2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Approximate Planning in POMDPs via MDP Heuristic

Yong Lin, Xingjia Lu and Fillia Makedon

***Abstract*— MDP heuristic based POMDP algorithms have been considered as simple, fast, but imprecise solutions. This paper provides a novel MDP heuristic value iteration algorithm for POMDPs. Besides the help of MDP, our algorithm utilizes a weighted graph model for the belief point approximation and reassignment, to further improve the efficiency and decrease the space complexity. Experimental results indicate our algorithm is fast and has high solution quality for POMDP problems.**

I. Introduction

Partially observable Markov decision processes (POMDPs) is a powerful probabilistic model for planning problems with uncertainty. Extensive research works have been taken on POMDPs and value iteration algorithm. We summarize two trends of the POMDP research:

First, POMDP models have been thought as notoriously hard to be solved. In a problem with n physical states, POMDP planners must reason about belief states in an $(n-1)$-dimensional continuous space [3]. The belief space can be exponentially with the number of states. This will lead to an intolerable time complexity. Researchers have tried various approaches to improve the performance.

Second, more robot planning and controlling problems are modeled as POMDP model. Former POMDP research focused on simple problems, like Tiger, Shuttle, 4×4 Grid, Cheese Maze and 4×3 Grid [6] with less than 20 states. These problem configurations try to build a simple approximation of the real world. Later research puts into consideration more complex problems and a closer approximation of the real world, such as Hallway, Hallway2, Tiger-grid [4] and Tag-domain [8]. They have tens or hundreds of states, with actions and observations also being more than earlier simple POMDP problems. More and more researchers choose to believe that POMDPs is an excellent mathematical approach to model real world control problems. Therefore, with the development of novel problem models, POMDP algorithms have to be reevaluated and improved synchronously.

The solution quality is also an importance consideration of POMDP algorithms. After all, the final purpose of POMDP algorithms is to find optimal policies for the planning and execution. The approach of exact belief space is often considered in small POMDP problems. However, more and more real world applications appear to have hundreds and thousands of states, approximate algorithms seem to be more feasible in solving these problems.

Yong Lin, Xingjia Lu are with the College of Science, Ningbo University of Technology, Ningbo, Zhejiang, CN (email: {ylin, xjlu}@nbut.edu.cn).

Fillia Makedon is with the Department of Computer Science and Engineering, University of Texas at Arlington, Texas, USA (Email: makedon@uta.edu).

A POMDP solution with a finite set of belief points is called a point-based algorithm [8]. We build the algorithm as a learning process by explore and exploit of the belief space. A belief graph approach is introduced for the approximation of belief states. Policies from MDP are taken as heuristic to solve the POMDP problems.

Several MDP heuristic based algorithms from prior research exhibit poor solution quality. The results of QMDP [4] sometimes cannot achieve goal in POMDP problems. The most likely state (MLS) [2] algorithm also does not hold history information. From our experiment, MLS is very fast for several POMDP problems, but it cannot work well in the POMDP problem like Hallway2.

In this work, we make use of MDP heuristic in a different way. Historical belief states are stored in the belief space, optimal policies are computed by value iteration. The use of MDP heuristic improves the performance. We make the belief space size adjustable simply by configuration parameters. The experimental results have verified the robustness and preciseness of the algorithm.

II. Technical Preliminary

A. POMDP Model

A Markov decision process (MDP) defines the model an agent interacting synchronously with a fully observable environment. This is described as a tuple $\mathbb{T}_{\mathfrak{mdp}} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $T(s, a, s')$ is the transition probability from state s to s' using action a, $R(s, a)$ defines the reward when executing action a in state s, and γ is the discount factor. The optimal state-action mapping for the t^{th} step, denoted as π_t^*, can be calculated by the optimal $(t-1)$-step value function V_{t-1}^*:

$$\pi_t^*(s) = \arg\max_a \left[R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V_{t-1}^*(s') \right] \quad (1)$$

A POMDP models an agent's action in uncertainty world. At each time step, the agent needs to make decision based on the gathered historical information in Markov model. A policy is defined as a function of action selection under stochastic state transitions and noisy observations. A POMDP is represented as $\mathbb{T}_{\mathfrak{pomdp}-\mathcal{S}} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, \Omega, R, \gamma \rangle$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a set of actions, $\mathcal{O}$ is a set of observations. In each time step, the agent lies in some state $s \in \mathcal{S}$. After taking an action $a \in \mathcal{A}$, the agent goes into a new state s'. The transition is a conditional probability function $T(s, a, s') = p(s'|s, a)$, which is the probability the agent lies in s', after taking action a in state s. The agent makes observations to gather information, with observation result

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

$o \in \mathcal{O}$. This is modeled as a conditional probability, $\Omega(s, a, o) = p(o|s, a)$.

