www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

# POSTERIOR BELIEF CLUSTERING ALGORITHM FOR ENERGY-EFFICIENT TRACKING IN WIRELESS SENSOR NETWORKS

Bo Wu, Yanpeng Feng, Hongyan Zheng
Education Technology and Information Center,
Shenzhen Polytechnic, Shenzhen 518055, Guangdong, China
Emails: wubo@szpt.edu.cn; ypfeng@szpt.edu.cn; zhenghongyan@szpt.edu.cn

***Submitted: Feb. 20, 2014*** ***Accepted: July 2, 2014*** ***Published: Sep. 1, 2014***

***Abstract- In this paper, we propose a novel posterior belief clustering (PBC) algorithm to solve the tradeoff between target tracking performance and sensors energy consumption in wireless sensor networks. We model the target tracking under dynamic uncertain environment using partially observable Markov decision processes (POMDPs), and transform the optimization of the tradeoff between tracking performance and energy consumption into yielding the optimal value function of POMDPs. We analyze the error of a class of continuous posterior beliefs by Kullback–Leibler (KL) divergence, and cluster these posterior beliefs into one based on the error of KL divergence. So, we calculate the posterior reward value only once for each cluster to eliminate repeated computation. The numerical results show that the proposed algorithm has its effectiveness in optimizing the tradeoff between tracking performance and energy consumption.***

**Index terms*: *Partially observable Markov decision processes, wireless sensor networks, target tracking, energy consumption, posterior belief, clustering algorithm.**

## I. INTRODUCTION

The rapid deployment, self-organization and fault tolerance characteristics of wireless sensor networks (WSNs) make them a very promising sensing technique for military, environmental, health, home and commercial applications [1]. However, sensor nodes are usually battery-powered, all operations of the sensor nodes, including the transceiver operations, should be carefully managed to ensure a long operational lifetime [2]. The data processing capabilities and communication bandwidth of sensors nodes are both limited. The conflict of communication and the loss of transmission data cause WSNs to trap into dynamic uncertain environments. The tradeoff between target tracking performance and sensors energy consumption is a challenging problem in the dynamic uncertain environment.

Much effort has been spent in developing efficient power control algorithms to improve system performance even with imperfect channel state information [3, 4, 5, 6]. Recently, the type of target tracking algorithms based on Markov decision process (MDPs) or partially observable Markov decision processes (POMDPs) becomes the focus and hotspot in WSNs [7, 8]. According to the characteristics of the uncertain environment around the sensor nodes, we cast the scheduling problem of target tracking as the partially observable Markov decision processes (POMDPs). Thus target tracking problem of WSNs can be transformed into the optimal policy problem of POMDPs. For instance, the Monte Carlo solution method [9] is developed to use a combination of particle filtering for belief-state estimation and sampling-based Q-value approximation for lookahead. The decision-theoretic approach for dynamic sensor scheduling [10] is presented to optimize the problem in terms of maximizing coverage and improving localization uncertainty, with a focus on tracking a moving object in a network using only a limited number of sensors simultaneously. The smart sleeping policy [11] is proposed to derive a lower bound on the optimal energy-tracking tradeoff for discrete state spaces and continuous Gaussian observations. POMDPs-based target tracking algorithms provide an elegant solution to the optimal tradeoff between tracking performance and energy consumption. However, the existing algorithms usually trap into the curse of dimensionality. They only are suitable for the small scale WSNs systems. When the number of sensors is large or the states are continuous, the POMDPs algorithms could not achieve real-time convergence.

In this paper, we propose a novel posterior belief clustering (PBC) algorithm to solve the tradeoff between target tracking performance and sensors energy consumption. The key insight that can be drawn is that we calculate the posterior reward value only once for each cluster to eliminate repeated computation. Numerical results confirm that PBC can improve the target tracking performance and decrease sensors energy consumption simultaneously.

The rest of this paper is organized as follows. Section II gives the preliminaries of POMDPs. In Section III, we model the target tracking under dynamic uncertain environment using POMDPs. Section IV proposes a novel posterior belief clustering (PBC) algorithm. Experimental results are provided by Section V. Finally, Section VI presents the conclusions of the paper.

