A Multi-Criteria Value Iteration Algorithm for POMDP problems

Feng Liu, Tao Zheng, and Xia Hua
National Key Laboratory for Novel Software Technology
Software institute, Nanjing University, Nanjing, China
Email: fengliu@nju.edu.cn, zt@nju.edu.cn, mf1632024@smail.nju.edu.cn

Abstract— **Point-based value iteration algorithms have been deeply studied for solving POMDP problems. However, most of these algorithms explore the belief point set only by single heuristic criterion, thus limit the effectiveness. A novel value iteration algorithm (MCVI) based on multi-criteria for exploring belief point set is presented in the paper. MCVI filters the belief points on which the interval between upper and lower bounds of value function is less than the threshold, and then explores the successor belief point which is farthest away from the explored belief point set. MCVI can improve the effect and efficiency of convergence by guaranteeing that the explored point set is effective and fully distributed in the reachable belief space. Experiment results of four benchmarks show that MCVI can obtain better global optimal solution.**

I. Introduction

The Partially Observable Markov Decision Processes (POMDPs) provide a rich framework for planning and control problems under uncertain environments such as robotic application [1], helping disabled people [2], spoken dialogue systems [3], and so on. However, because solving POMDPs exactly is an NP hard problem, the application of the POMDPs has been limited for a long time. Point-based POMDP algorithms iteratively apply value updates only to a set of representative belief points [4]. These algorithms can significantly improve the overall efficiency by reducing the search space size, therefore become the current research hot spot.

The key content of point-based POMDP methods is about how to explore the reachable belief space. State of the art algorithms explore reachable belief points only based on single standard, thus hamper their efficiency. Approximate solutions based on density standard such as PBVI [4] does not consider value function and cannot control the scale of belief point set. Approximate solutions based on value function such as HSVI [5] and GapMin [6] explore belief point set only according to the difference between lower and upper bounds of the value function instead of the distribution information of the belief points, so they cannot guarantee the efficiency of convergence. This paper proposes a new value iteration algorithm MCVI (Multi-Criteria Value Iteration) to address the need for the convergence efficiency. Firstly, MCVI prunes the points from the explored belief point set on which the interval between upper and lower bounds of value function is less than the threshold after each iteration. Secondly, MCVI only considers successor belief points on which the interval between upper and lower bounds of value function is greater than the threshold to ensure that the value update on the successor belief point can reduce the value uncertainty of the precursor effectively. Lastly, MCVI only explores the belief point which is farthest away from the explored belief point set to make the explored belief point set fully distributed in reachable belief space. By examining the exploration value of reachable belief points during exploring, many meaningless explorations and iterations can be avoided, thus the efficiency and effectiveness of the algorithm can be guaranteed. Experimental results show that comparing with HSVI and PBVI, MCVI can achieve better solution quality with higher ADR and less cost on some large-scale problems.

This paper is structured as follows. Section II introduces the foundations about POMDPs. We also simply review the exploration of PBVI and HSVI algorithm. Section III explains the principles and processes of MCVI algorithm. Section IV reports experiments with four benchmark problems. Section V concludes this paper and lists some issues that we will study in the future.

II. Background

A. POMDP Framework

Formally, a POMDP is a tuple that consists of 8 elements, $(S, A, Z, T, O, R, b_0, \gamma)$. S is a set of states. A is a set of actions. Z is a set of observations. T is a set of conditional transition probabilities between states. O is a set of conditional observations. R is the reward function. b_0 is the initial belief distribution. and γ is the discount factor.

An agent in POMDP framework cannot directly get its own state, but only find the observations from the environments, so it has to plan the next action according to the history sequence $\{a_0, z_1, a_1, z_2, a_2, z_3, \ldots, a_{t-1}, z_t\}$. So a sufficient statistic belief state b is used to maintain the historical information [7]:

$$b_t(s) = P(s_t = s \mid z_t, a_{t-1}, \ldots, z_1, a_0).$$

The probability distribution b can be updated by the Bayes rule:

$$b_t(s') = \tau(b_{t-1}, a_{t-1}, z_t) = \frac{O(s', a_{t-1}, z_t) \sum_{s \in S} T(s, a_{t-1}, s') b_{t-1}(s)}{P(z_t | b_{t-1}, a_{t-1})},$$
$$P(z_t | b_{t-1}, a_{t-1}) = \sum_{s' \in S} O(s', a_{t-1}, z_t) \sum_{s \in S} T(s, a_{t-1}, s')\, b_{t-1}(s).$$