When belief state is taken into consideration, the original partially observable POMDP model changes to a fully observable MDP model, denoted as $\mathbb{T}_{\mathtt{pomdp}-\mathcal{B}} = \langle \mathcal{B}, \mathcal{A}, \mathcal{O}, \tau, \mathcal{R}, b_0, b_g, \gamma \rangle$, where b_0 is an initial belief state, and b_g is the absorbing states, representing the goals and/or termination states. The $\mathcal{B}$ is the set of belief states, i.e. belief space. The $\tau(b, a, b') = p(b'|b, a)$ is the probability the agent changes from b to b' after taking action a. The $\mathcal{R}(b, a) = \sum_s R(s, a)b(s)$ is the reward for belief state b.

The POMDP framework is used as a control model of an agent. In a control problem, utility is defined as a real-valued parameter to determine the action of an agent in every time step. This is represented as a real-valued reward $R(s, a)$, which is a function of state s and action a. The optimal action selection becomes a problem to find a sequence of actions $a_{1..t}$ to maximize the expected sum of rewards $E\left(\Sigma_t \gamma^t R(s_t, a_t)\right)$. In this process, what we concern is the control effect, achieved from the relative relationship of the values. When we use a discount factor γ, the relative relationship remains unchanged, but the values can converge to a fixed number. When states are not fully observable, the goal becomes maximizing expected reward for each belief. The n^{th} horizon value function can be built from the previous value V_{n-1} using a *backup* operator H, i.e. $V = HV'$. The value function is formulated as the following Bellman equation

$$V(b) = \max_{a \in \mathcal{A}} [\mathcal{R}(b, a) + \gamma \sum_{b' \in \mathcal{B}} \tau(b, a, b')V(b')] \quad (2)$$

Here, b' is the next step belief state,

$$b'(s) = b_t(s') = \eta \Omega(s', a, o) \Sigma_{s \in \mathcal{S}} T(s, a, s') b_{t-1}(s) \quad (3)$$

where η is a normalizing constant.

When optimized exactly, this value function is always piece-wise linear and convex in the belief space.

B. POMDP Solutions

Unlike MDPs, in which we can easily find out the optimal policies. The uncertainty property and the scale of problem models affect the solution of POMDP. Various approaches have been tried in prior works. Algorithms using the historical information to find the optimal policies can be classified into value iteration and policy iteration.

Value iteration is an algorithm originated from the utility theory. Some early algorithms, such as Witness algorithm [3], consider the use of exact value function for the belief states. Unfortunately, the time and space complexity of exact value iteration makes it only suitable to solve POMDPs with dozens of states. More researchers try to find approximate methods for value iteration. A smooth and differentiable approximation method called SPOVA is proposed in [7]. The Boyen-Koller approximation algorithm for belief states is proposed in [1]. It requires the model to be specified as a dynamic Bayesian network. In [13], belief states are approximated by sets of (weighted) samples drawn from particle filter. When a new belief state is encountered, its Q-value is obtained by finding k nearest neighbors from historical record, with result being the linearly averaging of the Q-values.

MDP heuristic is considered in Q_{MDP} [4] to solve POMDPs. The Q value for belief state b is estimated as $Q_a(b) = \sum_s b(s)Q_{MDP}(s, a)$. Policies within Q_{MDP} do not take actions to gain information, and decisions are made under one-step uncertainty. Therefore, Q_{MDP} can lead to loops of belief states [4]. Several methods are considered to improve the MDP heuristic approach. The most-likely-state approximation computes the most likely state $x^* = \arg\max_x b(x)$, and defines $\pi(b) = \pi^{MDP}(x^*)$. A transition entropy Q-MDP algorithm is proposed in [6]. From the experimental results of Hallway2 problem, the goal achievement percentage is 63.7%, comparing with 3.9% for Q-MDP.

Reachable belief state based algorithms (PBVI [8], Randomized PBVI [12]) are considered as a substitute for each belief update. PBVI produces new beliefs for every action in a single step forward trajectory. It only keeps one belief state which is farthest away from any points already in the belief set $\mathcal{B}$. The randomized PBVI initiates reachable belief states by randomly exploring the environment. Instead of backing up every $b \in \mathcal{B}$, it backs up random belief states. Classification and pattern adaptation are used in [9] to find a decomposition of belief space into a small number of belief features. The planning is taken over a low-dimensional space by discerning features and using standard value iteration to find policies over discrete beliefs. However, utilizing the classification technique on the belief spaces, the efficiency issue needs to be taken into consideration. Another is the cluster approach to aggregate states [14], according to the optimal MDP values. The soft clustered belief space is then projected onto the POMDP.

III. Belief Value Iteration with MDP Heuristic

In POMDPs, since the world is uncertain, the process to find optimal solution is a tradeoff of exploration and exploitation. By exploring current space, we try to find out the optimal solution. By exploiting unknown space, new and better paths to the goal could be found.