## II. POMDPs

Formally, the POMDPs model can be presented as a tuple (*S*, *A*, *Z*, *T*, *O*, *R*). *S* is the set of all the environment states. The state is not directly observable in POMDPs, where an agent can only compute a belief over the state space *S*. *A* is the set of all possible actions. Actions stochastically affect the state of the world. Choosing the right action as a function of history is the core problem in POMDPs. *Z* is the set of all possible observations. Observation is usually an incomplete projection of the world state, contaminated by sensor noise. *T* is the state transition probability distribution, $T : S \times A \rightarrow \prod(S)$. $T(s,a,s')$ represents the probability of ending in state *s'* if the agent performs action *a* in state *s*. *O* is the observation probability distribution, $O(s',a,z)$ is the probability that the agent will perceive observation *z* upon executing action *a* in state *s'*. *R* is the reward function, $R(s,a) : S \times A \times Z \mapsto \mathbf{R}$, is the reward obtained by executing action *a* in state *s*. The objective of POMDPs is to optimize action selection to collect as much reward as possible over time.

In this paper, unless otherwise specified, superscript represents time (*t*) and subscript stands for the specific variables. Such as, $s_i$ represents the *i*th state in the set of *S*, $s^t$ represents the state at time *t*, $P(s^t = s_i)$ represents the probability when state is $s_i$ at time *t*. Sometimes, if necessary, the current state is denoted *s* and the next state is denoted $s'$. For instance, *s* represents the *current* state, and $s'$ represents the *next* state.

The Markov assumption implies that all the historical information (the historical observation set $z^{1:t}$ and the historical action set $a^{0:t-1}$) needed to monitor or predict by a probability distribution over the possible states named a belief state *b* [12]. At each time point *t*, the belief state *b* can be calculated as follows:

$$b^t(s) = P(s^t = s \mid a^{0:t-1}, z^{1:t}) \tag{1}$$

According to the sequence of action $a^{t-1}$ and observation $z^t$, we define the prior belief and the posterior belief respectively [12]. The prior belief state of state *s* at time *t*, denoted $b^{(\cdot t)}(s)$, is the distribution over the state *s* at *t* when action $a^{t-1}$ has been occurred but not observation $z^t$. For the discrete states, the prior belief can be defined as follows.

$$b^{(\cdot t)}(s) = P(s^t \mid a^{0:t-1}, z^{0:t-1}) \tag{2}$$

For the continuous states, the prior belief can be defined as follows.

$$b^{(\cdot t)} = \int_{s \in S} T(s^{t-1}, a^{t-1}, s^t) b^{(\cdot t-1)} ds^{t-1} \tag{3}$$

The posterior belief state of state *s* at time *t* is the distribution over the state *s* at *t* when both action $a^{t-1}$and observation $z^t$ have been occurred. For the discrete states, the posterior belief can be defined as follows.

$$b^{(t\cdot)}(s) = P(s^t \mid a^{0:t-1}, z^{0:t-1}, z^t) \tag{4}$$

Usually, we represent the posterior belief state $b^{(t\cdot)}(s)$ as $b^{(t)}$ or $b'$. For the continuous state, the posterior belief can be defined recursively as follows.

$$b^{(t)}(s^t) = \eta O(s^t, a^{t-1}, z^t) b^{(\cdot t)}(s^t) \tag{5}$$

According to the above definitions of the prior belief and the posterior belief, the updating processes of the belief state *b* can be described as $b^{(t\cdot)} \xrightarrow{T} b^{(\cdot t+1)} \xrightarrow{O} b^{(t+1\cdot)}$, where *T* is the state transition function, *O* is the observation function. Actually, the two computational processes can be merged into one computational process. Using Bayes filter update, the belief at time *t*+1 represented by $b'$can be updated by the belief stat *b*, action *a* and observation *z* at *t*, defined as follows.

$$b'(s') = \tau(b, a, z) = \eta O(s', a, z) \int_{s \in S} T(s, a, s') b(s) d\ s \tag{6}$$

where, $\tau(b, a, z)$ is the updating function, $\eta$ is the normalization factor, $b'$ is the posterior belief of *b*, $s'$ is the state at time *t*+1.

In addition, Bellman shows that the state value function $V$ is defined as follows.

$$V^{t+1}(b) = \max_{a\in A}\left[R(b,a)+\gamma\int_{z\in Z}P(z\mid b,a)V^{t}(b)\right] \tag{7}$$

where, $P(z\mid b,a)=\int_{s\in S}P(z\mid s,a)b(s)ds$.

The policy $\pi$ of POMDPs is to map belief to action, denoted $\pi(b)\rightarrow a$. The goal of the agent is to find the optimal policy $\pi^{*}$ that maximizes the expected sum of discounted rewards (expected return) starting from the initial state.

$$\pi^{*}=\arg\max_{a}\left[\sum_{t=0}^{\infty}\gamma^{t}\mathrm{E}\left[R(b^{t},a^{t})\right]\right] \tag{8}$$

where, $\gamma$ is a discount factor.