The policy π for a POMDP is to plan actions for belief states: $\pi(b) \rightarrow a$. For a policy π, the expected total reward value of π is

$$V_\pi(b) = E\left[\sum_{t=t_0}^{T} \gamma^{t-t_0} R(b_t, \pi(b_t))\right].$$

The solution for a given POMDP model is to find an optimal policy π^*, which can maximize the expected total reward value. The optimal policy can be obtained by Bellman iteration:

$$V_{k+1}(b) = \max_{a \in A}\left[\sum_{s \in S} b(s) R(s, a) + \gamma \sum_{z \in Z} P(z|b, a) V_k(\tau(b, a, z))\right].$$

The corresponding policy is:

$$\pi_{k+1}(b) = \underset{a \in A}{\operatorname{argmax}}\left[\sum_{s \in S} b(s) R(s, a) + \gamma \sum_{z \in Z} P(z|b, a) V_k(\tau(b, a, z))\right].$$

Smallwood and Sondik have proved that the optimal reward value is a piecewise linear convex function on the belief space for any finite horizon [7], which can be expressed as a collection of vectors:

$$\Gamma_t = \{\alpha_0, \alpha_1, \ldots \alpha_{|\Gamma_t|}\},\ V_t(b) = \max_{\alpha \in \Gamma_t} b \cdot \alpha.$$

POMDP problems can be solved by Bellman value iteration. However, the complexity of the exact update operation from the vectors set Γ_t to Γ_{t+1} is approximate $O(|S|^2|A||\Gamma_t|^{|Z|})$, so the curses of dimensionality and history are the main problem for POMDP exact value iteration algorithms.

B. Point-based Methods for POMDPs

The computational cost for POMDP exact value iteration algorithms is exponential, therefore there has been a lot of work on useful approximation algorithms. In the point-based techniques, a set of representative belief points $R(b_0)$ are selected from the belief space (called the simplex Δ) which contains only some useful belief points reachable from b_0 [4]. Point-based algorithms then only update the vector set to those belief points and obtain the approximate value function with a certain error bound.

The difference between point-based algorithms and exact algorithms on update operation is shown as Fig. 1. Exact algorithms update the vector set on the whole simplex Δ, so they have to cross-sum all vectors in the update operation. Point-based algorithms only update the vector set to the sampled belief point set, thus the optimal vector for an belief point b is determined in the update operation and the reward of action a is calculated by $|Z|$ optimal vectors. Point-based algorithms generate the optimal vector for belief point set by backup (see Alg. 1) operations.

The computational cost of the backup operation on point set B to update Γ_{t+1} from Γ_t is approximate $O(|S|^2|A||Z||B|^2)$. There are two main parts in point-based algorithms: the exploration of the sampled belief point set B, and the backup operation on the belief point set B. In general, the main difference between point-based algorithms is how they collect the belief subset B [8].

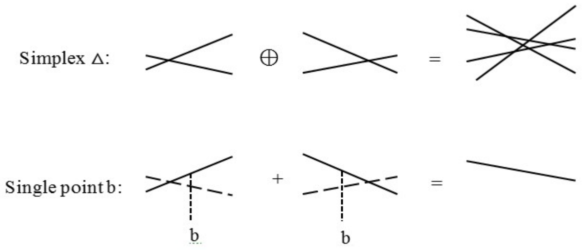

Fig. 1. The Difference Between Update Operation to Single Point b and on the Simplex Δ

```
Algorithm 1: backup
Input: POMDP, B,$\Gamma_t$
Output: $\Gamma_{t+1}$
 $\Gamma_{t+1} \leftarrow \Gamma_t$
for each b ∈ B do
  $a^* \leftarrow \operatorname{argmax}_a(R(a, b) + \gamma \sum_z \max_{\alpha \in \Gamma_t} \sum_{s'} T(s, a, s')\, O(a, s', z)\, b_a^z(s')\alpha$
  $\alpha_b \leftarrow R(a^*, b) + \gamma \sum_z \max_{\alpha \in \Gamma_t} \sum_{s'} T(s, a^*, s')\, O(a^*, s', z)\, b_{a^*}^z(s')\alpha$
  $\Gamma_{t+1} \leftarrow \Gamma_{t+1} \cup \{\alpha_b\}$
end for
```