A. Belief States Approximation

Belief states are continuous probabilities representing the current realization of an uncertain environment. A belief state is a sufficient statistic of the history. Thus, the POMDP model can be solved using $\mathbb{T}_{\mathtt{pomdp}-\mathcal{B}}$. The belief states approximation becomes an estimation from an infinite continues belief space to a finite discrete belief space, denoted as $\widetilde{\mathcal{B}} = SE(\mathcal{B})$, where SE is the belief state estimation function. The solution becomes $\mathbb{T}_{\mathtt{pomdp}-\widetilde{\mathcal{B}}} = \langle \widetilde{\mathcal{B}}, \mathcal{A}, \mathcal{O}, V, \Pi, \tau, \mathcal{R}, b_0, b_g, \gamma \rangle$, where $V(s)$ is the optimal value derived from $\mathbb{T}_{MDP}$, $\Pi(b)$ is the optional set of optimal actions for belief state b, and $\widetilde{\mathcal{B}}$ is a finite set of belief states, comparing with the infinite set $\mathcal{B}$.

In a point-based approach, a belief space is built using the reachable points of belief states [8]. We adopt this kind of reachable belief state points. A belief state $b \in \mathcal{B}$ is

a probability distribution over discrete state space $s_{1..n}$, $\sum_{s=1}^{n} b(s) = 1$. We define a *belief point* as a label to identify the belief state b, denoted as $\widetilde{b}$. The identification of a belief state $b \in \mathcal{B}$ is a classification of belief points in the belief space. Our method is in two stages. In the first stage, we use *binary discrimination* as a rough classification of probability distributions. Let b_1, b_2 be two belief states with the same state space, we can easily get two binary similarities:

1) b_1 and b_2 are ϵ-*binary similar* iff $\forall s, b_1(s) > \epsilon \wedge b_2(s) > \epsilon$, or $b_1(s) \leq \epsilon \wedge b_2(s) \leq \epsilon$;
2) $b_1(s)$ and $b_2(s)$ are $(1-\epsilon)$-*binary similar* iff $\forall s, b_1(s) > 1-\epsilon \wedge b_2(s) > 1-\epsilon$, or $b_1(s) \leq 1-\epsilon \wedge b_2(s) \leq 1-\epsilon$.

If belief states b_1 and b_2 are both ϵ-binary similar and $(1-\epsilon)$-binary similar, then b_1 and b_2 are binary similar, denoted as $b_1 \approx b_2$.

Using binary belief discrimination, the continuous infinite belief space $\mathcal{B}$ is approximated to a finite belief space $\widetilde{\mathcal{B}}$, with size $|\widetilde{\mathcal{B}}| = 2^{|\mathcal{S}|}$. In section III-C, we will introduce an approach for the compact representation of belief states in $\widetilde{\mathcal{B}}$. The binary belief discrimination is a rough but efficient classification for discrete states distribution. Its time complexity is $\mathrm{O}(|\widetilde{\mathcal{B}}||\mathcal{S}|)$.

Although binary discrimination is efficient in approximating the infinite belief space onto a finite horizon, we have to consider a more precise method to match belief states with labeled belief points, which becomes the second stage. There are several mathematical models to represent the difference of distribution, such as K-L divergence. Considering the specific belief states distribution, we adopt the distribution *distance* and *affinity* theory [5]. Let b_1 and b_2 be two belief states defined on the same belief space B, the distance between b_1 and b_2 is $d(b_1, b_2) = \sqrt{2(1-\rho(b_1, b_2))}$, where $\rho(b_1, b_2)$ is the affinity of b_1 and b_2, representing the likeness of the probability distributions,

$$\rho(b_1, b_2) = \sqrt{\sum_{s \in \mathcal{S}} b_1(s) b_2(s)} \tag{4}$$

Matusita [5] proved $\rho(b_1, b_2)$ has the following properties: (i) $0 \leq \rho(b_1, b_2) \leq 1$; (ii) $\rho(b_1, b_2) = 1$ iff $b_1 = b_2$.

In Algorithm 1, we list the process for the estimation of belief state b', which is the next belief state computed from b. If b' is an existing belief state in $\widetilde{\mathcal{B}}$, the algorithm returns its belief point label. For a new belief point, we will insert b' in $\widetilde{\mathcal{B}}$.

