-ticketing ($p_{\text{err}} = 0$)	397	397	1624545	31.872 s	4
2-city-ticketing ($p_{\text{err}} = 0.1$)	397	397	1624545	23.873 s	4
3-city-ticketing ($p_{\text{err}} = 0$)	1945	1945	61123604	2585.770 s	12
3-city-ticketing ($p_{\text{err}} = 0.1$)	1945	1945	61123604	2601.543 s	12

Figure 7: Model minimization and graph automorphism results on benchmark problems. $|S|$ is the number of states in the original model, Min $|S|$ is the number of states in the minimized model, $|V|$ is the number of vertices in the graph encoding of the model, and $|\Phi|$ is the number of automorphisms found by nauty including the identity mapping.

Problem	Algorithm	$\|B\|$	$\|\Gamma\|$	Iter	Exec time	$V(b_0)$	ϵ
Tiger	PBVI	19	5	89	0.07 s	6.40	0.01
	Symm-PBVI	10	5	89	0.05 s	6.40	
Tiger-grid	PBVI	590	532	88	359.69 s	0.80	0.03
	Symm-PBVI	300	529	85	196.09 s	0.80	
2-city-ticketing ($p_{\text{err}} = 0$)	PBVI	51	5	167	157.80 s	8.74	0.02
	Symm-PBVI	17	5	168	57.60 s	8.74	
2-city-ticketing ($p_{\text{err}} = 0.1$)	PBVI	104	9	167	546.04 s	7.76	0.02
	Symm-PBVI	30	10	167	201.97 s	7.73	
3-city-ticketing ($p_{\text{err}} = 0$)	PBVI	261	37	91	43094.32 s	8.08	1.00
	Symm-PBVI	36	42	91	9395.06 s	8.08	
3-city-ticketing ($p_{\text{err}} = 0.1$)	PBVI	275	39	91	43286.92 s	6.95	1.00
	Symm-PBVI	30	133	91	16791.17 s	6.94	

Figure 8: Performance comparisons of the PBVI algorithm with automorphisms. Symm-PBVI is the PBVI algorithm exploiting the automorphisms, *i.e.*, symmetric belief collection and symmetric backup. $|B|$ is the number of belief points given to the algorithms, $|\Gamma|$ is the number of α-vectors comprising the policy, Iter is the number of iterations until convergence, $V(b_0)$ is the average return of the policy starting from initial belief b_0, and ϵ is the convergence criteria of each algorithm for running until $\max_{b \in B} |V^{(n)}(b) - V^{(n-1)}(b)| \leq \epsilon$. All $V(b_0)$'s are within the 95% confidence interval of the optimal.

tomorphisms yield $b \neq f(b)$ for each belief point b, and include symmetric images of α-vectors only for these automorphisms for further performance improvement of PBVI.

Experiments

Before we demonstrate the performance gain of the PBVI algorithm by using symmetric backup operator, we first report test results for the existence of automorphisms in standard POMDP benchmark problems. Most of the benchmark problems are already compact in the sense that the model minimization algorithm was not able to further reduce the size in most of the problems. For the tiger-grid problem, we were able to reduce the size and find symmetries. For the tiger problem, we were not able to reduce the size, but still find symmetries.

We further tested for automorphism existence on larger domains for the spoken dialogue management problems by Williams, Poupart, & Young (2005). In this domain, the user is trying to buy a ticket to travel from one city to another city, and the machine has to request or confirm information from the user in order to issue the correct ticket. We instantiated the problem for 2 and 3 possible cities, and for two different rates of speech recognition errors p_{err}, where $p_{\text{err}} = 0$ assumes no speech recognition error and $p_{\text{err}} = 0.1$ assumes an error rate at 10%. Note that even in the case where $p_{\text{err}} = 0$, the problem is still a POMDP since the user may provide partial information about the request (*e.g.*, origin city only). All of these problems could not be reduced in size, but still had symmetries. Regardless of the value of p_{err}, the graphs encoding the POMDP problems were exactly the same. The small differences in the nauty execution times may be due to the differences in the orderings of the vertices of the graph. Figure 7 summarizes the result of automorphism finding experiments.

Next, we experimented with the PBVI algorithm on the above benchmark problems using the automorphisms found by nauty. First, we sampled a fixed number of symmetric belief points (*e.g.*, 300 for the tiger-grid) and ran the symmetric version of PBVI. We then checked the number of unique belief points if the symmetric belief points were to be expanded by the automorphisms. We set this number (*e.g.*, 590 for the tiger-grid) as the number of belief points to be used by PBVI, and ran PBVI in the same setting without automorphisms. Note that our implementation of PBVI differs from the original version in that the original PBVI *interleaves* the belief point exploration and the value iteration, rather than fixing the belief points in the onset of execution. This was to analyze the efficiency of the symmetric backup isolated from the effects of symmetric belief point exploration. Figure 8 shows the results of the experiments. In summary, automorphisms help to significantly improve the performance of PBVI in running time without sacrificing the quality of policy.