C. The Exploration of PBVI and HSVI Algorithm

PBVI algorithm explores the belief point set only based on the density standard. For each point in the belief point set, PBVI calculates the distances between its successors and the belief point set, and explores the farthest successor. After expansion PBVI updates the value on the points randomly until convergence.

```
Algorithm 2: PBVIExploration($B$)
  Input: $B$
  Output: $B$
  for each $b \in B$ do
      successor($b$) $\leftarrow \{ b' | b' = b_a^z, \forall a\}$
      $b' \leftarrow \operatorname{argmax}_{b' \in \text{successor}(b), b' \notin B} \|b' - B\|_1$
      $B \leftarrow B \cup \{b'\}$
  end for
  return $B$
```

```
Algorithm 3: HSVIExploration($b, \varepsilon, t$)
  Input: $B$
  Output: $B$
  if $(\overline{V}(b) - \underline{V}(b)) > \epsilon\gamma^{-t}$ do
      $a^* \leftarrow \operatorname{argmax}_a Q^{\overline{V}}(b, a)$
      $z^* \leftarrow argmax_z(P(z|b, a^*)\,(\overline{V}(b^{a^*,z}) - \underline{V}(b^{a^*,z})\,))$
      $b' \leftarrow \tau(b, a^*, z^*)$
      $B \leftarrow B \cup \{b'\}$
      HSVIExploration($b', \varepsilon, t + 1$)
  end if
  return $B$
```

HSVI expands belief point set based on value function standard. It maintains both upper bound $\overline{V}$ and lower bound $\underline{V}$ of value function at the same time. HSVI iterates the value function repeatedly until the reward value function is convergent at the initial belief point *b*$_0$. For each round of exploration, HSVI selects the optimal action with the greatest upper bound of the value function according to the IE–MAX principle and then explores the subsequent belief point of the greatest uncertainty. HSVI's exploration recursively follows a single path down the search tree until satisfying a termination condition based on the width of the bounds interval:

$$\overline{V}(b) - \underline{V}(b) \leq \varepsilon\gamma^{-t}.$$

After each exploration, HSVI updates the upper and lower bounds on the value function to the explored belief point set in reverse order of exploration.

III. A Multi-Criteria Value Iteration Algorithm For POMDPs

A. Multi-Criteria for Exploration

PBVI explores the successor points with the farthest distance to expand the explored belief point set. Although the successor points found in the space are fully dispersed, they cannot guarantee the improvement of value function. HSVI explores the belief point set only according to the difference between upper and lower bounds of value function, which may lead to the problem that the same belief points and paths are repeated explored.

In order to improve the efficiency and effectiveness of exploration, this paper proposes a new value iteration algorithm MCVI, which optimizes the expansion process of belief point set according to multi-criteria. MCVI evaluates the expansion value of each explored points on account of value function heuristic criterion and then selects reasonable successors by both the density criterion and the value function heuristic criterion.

MCVI prunes belief points from explored belief point set on which the interval between upper and lower bounds of value function is less than the threshold after each iteration:

$$Gap_t = max(\ \overline{V}_t(b) - \underline{V_t}(b))/3,$$

$$B_{prune} = \{\ b | b \in B, \overline{V}_t(b) - \underline{V_t}(b) \geq \frac{\varepsilon}{\gamma^{t-1}} \wedge \overline{V}_t(b) - \underline{V_t}(b) \geq Gap_t\}.$$

Where t is the number of the iteration.

For each belief point in B_{prune}, MCVI only takes into account those successor belief points on which the interval between upper and lower bounds of value function is also greater than threshold:

$$successor(b) = \{\ b' | b' = \ b_a^z, \overline{V}_t(b') - \underline{V_t}(b') \geq \frac{\varepsilon}{\gamma^t} \wedge \overline{V}_t(b') - \underline{V_t}(b') \geq Gap_t\}.$$

Then, MCVI only explores the belief points which is farthest away from the explored belief point set to ensure the explored belief point set fully distributed in reachable belief space.

$$\|b' - B\|_1 = \min_{b \in B} \|b' - b\|_1,$$

$$explore(b) = argmax_{b' \in successor(b) \backslash B} \|b' - B\|_1.$$

Compared with PBVI algorithms, MCVI only considers quality belief points based on value function heuristic criterion. On the one hand, MCVI reduces the size of explored belief point set and improves algorithm's adaptability. On the other hand, the value iteration performed only on the effective belief points guarantees the efficiency of the algorithm.

Compared with value function based algorithms such as HSVI, MCVI ensures the explored belief point set distributed as far as possible, so it can avoid the interference of local optimum which may lead to meaningless iterations.

B. MCVI Algorithm

MCVI(see Alg. 4) is a multi-criteria value iteration algorithm, and it optimizes the exploration of belief point set by MCVIExploration (see Alg. 5). MCVI maintains both lower and upper bounds on the value function. The lower bound on the value function $\underline{V}$ is initialized by Blind Policy (see Alg. 6) and iterated by backup function (see Alg. 1). The upper bound of the value function $\overline{V}$ is expressed as a set of belief/value points $(b_i, \overline{v}_i)$. $\overline{V}$ is initialized according to Fast Informed Bound algorithm (see Alg. 7) and updated by adding explored belief point and its upper value:

$$\overline{V}' = \overline{V} \cup \left(b, \max_{a \in A} \overline{Q}\,(b,a)\right),$$

$$\overline{Q}(b,a) = \sum_{s \in S} R(s,a) \cdot b(s) + \gamma \sum_{z \in Z} P(z|b,a)\, \overline{V}(b^{a,z}).$$

The upper value $\overline{V}(b)$ at point *b* is the projection of *b* onto the convex hull formed by $\overline{V}$. The exact upper value of a new belief point can be calculated by dynamic programming techniques. However, the computational cost for the dynamic programming is relatively high. So in MCVI algorithm we calculate the projection approximately by sawtooth algorithm (see Alg. 8).