B. The MHVI Algorithm

As discussed above, the MDP heuristic based belief value iteration (MHVI) algorithm utilizes an alternative tuple $\mathbb{T}_{POMDP-\widetilde{\mathcal{B}}}$ to model the POMDP problems. Solution for the POMDP problem becomes a process to compute the tuple. Besides the elements provided in Section III-A, we introduce other elements in the following. The Π can be obtained from the result of $\mathbb{T}_{MDP}$. We will provide the method to compute $\mathcal{B}$ and τ later. With the above

Algorithm 1 Belief States Estimation

```
search b' in {b_x | τ(b, a, b_x) > 0} by binary discrimination
if b' ∈ {b_x} then
    find a best match belief point y by belief affinity
else
    search b' in B̃ − {b_x} by binary discrimination
    find a best match belief point y by minimal distribution
    distance or belief affinity
    if not found then
        insert b' into B̃
    end if
end if
```

information, the value for belief point can be computed by iterating equation (2). The approximate belief space $\widetilde{\mathcal{B}}$ is initialized as $|\mathcal{O}| + 2$ belief points (one belief point for each observation, plus b_0 and b_g), denoted as $\widetilde{\mathcal{B}}_{0..|\mathcal{O}|+1}$. Let belief point created by observation o be $\widetilde{\mathcal{B}}(o, s)$, $\widetilde{\mathcal{B}}(o, s) = \eta\Omega(s, \forall a \in \mathcal{A}, o)$. Unlike other belief points, during the entire algorithm, the belief points $\widetilde{\mathcal{B}}_{0..|\mathcal{O}|+1}$ keep fixed without change.

On initialization, τ has a single entry, $\tau(b_g, \forall a \in \mathcal{A}, b_0) = 1$. More transitional relationships can be learned during the explore-exploit process of value iteration.

Algorithm 2 MHVI Explore-exploit Iteration

```
repeat
    b ← b' of previous step
    for any a ∈ A do
        exploit belief point b' from b by simulate one step forward
        trajectory using action a
        insert new belief point b' to B̃
        update τ by entry τ(b̃, a, b̃')
        update V(b') = b' × V(S)
        if a ∈ Π(b) then
            compute V_a(b) using equation (2)
        end if
    end for
    find b' by π = arg max_a V_a(b)
until converge
```

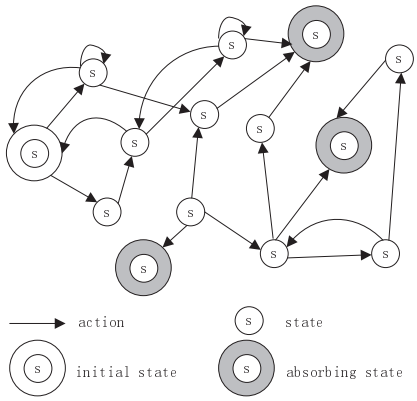
As is shown in Algorithm 2, MHVI assumes the optimal policy π to be an element of the optimal policy set for belief point $\Pi(\widetilde{b})$, which is obtained from the tuple $\mathbb{T}_{\mathfrak{mdp}}$. We do not constrain the exploit process using the policy set. Therefore, potential unknown belief points can still be found, which will benefit the final solution. Benefit of this constraint for optimal policies over $\Pi(\widetilde{b})$ is, the solution can be more failure-proof. Since $|\Pi(b)| \leq |\mathcal{A}|$, this will also be helpful to decrease the exploration and search time.

For the transition probabilities of belief points, we have

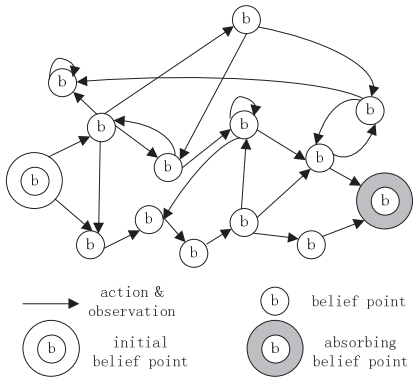
$$\begin{aligned} \tau(\widetilde{b}, a, \widetilde{b'}) &= \Sigma_{o \in \mathcal{O}} \left[\Omega(b, a, o)\mathcal{T}(b, a, o, b')\right] \\ &= \Omega(b, a, o) \end{aligned}$$

where $\mathcal{T}(b, a, o, b') = 1$ if $SE(b, a, o) = \widetilde{b'}$, $\mathcal{T}(b, a, o, b') = 0$ otherwise.

We find out $\Omega(b, a, o)$ by the following processes. First we compute b', $b'(s) = b(s') = \eta\Sigma_{s \in \mathcal{S}} b(s) T(s, a, s')$, where η is a normalizing factor. By $\Omega(b, a, o, s) = \eta b'(s)\mathcal{B}(o, s)$, we can finally obtain the transition probability for $\tau(\widetilde{b}, a, \widetilde{b'})$.