## III. SYSTEM MODEL

In the paper, we cast the target tracking scheduling as POMDPs. Then base station in WSNs can be seemed as an agent. The best scheduling policy of target tracking is the optimal policy of POMDPs.

### a. States and transitions Model

The states set $S$ of WSNs is composed of the states of the moving target $G$ and the states of nodes $F$. For each state $s$ in $S$, $s=G\times F$, $G$ and $F$ are independent of each other. $G=[x,y,v_x,v_y]^T$, where, $x$ and $y$ denote the position of the moving target in a two-dimensional coordinate system, $v_x$ and $v_y$ represent the speed of the moving target in the two-dimensional coordinate system. $F=[F_1,F_2,...,F_N]^T$, $F_n\in\{0,1\}$ $(1\le n\le N)$ denotes the states of nodes in the wake or sleep states respectively. The action set $A$ represents all possible scheduling policies chosen by sensor nodes. If we select $k$ nodes during the target tracking scheduling, the size of $A$ is $|A|=C_N^k$. The state of node at time $t$+1 is dominated by the scheduling policy $a^t$ at $t$, the state transition function of node is defined as follows.

$$F_n^{t+1} = \begin{cases} 0, & n \text{ is not selected by } a^t \\ 1. & n \text{ is selected by } a^t \end{cases} \quad (1 \le n \le N) \tag{9}$$

The state of the moving target at time *t*+1 is defined as follows.

$$G^{t+1} = \begin{bmatrix} x^{t+1} \\ y^{t+1} \\ v_x^{t+1} \\ v_y^{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^t \\ y^t \\ v_x^t \\ v_y^t \end{bmatrix} + \begin{bmatrix} T_s^2/2 & 0 \\ 0 & T_s^2/2 \\ T_s & 0 \\ 0 & T_s \end{bmatrix} \begin{bmatrix} \beta_x^t \\ \beta_y^t \end{bmatrix} \tag{10}$$

where, $T_s$ is the length of time slice. $\beta_x^t$ and $\beta_y^t$ are the noise function directed *x*-coordinate and *y*-coordinate respectively, and presented by Gaussian distribution $N(0,\sigma_x^2)$ and $N(0,\sigma_y^2)$ respectively.

The state transition function $T(s^t, a^t, s^{t+1})$ represents the change of the moving target states and the nodes states between two adjacent time slices. Because the moving target states and the sensor node states are independent. Thus we can factor the state transition function into two small functions.

$$T(s^t, a^t, s^{t+1}) = [P(G^{t+1} \mid G^t, \beta^t), P(F^{t+1} \mid a^t)]^T \tag{11}$$

where, $P(G^{t+1} \mid G^t, \beta^t)$ is the transition function of the moving target, $\beta^t$ is noise at *t*, $P(F^{t+1} \mid a^t)$ is the transition function of sensor node states.

b. Observation model

The observation set *Z* denotes all possible observation of the base station to the moving target. At each time horizon, when the moving target can be detected, the current awake sensor nodes can obtain the distance, angle and speed of the moving target. The sensor nodes will send the obtained information to the base station. Through information fusion, we can acquire an observation *z* for the moving target. The observation function $O(s^{t+1}, a^t, z^t)$ is used to reflect the uncertainty. When the data transmission error is low, the perception information about environment is accurate and the noise is small, the probability of observation $z^t$ accorded with $s^{t+1}$ is high, the opposite is low. The observation function upon executing action $a^t$ in state $s^{t+1}$ is defined as follows.

$$O(s^{t+1},a^t,z^t)=P(z^t\,|\,s^{t+1},a^t)=\frac{P(s^{t+1},a^t,z^t)}{P(s^{t+1},a^t)} \tag{12}$$

c. Reward model

The reward function is used to adjust the balance between target tracking performance and energy consumption. We should try to not only choose the nodes with low noise and high accuracy information, but also avoid the nodes with premature death. The reward function upon executing action $a^t$ in state $s^t$ is defined as follows.

$$R(s^t,a^t)=\sum_{i=1}^{k}\frac{\eta}{d(G^t,E_i)}+\sum_{n=1}^{N}(R_n^{Tar}g(n,a^t)+R_n^{Sel}\,|F_n^t-g(n,a^t)\,|) \tag{13}$$

where, $g(n,a^t)$ is 1 when the sensor node *n* is selected by $a^t$, otherwise is 0, $d(G^t,E_i)$ is the Euclidean distance between the target position and the selected sensor node $E_i$ by $a^t$, $\eta$ is reward adjusted factor, $R_n^{Tar}$ is the reward of the sensor node *n* which can perceive the moving target, and $R_n^{Sel}$ is the reward that the node *n* is not chosen repeatedly.