Algorithm 4: MCVI

```
Input: POMDP
Output: $\pi^*$, $\overline{V}$, $\underline{V}$
  $\underline{V}$ ← $BlindPolicy$(POMDP)
  $\overline{V}$ ← $FIB$(POMDP)
  $B$ ← { $b_0$ }
  while ( $\overline{V}$( $b_0$) - $\underline{V}$( $b_0$) ) > Ɛ do
      B ← MCVIExploration (B)
      $\underline{V}$ ← backup(B, $\underline{V}$)
      $\overline{V}$ ← sawtooth(B, $\overline{V}$)
  end while
```

```
Algorithm 5: MCVIExploration ($B$)
  Input: B
  Output: B
  $GAP = max(\overline{V}(b) - \underline{V}(b))/3$
  $B_{prune} \leftarrow \{ b|b \in B, \overline{V}(b) - \underline{V}(b) \geq \frac{\varepsilon}{\gamma^{t-1}} \wedge \overline{V}(b) - \underline{V}(b) \geq GAP \}$
  for ($b \in \mathrm{B_{prune}}$ )
    $a^* = random(a)$
    $\mathrm{successor}(b) \leftarrow \{ b'|b' = b^z_{a^*}, \overline{\mathrm{V}}(b') - \underline{\mathrm{V}}(b') \geq \frac{\varepsilon}{\gamma^{\mathrm{t}}} \wedge \overline{V}(b') - \underline{V}(b') \geq GAP, \forall z\}$
    $b' \leftarrow \mathrm{argmax}_{b' \in \mathrm{successor}(b), b' \notin B} \|b' - B\|_1$
    $B = B \cup \{ b' \}$
  end for
  return $B$
```

```
Algorithm 6: Blind Policy
  Input: POMDP
  Output: $\underline{V}$
  $\underline{V_a}(\mathrm{s}) \leftarrow min_{a,s} R_a(\mathrm{s}') / (1-\gamma) \ \forall s, a$
  repeat
        $\underline{V_a}(\mathrm{s}) \leftarrow R_a(\mathrm{s}) + \gamma \sum_{\mathrm{s}'} \mathrm{T}(s, a, s') \underline{V_a}(\mathrm{s}') \ \forall a, s$

  until convergence
  return $\underline{V}$
```

```
Algorithm 7: Fast Informed Bound
  Input: POMDP
  Output: $\overline{V}$
  $\overline{V}_a(\mathrm{s}) \leftarrow max_{a,s} R_a(\mathrm{s})/(1-\gamma) \ \forall s, a$
  repeat
     $\overline{V}_a(\mathrm{s}) \leftarrow R_a(\mathrm{s}) +$
              $\gamma \sum_{\mathrm{z}} max_{a'} \sum_{\mathrm{s}'} \mathrm{T}(s, a, s') \, \mathrm{O}(a, s', z) \, \overline{V}_{a'}(\mathrm{s}') \ \forall a, s$
  until convergence
  return $\overline{V}$
```

```
Algorithm 8: sawtooth
  Input: POMDP
  Output: $\overline{\mathrm{V}}$
  for each $\mathrm{b} \in \mathrm{B}$ do
    $\mathrm{V_{con}} \leftarrow \{ \mathrm{b} \mid \mathrm{b(s)} = 1, \exists \mathrm{s} \in \mathrm{S} \})$
    $\mathrm{v^0_b} \leftarrow \sum_{\mathrm{b} \in \mathrm{V_{con}}} \mathrm{v(b)} \cdot \mathrm{b}$
    for each $<\mathrm{b_i}, \mathrm{v_i}> \in \mathrm{V} - \mathrm{V_{con}}$ do
      $\mathrm{c(b_i)} \leftarrow \min_{\mathrm{s:b_i(s)} \neq 0} \mathrm{b(s)/b_i(s)}$
      $\mathrm{f(b_i)} \leftarrow \mathrm{v_i} - \sum_{\mathrm{b} \in \mathrm{V_{con}}} \mathrm{v(b)} \cdot \mathrm{b_i(s)}$
    end for
    $\mathrm{v} \leftarrow \mathrm{v^0_b} + \min_{\mathrm{i}} \mathrm{c(b_i)} \, \mathrm{f(b_i)}$
    $\mathrm{V}' \leftarrow \mathrm{V}' \cup <\mathrm{b}, \mathrm{v}>$
  end for
  return $\overline{\mathrm{V}}$
```

C. Algorithm Analysis

The computational complexity of measuring the expansion value of each belief point in set B for further extending is similar to $\mathrm{O}(|B|)$. The computational complexity of estimating the value of each subsequent points for exploring is about $\mathrm{O}(|\mathrm{B}||\mathrm{A}||\mathrm{Z}|)$.