(a) State Graph



(b) Belief Graph

Fig. 1. Example of State Graph and Belief Graph

C. Belief Graph and Belief Space Compression

The transitional relationship for fully observable problems is often represented as a *state graph* (Fig. 1(a)). In a state graph, a *node* is a system state. States in a state space are bijective to nodes in the state graph. The initial state s_0 is the beginning node of a trajectory, and it can be revisited during the traversal. The application appoints which node be the initial state. Fig. 1(a) has only a single initial state. This is because in every trajectory, we have only one initial state. A state graph may have multiple absorbing states. An *edge* in a state graph represents an action $a \in \mathcal{A}$. The transitional relationship is described as the probability $T(s, a, s')$. A state graph may have *loop*, which starts from state s and ends in s. A state graph may contain *cycle*, which passes through state s, goes in some other states, and returns to s. The absorbing states cannot be in a cycle. In a state graph, reward from action a is represented as the weight of edge, i.e. $weight(s, s') = R(s, a, s')$.

The optimal policies of a state graph can form a path $\mathcal{P}_s^*$, starting from s_0 and going into s_g. *There is no loop or cycle in the optimal path $\mathcal{P}_s^*$ of a state graph.* In a specific application, the action may be uncertain, for example, a traveler has 0.8 probability to reach new state and 0.2 probability to stay in current position of each step. This may result in a loop in the real trajectory, but not a loop in the optimal path $\mathcal{P}_s^*$.

When it comes to the partially observable problems, we can use the belief graph to show the transitional relationship (Fig. 1(b)). In a belief graph, a node is a belief point $\widetilde{b}$, which labels a belief state b. By using approximate belief space, the cardinality $|\mathcal{B}|$ changes from infinite to a finite value $|\widetilde{\mathcal{B}}|$. Therefore, belief states from the real belief space are surjective to nodes in belief graph. A belief graph has an initial belief point b_0 and an absorbing belief point b_g. An edge in a belief graph represents the action and observation pair (a, o). A belief graph can contain a loop, which starts from a belief point b and ends at b. A belief graph may contain *cycle*, which passes through belief point b, goes in some other belief points, and returns to b. In a belief graph, the weight for an edge is the reward from the action and observation pair (a, o), i.e. $weight(\widetilde{b}, \widetilde{b'}) = R(b, a, o, b')$.

Each benchmark POMDP problem initialized by former researchers makes an assumption that, states in POMDP are only partially observable, except for the absorbing states. This makes the absorbing belief states identifiable.

The optimal policies of a belief graph form a path $\mathcal{P}_b^*$, starting from b_0 and going into b_g. Suppose a belief graph is built from the exact belief space, i.e. every node represents a different belief state, there will be no cycle or loop in $\mathcal{P}_b^*$. When we use the approximate belief space $\widetilde{\mathcal{B}}$, multiple belief states may be projected onto one belief point by surjection, therefore $\mathcal{P}_b^*$ may contain loop or even cycle. This is different from the optimal path of a state graph. If a path $\mathcal{P}_b^*$ is optimal, we have two *constraint conditions* for the belief graph:

1) (Loop) $\forall \widetilde{b}, \widetilde{b'} \in \mathcal{P}_b^*, \widetilde{b} = \widetilde{b'} \Rightarrow b \neq b'$;
2) (Cycle) $\widetilde{b_1}, \widetilde{b_2}, .., \widetilde{b_k}, \widetilde{b'_1} \in \mathcal{P}_b^*, \widetilde{b_1} = \widetilde{b'_1} \Rightarrow b \neq b'_1$.

Belief space is an important factor influencing the performance of the POMDP algorithm. The computing cost for one iteration can be $\mathrm{O}(|\mathcal{A}||\mathcal{S}|^2 + |\mathcal{A}||\widetilde{\mathcal{B}}||\mathcal{S}|)$. We cannot change $|\mathcal{S}|$ and $|\mathcal{A}|$. For POMDP problem with big state space, the compression of belief space is an effective strategy to maintain good performance. Other algorithms also have some kind of constraints in order to restrict the size of belief space. PBVI appoints only one belief point in each time step. In this part, we introduce a *least visited belief point reassignment* (LVBPR) approach.

At each time step, we record the visited count for each belief point. Let ℓ be the *belief point reassignment level*. We setup a buffer to store the belief points. At every time step, the system has a probability to release the belief point with visited count lower than ℓ, and assign it to the new belief state. Details of LVBPR approach are as following:

Let $\varphi(\widetilde{\mathcal{B}})$ be the buffer belief space with lower-bound $\llcorner\varphi(\widetilde{\mathcal{B}})\lrcorner$, size $|\varphi(\widetilde{\mathcal{B}})|$, and upper-bound $\ulcorner\varphi(\widetilde{\mathcal{B}})\urcorner$. At initial time, let $\llcorner\varphi(\widetilde{\mathcal{B}})\lrcorner = \widetilde{\mathcal{B}}_{|\mathcal{O}|+2}$, and the upper-bound of the buffer belief space be $\ulcorner\varphi(\widetilde{\mathcal{B}})\urcorner = \widetilde{\mathcal{B}}_{|\mathcal{O}|+2+|\varphi(\widetilde{\mathcal{B}})|}$.