The purpose of $R_n^{Tar}$ is to improve the tracking performance. It is associated with the noise and sensory information error of sensor nodes. The data of sensor nodes is more accurate, the value of $R_n^{Tar}$ is higher. $R_n^{Sel}$ is identical for all sensor nodes. If the node is chosen repeatedly, the reward $R_n^{Sel}$ is 0. The objective of $R_n^{Sel}$ is to reduce the node energy consumption accompanied by repeated selection.

## IV. POSTERIOR BELIEF CLUSTERING (PBC) ALGORITHM

Section III models the target tracking problem as POMDPs problem. Form Equation (13), we can conclude that the optimal tradeoff between tracking performance and energy consumption is the optimal value function of POMDPs. The belief states space is the posterior distribution of the tracking system states conditioned on the observable history at each time [9]. However, planning of POMDPs over the belief states space is the curse of dimensionality which makes it impossible to obtain an online solution of target tracking scheduling [13].

In order to solve this problem, we present a novel posterior belief clustering (PBC) algorithm. The main idea of PBC is as follows. We construct a belief tree rooted by the current belief, and calculate its prior belief. Then we use Bayesian filter updating method to calculate the optimal posterior belief, and minimize the error between the posterior belief and the clustering point according to the KL divergence to obtain the optimal observation. In the range of error, we conduct clustering operation. Further, we compute the reward of the clustered posterior belief. For the same clustering posterior belief, we calculate the posteriori expected reward just only once, and assign the reward to all same clustering posterior belief sensor nodes. At last, PBC compares the reward of the clustered posterior belief with the upper and lower bounds of the value function, when the result is less than, the branch-and-bound pruning approach is exploited to prune the belief sub-trees to reduce the scale of belief states space. When the termination condition is satisfied, PBC will get the local optimal policy.

The Kullback–Leibler ( KL ) Divergence is a non-symmetric measure of the difference between two probability distributions [12]. If $\varphi$ and $\psi$ are the same distributions over the same space $\Omega$, then the KL divergence ( relative entropy ) of $\varphi$ to $\psi$ is defined as follows.

$$D(\varphi \| \psi) \underline{\underline{\Delta}} E_{\varphi}[\ln \frac{\varphi}{\psi}] = \sum_{\omega_i \in \Omega} \varphi[\omega_i] \ln \frac{\varphi[\omega_i]}{\psi[\omega_i]} \tag{14}$$

Supposed there are $k$ observations, their optimal posterior belief $b^*(s_k)$ updated by the Particle filters (Particle filters are a sample-based variant of Bayes filters) update is defined as follows.

$$b^*(s_k) \propto \int \cdots \int \prod_{i=1}^{k} P(z_i^t \mid s_i^t) P(s_i^t \mid s_i^{t-1}, a_i^{t-1}) \cdot b(s_0^t) ds_0^t \cdots ds_{k-1}^t \tag{15}$$

where, $b(s_0^t)$ is the last posterior belief. Let $b_i(s_k^t)$ to be the posterior belief of the $i$th observation in the set of $k$ observations, then $b_i(s_k^t)$ can be calculated by:

$$b_i(s_k^t) \propto \int \cdots \int P(z_i^t \mid s_i^t) \cdot \prod_{j=1}^{k} P(s_j^t \mid s_{j-1}^t, a_{t-1}^t) b(s_0^t) ds_o^t \cdots ds_{k-1}^t \tag{16}$$

Let $b_{\text{mix}}(s_k^t \mid \alpha)$ to be the weighted sum of $b_i(s_k^t)$, $\alpha$ is the mixture weight, then the relationship between $b_{\text{mix}}(s_k^t \mid \alpha)$ and $b_i(s_k^t)$ is:

$$b_{\text{mix}}(s_k^t \mid \alpha) \propto \sum_{i=1}^{k} \alpha_i b_i(s_k^t) \tag{17}$$

where, $\alpha_i \geq 0, \sum_i \alpha_i = 1$.