Conclusion and Future Work

In this paper, we have presented a graph theoretic framework for finding symmetries in POMDPs. Given a POMDP, we can cast the problem of finding a symmetry as a graph automorphism (GA) problem. Our approach is an extension to the traditional model minimization methods in the sense that there are cases where minimal models still exhibit a number of symmetries. Although the problem of finding a graph automorphism is not yet known to be tractable, we were able to efficiently compute automorphisms of graphs with more than 60 million vertices, due to sparseness of the encoded graph. We also demonstrated how thus found symmetries could be exploited for speeding up conventional POMDP algorithms, especially the point-based backup operator in PBVI. Our approach is typically effective in the types of problems where there are multiple goals and the symmetries exist among these goals.

Although we have demonstrated the effectiveness of our approach only in the context of PBVI, we believe that our approach can be applied to a wide variety of POMDP algorithms as well. For example, heuristic search value iteration (Smith & Simmons 2005) algorithm can be extended to exploit symmetries in belief point exploration and value function bound updates. Our approach can also be used to find symmetries for factored POMDPs (Boutilier & Poole 1996) or DEC-POMDPs (Bernstein *et al.* 2002) by working in the level of variables or agents.

Finally, we are also working on extending *approximate* equivalence in model minimization (Givan, Leach, & Dean 2000) to *approximate* symmetries in order to achieve further speed up of POMDP algorithms by trading off the potential loss in the optimality of the solution and the speed gain in the algorithm.

Acknowledgments

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant (R01-2007-000-21090-0) funded by the Korea government (MOST)

Appendix

Theorem If b is a reachable belief point, then $f(b)$ is also a reachable belief point.

Proof If b is a reachable point from initial belief point b_0 by executing a policy tree, $f(b)$ can also be reached by executing the policy tree where action nodes are relabeled as $g(a)$ and the observation edges are relabeled as $h(z)$. This is because the automorphism ensures that $T(s,a,s') = T(f(s), g(a), f(s'))$, $O(s,a,z) = O(f(s), g(a), h(z))$, and $b_0(s) = b_0(f(s))$.

Theorem If $\alpha \in \Gamma^*$, then $f(\alpha) \in \Gamma^*$.

Proof We prove by induction on time step t in Γ_t. By the definition of automorphism, $R(s,a) = R(f(s), g(a))$. Hence, if $\alpha \in \Gamma_0$ then $f(\alpha) \in \Gamma_0$.

Suppose that the argument holds for Γ_{t-1}. This implies that $\forall \alpha \in \Gamma_t^{a,z}$, $f(\alpha) \in \Gamma_t^{g(a),h(z)}$ by the definition of $\Gamma_t^{a,z}$. If $\alpha \in \Gamma_t$, then by definition, for some a and b,

$$\alpha(s) = \alpha_b^a(s) = R(s,a) + \sum_{z \in Z} \operatorname*{argmax}_{\alpha' \in \Gamma_t^{a,z}} (\alpha' \cdot b)$$

Consider the symmetric image defined as

$$\alpha(f(s)) = R(f(s), g(a)) + \sum_{h(z) \in Z} \operatorname*{argmax}_{\alpha'' \in \Gamma_t^{g(a),h(z)}} (\alpha'' \cdot f(b)).$$

For each observation $h(z)$, the argmax will select α'' which is the symmetric image of α' selected in the $\operatorname{argmax}_{\alpha' \in \Gamma_t^{a,z}} (\alpha' \cdot b)$. Hence we have $f(\alpha) \in \Gamma_t$.

References

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research* 27(4).

Bondy, J. A., and Murty, U. S. R. 1976. *Graph Theory with Applications*. Elsevier Science.

Boutilier, C., and Poole, D. 1996. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of AAAI-1996*.

Dean, T., and Givan, R. 1997. Model minimization in Markov decision processes. In *Proceedings of AAAI-1997*.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

Givan, R.; Leach, S.; and Dean, T. 2000. Bounded-parameter Markov decision processes. *Artificial Intelligence* 122.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101.

McKay, B. D. 2007. nauty user's guide (version 2.4).

Pineau, J.; Gordon, G.; and Thrun, S. 2003. Policy-contingent abstraction for robust robot control. In *Proceedings of UAI-2003*.

Pineau, J.; Gordon, G.; and Thrun, S. 2006. Anytime point-based approximation for large POMDPs. *Journal of Artificial Intelligence Research* 27.

Ravindran, B., and Barto, A. G. 2001. Symmetries and model minimization in Markov decision processes. Technical Report CMPSCI 01-43, University of Massachusetts, Amherst.

Smith, T., and Simmons, R. 2005. Point-based POMDP algorithms: Improved analysis and implementation. In *Proceedings of UAI-2005*.

Williams, J. D.; Poupart, P.; and Young, S. 2005. Factored partially observable Markov decision processes for dialogue management. In *Proceedings of IJCAI-2005 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Wolfe, A. P. 2006. POMDP homomorphisms. In *Proceedings of NIPS-2006*.