In every time step, we verify the visited count of $\llcorner\varphi(\widetilde{\mathcal{B}})\lrcorner$ with ℓ, if it is upper than ℓ, $\llcorner\varphi(\widetilde{\mathcal{B}})\lrcorner \leftarrow \llcorner\varphi(\widetilde{\mathcal{B}})\lrcorner + 1$. We get the available belief point from buffer $\varphi(\widetilde{\mathcal{B}})$. The current location is set to an entry between $\llcorner\varphi(\widetilde{\mathcal{B}})\lrcorner$ and $\ulcorner\varphi(\widetilde{\mathcal{B}})\urcorner$, denoted as $\varphi(\widetilde{\mathcal{B}}_x)$. If the visited count of current entry $\phi = \varphi(\widetilde{\mathcal{B}}_x)$ is lower than ℓ, we release $\tau(\phi, \forall a \in \mathcal{A}, \forall \widetilde{\mathcal{B}}_i \in \widetilde{\mathcal{B}})$ and $\tau(\forall \widetilde{\mathcal{B}}_i \in \widetilde{\mathcal{B}}, \forall a \in \mathcal{A}, \phi)$, with ϕ assigned to the new

belief state. Otherwise we search the available belief point in next buffer entry. At each time step, if $\varphi(\widetilde{\mathcal{B}}_x) \leq \ulcorner \varphi(\widetilde{\mathcal{B}}) \urcorner$,

$$\varphi(\widetilde{\mathcal{B}}_x) = \begin{cases} \varphi(\widetilde{\mathcal{B}}_x) + 1 \sim 1 - p \\ \llcorner \varphi(\widetilde{\mathcal{B}}) \lrcorner \sim p \end{cases}$$

where p is the probability to return to $\llcorner \varphi(\widetilde{\mathcal{B}}) \lrcorner$.

If $\varphi(\widetilde{\mathcal{B}}_x) > \ulcorner \varphi(\widetilde{\mathcal{B}}) \urcorner$, then $\ulcorner \varphi(\widetilde{\mathcal{B}}) \urcorner \leftarrow \ulcorner \varphi(\widetilde{\mathcal{B}}) \urcorner + 1$, the entry of $\varphi(\widetilde{\mathcal{B}}_x)$ is assigned to the current belief state, and $\varphi(\widetilde{\mathcal{B}}_x) \leftarrow \llcorner \varphi(\widetilde{\mathcal{B}}) \lrcorner$. We make use of the parameters ℓ, $\varphi(\widetilde{\mathcal{B}})$ and p to adjust the performance. Therefore, LVBPR is a dynamic and configurable algorithm.

IV. Experimental Evaluation

A. Performance Comparison

To evaluate the performance of MHVI, the benchmark problems of Tiger-grid, Hallway, Hallway2 [4], Tag [8] and RockSample [10] are chosen in the simulation experiments. The experiments are implemented on Intel 2.4GHz CPU 2GB memory by Matlab.

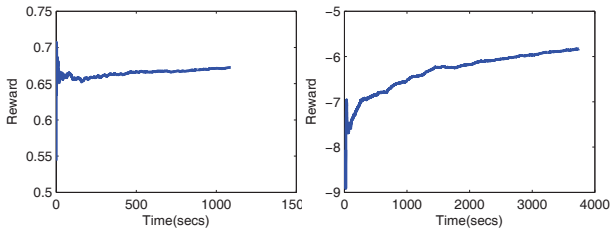
The first experiment aims to get comparable results with other algorithms. Replicating earlier experimental setting, each problem is executed 100 times, terminates after convergence, and max 251 steps. Results are averaged over 100 runs. Table I provides the comparison results with selected previous published algorithms. The comparison is based on the goal completion rate, sum of rewards, policy computation time and number of belief points. We test Persus and MHVI in the same environment. Other algorithms are listed as a reference. The comparison results indicate MHVI achieves competitive performance.