The density δ_B of a set of belief points B is the maximum distance from any legal belief to B, $\delta_B = \max_{b' \in \Delta} \|b' - \mathrm{B}\|_1$. Then, the error introduced by MCVI's each iteration is at most $\frac{(R_{max} - R_{min})\delta_B}{1-\gamma}$ [4]. Let $b' \epsilon \Delta$ be the point where MCVI makes its worst iteration error, and $b \ \epsilon B$ be the closest sampled belief point to b'. Let α' be the vector which is the exact value function at b', and α be maximal at b.

Then:

$$
\begin{aligned}
\varphi_{\mathrm{iteration}} &\leq \alpha' \cdot b' - \alpha \cdot b' \\
&= \alpha' \cdot b' - \alpha \cdot b' + (\alpha' \cdot b - \alpha' \cdot b) \\
&\leq \alpha' \cdot b' - \alpha \cdot b' + \alpha \cdot b - \alpha' \cdot b \\
&= (\alpha' - \alpha) \cdot (b' - b) \\
&\leq \|\alpha' - \alpha\|_\infty \|b' - b\|_1 \\
&\leq \|\alpha' - \alpha\|_\infty \delta_B \\
&\leq \frac{(R_{max} - R_{min})}{1-\gamma} \delta_B
\end{aligned}
$$

For any belief set B and any horizon n, the error of MCVI is $\left\|\mathrm{V^B_t} - \mathrm{V}^*\right\|_\infty$. The overall error is bounded by $\|\mathrm{V^*_t} - \mathrm{V}^*\|_\infty + \left\|\mathrm{V^B_t} - \mathrm{V^*_t}\right\|_\infty$.The first term is bounded by $\gamma^t \|V^*_0 - V^*\|_\infty$, the second is bounded by the theorem below.

$$
\begin{aligned}
\|V^B_t - V^*_t\|_\infty &= \|J_{HHVI} V^B_{t-1} - J V^*_{t-1}\|_\infty \\
&\leq \|J_{HHVI} V^B_{t-1} - J V^B_{t-1}\|_\infty + \|J V^B_{t-1} - J V^*_{t-1}\|_\infty \\
&\leq \varphi_{\mathrm{iteration}} + \|J V^B_{t-1} - J V^*_{t-1}\|_\infty \\
&\leq \varphi_{\mathrm{iteration}} + \gamma \|V^B_{t-1} - V^*_{t-1}\|_\infty \\
&= \varphi_{\mathrm{iteration}} + \gamma \varphi_{\mathrm{t-1}} \\
&\leq \frac{(R_{max} - R_{min})}{1-\gamma} \delta_B + \gamma \varphi_{\mathrm{t-1}} \\
&\leq \frac{(R_{max} - R_{min})}{(1-\gamma)^2} \delta_B
\end{aligned}
$$

So the maximal error between value function V_t and exact value function in simplex Δ is $\frac{(R_{max} - R_{min})}{(1-\gamma)^2} \delta_B$.

IV. EXPERIMENTS AND ANALYSIS

A. Experimental Setup

We run MCVI, PBVI and HSVI with four well-known benchmarks: Hallway2, Tiger-grid, TagAvoid and UnderwaterNav. Hallway2 and Tiger-grid are classic maze problems. Tag Avoid simulates the chase of robotics. Underwater Navigation is an instance of coastal navigation. The characteristics of POMDP benchmarks are described in Table I, where |S| is the number of states, |A| is the number of actions and |Z| is the number of observations.

We implement MCVI, PBVI and HSVI by the package MCVI. In all experiments the discount factor γ is set to 0.95 and the threshold ε is set to 0.001. We log the iteration time, |B|, |Γ| when a new value function is obtained, then execute actions from b_0 for 100 horizons to simulate the discounted reward, and repeat this simulation 500 times to calculate the average discounted reward (ADR) of the value function. Experiment will be terminated when presupposed ADR or time is reached. We report the highest ADR, the number of vectors and belief states, and the iteration time of each algorithm on each benchmark.

B. Experimental Results

The experimental results of MCVI, PBVI and HSVI are compared in Table II and Table III. In Table II, column 2 lists the ADR for each algorithm and column 3 lists the iteration time it costs. Table III shows the number of vectors and belief points when each algorithm is convergent.

Experimental results show that the convergence speed of MCVI is faster than PBVI and HSVI algorithm, and MCVI algorithm can achieve higher ADR in relatively shorter time as the scale of POMDP problems goes up.

Table I
CHARACTERISTICS OF POMDP BENCHMARKS

problems	|*S*|	|*A*|	|*Z*|
Tiger-grid	36	5	17
Hallway2	92	5	17
TagAvoid	870	5	30
UnderwaterNav	2650	6	103

Table II
THE COMPARISON FOR MCVI, PBVI AND HSVI ON ADR AND TIME

problems	ADR			Time(s)		
	PBVI	HSVI	MCVI	PBVI	HSVI	MCVI
Tiger-grid	2.278	2.231	2.353	375	385	365
Hallway2	0.478	0.481	0.500	115	95	65
tagAvoid	-6.361	-6.205	-5.966	265	125	100
UnderwaterNav	731.503	737.835	743.631	150	105	55

Table III
THE COMPARISON FOR MCVI, PBVI AND HSVI ON \|Γ\| AND \|B\|

problems	|Γ|			|B|		
	PBVI	HSVI	MCVI	PBVI	HSVI	MCVI
Tiger-grid	2861	1762	1302	828	963	728
Hallway 2	395	933	162	226	282	128
tagAvoid	2198	1233	906	633	1865	426
UnderwaterNav	1036	501	432	415	812	238

C. Experimental Analysis

Fig.2 to 5 show the comparison for MCVI, PBVI and HSVI on four problems. The X-axis denotes the iteration time (s) of algorithms and the Y-axis denotes ADR. In all figures, MCVI is denoted by the solid line, PBVI is denoted by the dashed line, and HSVI is denoted by the dot line. The dotted line perpendicular to the X axis indicates the time when MCVI is convergent.