Method	Goal%	Reward	Time(s)	$\lvert\mathcal{B}\rvert$
Tiger-grid ($36s$ $5a$ $17o$)				
QMDP[8]	n.a.	0.198	0.19	n.a.
PBVI[8]	n.a.	2.25	3448	470
Persus*	n.a.	2.34	61.6	93
HSVI2[11]	n.a.	2.30	52	1003
MHVI*	n.a.	3.21	1.67	61
Hallway ($60s$ $5a$ $21o$)				
QMDP[4]	47.4	0.261	0.51	n.a.
PBVI[8]	96	0.53	288	86
Persus*	n.v.	0.51	61.4	105
HSVI2[11]	n.v.	0.52	2.4	147
MHVI*	100	0.71	2.80	72
Hallway2 ($92s$ $5a$ $17o$)				
QMDP[4]	25.9	0.109	1.44	n.a.
PBVI[8]	98	0.34	360	95
Persus*	n.v.	0.35	64.7	124
HSVI2[11]	n.v.	0.35	1.5	114
MHVI*	100	0.65	3.96	66
Tag ($870s$ $5a$ $30o$)				
QMDP[8]	17	-16.9	16.1	n.a.
PBVI[8]	59	-9.18	180880	1334
Persus*	n.v.	-6.17	1542	418
HSVI2[11]	n.v.	-6.36	24	415
MHVI*	100	-7.37	9.24	104
RockSample$[4, 4]$ ($257s$ $9a$ $2o$)				
HSVI1[10]	n.a.	18.0	577	458
HSVI2[11]	n.a.	18.0	0.75	177
MHVI*	n.a.	18.4	6.64	74
RockSample$[5, 5]$ ($801s$ $10a$ $2o$)				
HSVI[10]	n.a.	19.0	10208	699
MHVI*	n.a.	20.4	13.56	83
RockSample$[5, 7]$ ($3201s$ $12a$ $2o$)				
HSVI[10]	n.a.	23.1	10263	287
MHVI*	n.a.	23.0	225	98
RockSample$[7, 8]$ ($12545s$ $13a$ $2o$)				
HSVI1[10]	n.a.	15.1	10266	94
HSVI2[11]	n.a.	20.6	1003	2491
MHVI*	n.a.	21.6	1959	140

n.a.=not applicable, n.v.=not available

TABLE I

Performance Comparison of Different Algorithms (the items labeled by $*$ are tested in our platform)



(a) Hallway2 (b) Tag

Fig. 2. MHVI Performance of Reward

In experiment 1, since the performance of MHVI is good for every POMDP problem, we make them use the same configuration. Rewards and $|\widetilde{\mathcal{B}}|$ are greatly affected by the experimental configuration. For example, in Tag-domain problem, the reward of MHVI is -7.37 and Persus can be -6.17. In our experiments, the reward of MHVI can even be -5.83. The size of $\widetilde{\mathcal{B}}$ is also changeable, i.e. it can be small or large simply by changing the configuration.

B. Stage Convergence and Post-Convergence Iteration

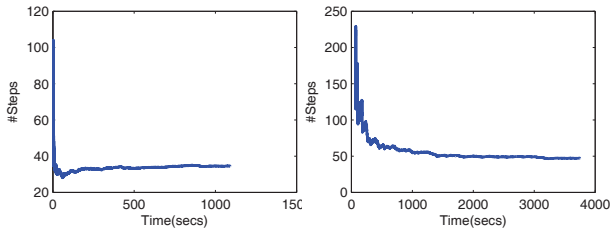
Rewards and $|\widetilde{\mathcal{B}}|$ from the experiment 1 are taken under given experimental configuration. Since we use discount factor γ in Bellman equation, and the state space of MDP is fixed, the value for each state is deterministic to converge after sufficient iteration steps. The iteration process for MHVI serves as a learning process to build the POMDP model. The algorithm does not prohibit the increase of the set $\widetilde{\mathcal{B}}$. However, LVBPR method guarantees the increase of historical information in a slow process.

With discount factor γ, the value function will always converge for the POMDP problems. Given an approximation belief point set $\widetilde{\mathcal{B}}$, since the belief point of absorbing states $b_g \subset \widetilde{\mathcal{B}}$, if the values $\mathcal{V}(\widetilde{\mathcal{B}}_{0..|\widetilde{\mathcal{B}}|})$ converge, the absorbing states are sure to be visited, i.e. the goal has been completed at least one time when values converge.

A belief space is a continuous space containing infinite belief points. For a finite approximate belief space $\widetilde{\mathcal{B}}$ with converged values $\mathcal{V}(\widetilde{\mathcal{B}}_{1..|\widetilde{\mathcal{B}}|})$, when a new belief point is discovered, $\widetilde{\mathcal{B}}$ will be changed to $\widetilde{\mathcal{B}}'$, and values will converge to $\mathcal{V}(\widetilde{\mathcal{B}}_{1..|\widetilde{\mathcal{B}}'|})$. This indicates, unlike MDP, a POMDP model can converge in multiple stages, which we call *stage convergence*. We refer *post-convergence iteration* to the iteration after a convergence stage. Result of post-converge iteration will lead to a new stage of convergence.

In the MHVI algorithm, there are multiple stages of convergence. The second experiment clearly shows this process. From Table I, MHVI converges within no more than 251 steps for all four problems. In experiment 2, the POMDP problems are all executed 10000 time steps and then stops after the values converge. The results indicate they often stop within 50 steps after the original 10000 steps. Fig. 2 is the average of discounted reward.