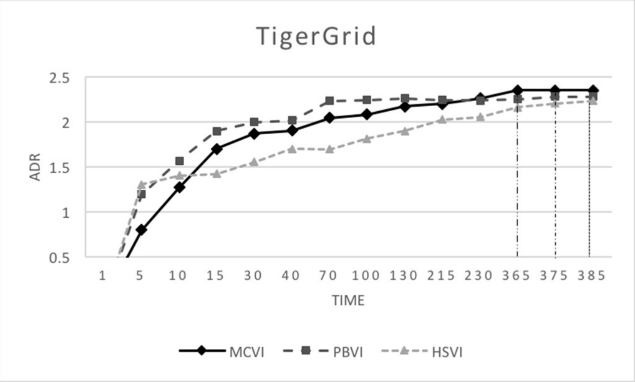

Fig.2. The comparison of ADR for MCVI, PBVI and HSVI on Tiger-grid

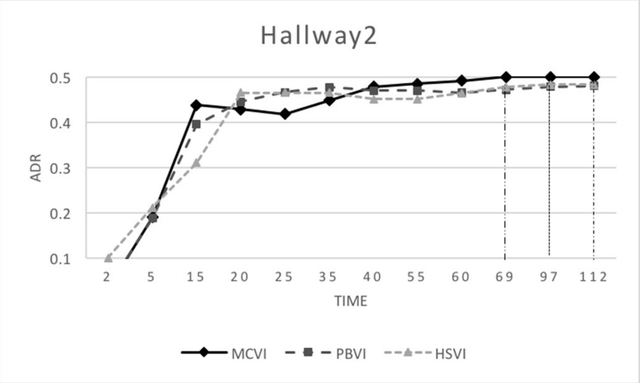

Fig.3. The comparison of ADR for MCVI, PBVI and HSVI on Hallway2

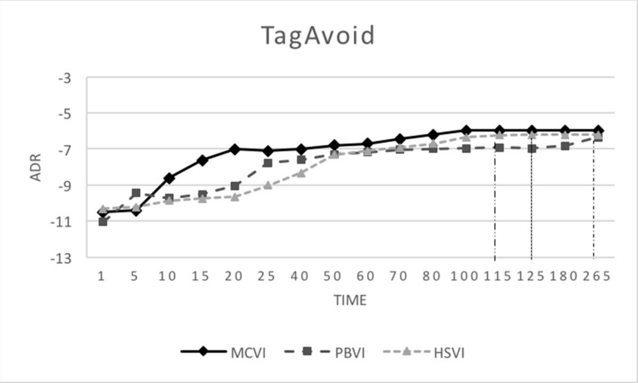

Fig.4. The comparison of ADR for MCVI, PBVI and HSVI on TagAvoid

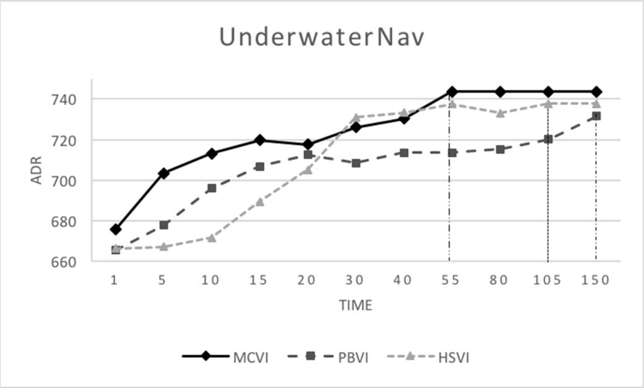

Fig.5. The comparison of ADR for MCVI, PBVI and HSVI on UnderwaterNav

As Tiger-grid is a small-scale problem, in the experiments MCVI converges faster than PBVI and HSVI and achieves slightly higher ADR. In the solution of Hallway2, MCVI converges faster than HSVI by 1.46 times and PBVI by 1.77 times. As shown in Fig. 2 and Fig. 3, due to the less sampling of HSVI than MCVI in each iteration, the convergence rate of HSVI is relatively higher during the early stage of iterations. However, MCVI keeps on exploring distributed belief points which are helpful to the convergence of value function while HSVI spends time in meaningless iteration in the later period, so MCVI can gain better convergence efficiency and effect at last.

In the experiments of TagAvoid, MCVI gains higher ADR than HSVI when the algorithm converges, and the convergence efficiency of the MCVI is 1.25 times faster than that of HSVI. MCVI's ADR is much higher than PBVI, and converges faster than PBVI by 2.65 times. In UnderwaterNav, MCVI also achieves better ADR than HSVI and PBVI. MCVI converges faster than HSVI by 1.91 times and PBVI by 2.73 times.

Fig.6 to 9 show the comparison of the number of vectors in value function |Γ| for MCVI, PBVI and HSVI on four problems. The X-axis represents the iteration time (s) of algorithms and the Y-axis represents the number of vectors.