The average discount reward for Tag problem increases to -5.83 at the 10000 time step. From Tag problem (Fig. 3(b)), the average steps to absorbing states decrease steadily when the time steps increase. This indicates,

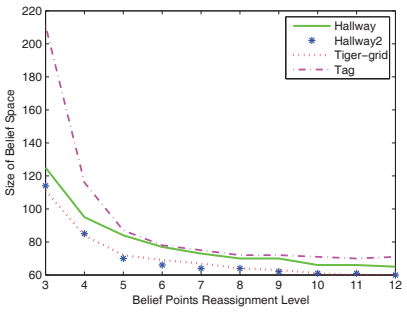


(a) Hallway2 (b) Tag

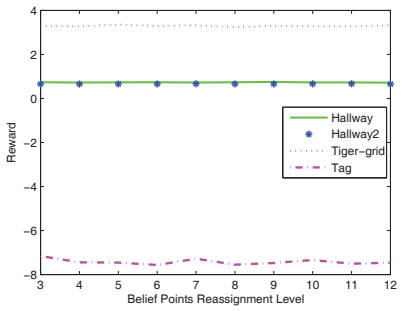
Fig. 3. MHVI Performance of Average Steps to Absorbing States

during the post-convergence iteration, the POMDP finds better paths to the goal. The value iteration serves as a learning process for the Tag POMDP model. However, the increase of reward for Hallway2 (Fig. 3(a)) is not obvious. The average steps to absorbing states keep nearly the same. That is to say, the model is nearly optimal. These experimental results verify that in MHVI, the model comes close to optimal after multiple convergence processes.

In experiment 3, we want to make clear the effect of belief points reassignment level ℓ on the performance. The system takes 500 time steps and stops after convergence. For each ℓ, we execute the system 10 times and average the results. Results of the test problems Hallway, Hallway2, Tiger-grid and Tag are considered for the comparison. When ℓ takes 2, there is no belief points reassignment, the belief points can be over 2000. In Fig. 4(a), $|\widetilde{\mathcal{B}}|$ is high for each problem, and it is lower than 100 for all the problems. Comparing with the results in Table I, the time step is lower than 251, $|\widetilde{\mathcal{B}}| = 107$, and $\ell = 3$. By Fig. 4(b), the reward decreases only a little when ℓ increases. Thus, using ℓ, $|\widetilde{\mathcal{B}}|$ becomes an adjustable parameter.



(a) Size of Belief Space $|\mathcal{B}|$



(b) Reward

Fig. 4. Effect of Belief Points Reassignment Level ℓ on the Performance

V. Conclusions and Future Works

We present the algorithm of MHVI to solve POMDP problems. MHVI adopts MDP heuristic and weighted graph to model the approximate belief space, as well as a dynamic, configurable mechanism to manage graph nodes. Theoretical analysis and experimental results indicate that MHVI is a fast, flexible and robust algorithm.

Potential improvements of the MHVI algorithm are as follows. First, every benchmark problem domain is based on the assumption that there are absorbing states. Whether or not the POMDP model without any goal or termination state has realistic meanings to improve the graph models is still unclear. Second, whether or not a more intelligent management of belief space could benefit the performance for big state space problem domains needs to be determined in further work.

VI. Acknowledgments

This work is sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

References

[1] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *Proceedings of UAI*, pages 33–42, July 24–26 1998.

[2] Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 963–972, 1996.

[3] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2):99–134, 1998.

[4] Michael L. Littman, Anthony R. Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: scaling up. In *Proceedings of ICML*, pages 362–370, 1995.

[5] Kameo Matusita. On the notion of affinity of several distributions and some of its applications. *Ann. Inst. Statist. Math.*, 19:181–192, 1967.

[6] Francisco S. Melo and M. Isabel Ribeiro. Transition entropy in partially observable markov decision processes. In *Intelligent Autonomous Systems*, pages 282–289. IOS Press, 2006.

[7] Ronald Parr and Stuart Russell. Approximating optimal policies for partially observable stochastic domains. In *Proceedings of IJCAI*, pages 1088–1094, 1995.

[8] Joelle Pineau, Geoffrey J. Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proceedings of IJCAI*, pages 1025–1032, 2003.

[9] Nicholas Roy and Geoffrey Gordon. Exponential family pca for belief compression in pomdps. In *Proceedings of NIPS*, pages 1043–1049, 2003.

[10] Trey Smith and Reid G. Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of UAI*, 2004.

[11] Trey Smith and Reid G. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In *Proceedings of UAI*, pages 542–547, 2005.

[12] Matthijs T. J. Spaan and Nikos A. Vlassis. A point-based POMDP algorithm for robot planning. In *Proceedings of ICRA*, pages 2399–2404, 2004.

[13] Sebastian Thrun. Monte Carlo POMDPs. In *Proceedings of NIPS*, pages 1064–1070, 2000.

[14] Yan Virin, Guy Shani, Solomon Eyal Shimony, and Ronen I. Brafman. Scaling up: Solving POMDPs through value based clustering. In *Proceedings of AAAI*, pages 1910–1911, 2007.