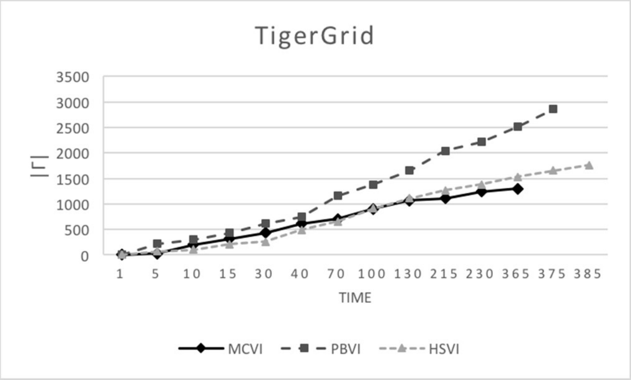

Fig.6. The comparison of |Γ| for MCVI, PBVI and HSVI on Tiger-Grid

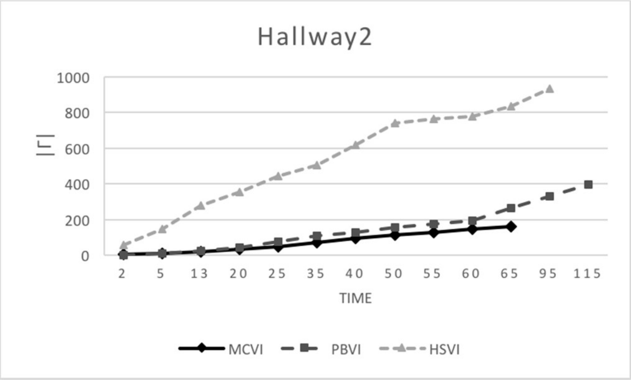

Fig.7. The comparison of |Γ| for MCVI, PBVI and HSVI on Hallway2

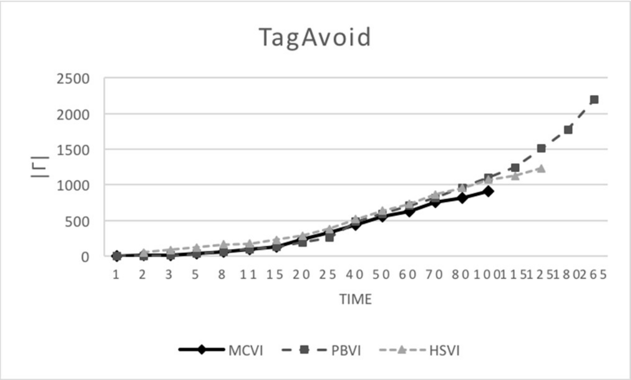

Fig.8. The comparison of |Γ| for MCVI, PBVI and HSVI on TagAvoid

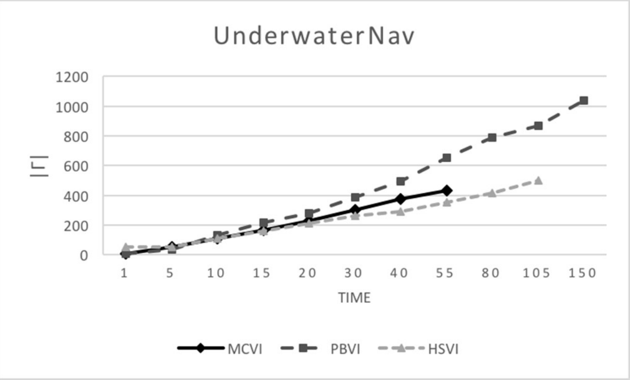

Fig.9. The comparison of |Γ| for MCVI, PBVI and HSVI on UnderwaterNav

These figures show that the vectors number of PBVI and HSVI in each iteration is obviously more than that of MCVI algorithm. So MCVI can achieve better convergence with less value function vectors during the iteration on all benchmark problems, which means that MCVI can obtain more reasonable belief point set through the exploration based on hybrid heuristic criterion.

Although both MCVI and PBVI use density criterion to explore reachable belief point set, MCVI maintains the upper bound of the function value and filters the belief points which do not have exploration value according to the value function criterion, thus guarantees the effectiveness of the sampled belief points. Compared with HSVI, MCVI breadth-first explores belief points by density criterion, thus can avoid the interference of local optimization.

As shown in experimental results, MCVI is not only able to obtain a better ADR value, but also improve the convergence efficiency, especially in the large-scale problem UnderwaterNav. So it is convinced that MCVI is more capable of dealing with large-scale problems. Furthermore, it is also obvious that MCVI can avoid the limitations of single exploration standard, enhance the adaptability to different POMDP problems, and improve the efficiency and effectiveness of value iteration.

V. CONCLUSION

This paper proposes a new algorithm MCVI which makes up the defects of the single exploration standard by using both the density standard and the value function heuristic criterion for exploring reachable belief points. MCVI integrates PBVI and HSVI, and maintains the upper and lower bounds of the value function at the same time. The larger the difference between the upper and lower bounds, the greater the uncertainty of value function at the belief point. So MCVI filters points on which the interval between the upper and lower bounds of value function is less than a given threshold to ensure the sampling effect during exploring the belief point set. On the other hand, MCVI only explore belief point with the most distance from the explored point set, which guarantees that the explored point set is fully distributed in reachable belief space.

In next step, we will try to explore a number of effective successor belief points every time for each belief point in B_{prune} to further improve the efficiency of the exploration of the reachable belief space according to GapMin algorithm.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61373010) and the Fund of State Key Laboratory for Novel Software Technology (Nanjing University, No. ZZKT2016B07).

REFERENCES

[1] T. Smith, “Probabilistic planning for robotic exploration”, Doctoral dissertation, Massachusetts Institute of Technology, 2007

[2] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie and A. Mihailidis, “A decision-theoretic approach to task assistance for persons with dementia”, In Proceedings of the international joint conference on artificial intelligence, IJCAI (pp. 1293–1299), 2005

[3] J. D. Williams, and S. Young, “Partially observable Markov decision processes for spoken dialog systems”, Computer Speech & Language 21(2): 393-422, 2007

[4] J. Gordon, G. Pineau and S. Thrun, “Point-based value iteration: An anytime algorithm for POMDPs”, In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp.1025–1032, 2003.

[5] T. Smith and R. Simmons, “Point-based POMDP algorithms: Improved analysis and implementation”, In Proceedings of the 21th conference on Uncertainty in artificial intelligence, pp.542-547, 2005.

[6] P. Poupart, K. E. Kim and D. Kim, “Closing the Gap: Improved Bounds on Optimal POMDP Solutions”, In Proceedings of the 21st International Conference on Automated Planning and Scheduling, pp.194–201, 2011.

[7] R. D. Smallwood and E. J. Sondik, “The optimal control of partially observable Markov processes over a finite horizon”, Operations Research, 21(5): 1071-1088, 1973.

[8] G. Shani, J. Pineau and R. Kaplow, “A survey of point-based POMDP solvers”, Autonomous Agents and Multi-Agent Systems 27(1): 1-51, 2013.