The weight $\alpha_i$ is importance for observation $z_i^t$ to the optimal posterior beliefs. The optimal observation can be evaluated by minimizing the KL divergence between $b_{\text{mix}}$ and $b^*$. The optimal $\alpha^*$ can be calculated as follows.

$$\begin{aligned}\alpha^* &= \arg\min_{\alpha\in\Gamma} \text{KL}(b_{\text{mix}}(\cdot \mid \alpha) \parallel b^*) \\ &= \arg\min_{\alpha\in\Gamma} \int b_{\text{mix}}(s_k^t \mid \alpha) \cdot \ln \frac{b_{\text{mix}}(s_k^t \mid \alpha)}{b^*(s_k^t)} ds_k^t\end{aligned} \tag{18}$$

where, $\Gamma = \{\alpha \mid \sum_{i=1}^k \alpha_i = 1, \alpha_i \geq 0\}$.

The above description is based on the assumption that $b^*$ is as known. In practice, this is not true. We can exploit Monte Carlo approach to estimate $b^*$ [14]. $\alpha^*$ can be optimized by using expectation-maximization (EM) or (constrained) gradient descent. In our approach, we perform a small number of gradient descent steps to find the mixture weights as [15].

From the above computation, we can get a posterior beliefs clustering of $k$ observations. The belief can be acquired through accumulating $\alpha_i$:

$$b(C_b^i) = \sum_{m\in C_b^i} \alpha_i^{(m)}) \tag{19}$$

Let $C_b$ is the set of all clustering, $C_b = \sum_{i=1}^H C_b^i$, $H$ is the number of clustering. The belief rewards before clustering can be calculated as follows.

$$\begin{aligned}R(b^c) = R(b^c, a_1) + \gamma \int_{z_1} P(z_1 \mid b^c, a_1) R(b_{a_1,z_1}, a_2) + \cdots + \\ \gamma^{D-1} \int_{\{z_i\}} \left[ \prod_{i=1}^m P(z_i \mid b_{\{a_j\},\{z_i\}}, a_{n-1}) \right] R(b_{\{a_j\},\{z_i\}}, a_n)\end{aligned} \tag{20}$$

After clustering, the same clustering have same rewards, the belief rewards after clustering can be calculated as follows.

$$R(b^c) = R(b^c, a_1) + \sum_{i=2}^H \gamma^{i-1} R(b(C_b^i)) \tag{21}$$

We propose a novel posterior belief clustering (PBC) algorithm, which is detailed in Algorithm 1, to overcome the curse problem of POMDPs. A Particle filters method is exploited to update the optimal posterior beliefs, and the *K*-means approach is used to cluster the posterior beliefs [16].

Table 1: Posterior belief clustering (PBC).

```
Algorithm 1. Posterior belief clustering (PBC)
1: initialization: $T$ denotes the posterior belief tree, $d$ denotes the depth of the tree, $D$ is the largest depth of this tree, $R_{\max}(b)$ denotes the optimal reward, $R^{c}$ denotes the current reward, $b^{c}$ denotes the current belief, $k$ denotes the number of clusters.
2: construct the posterior tree, the Particle filters method is exploited to update the optimal posterior beliefs;
3: cluster the posterior beliefs, $C_b = \text{Clustering}(T)$, for $C_b^j \in C_b$, $i \in [1, 2, \cdots, k]$, calculate $b_i(C_b^j) = \sum_{m \in C_b^j} \alpha_i^{(m)}$ and $b^*(C_b^j) = \sum_{m \in C_b^j} \prod_{i=1}^{k} \alpha_i^{(m)}$;
4: perform a small number of gradient descent steps to find the mixture weights: $\frac{\partial J}{\partial \alpha_i} = 1 + \sum_{C_b^j} b_i(C_b^j) \ln \frac{\sum_{m=1}^{k} \alpha_m b_m(C_b^j)}{b^*(C_b^j)}$;
5: calculate the clustered posterior belief rewards: $R(b^c) = R(b^c, a_1) + \sum_{i=2}^{H} \gamma^{i-1} R(b(C_b^i))$;
6: the branch-and-bound pruning approach is exploited to prune the belief sub-trees;
7: if $(d = D) \cup \left\| V^*(b) - R_{\max}(b) \right\| < \varepsilon$, PBC gets the local optimal policy.
```

From Algorithm 1, we can know that the size of the belief states space after clustering is $(|A||Z|\ k)^D$. According to the Bellman theorem [17, 18], the same posterior belief has the same reward. Therefore, for the same clustering posterior belief, we calculate the posteriori expected reward just only once. Our algorithm can avoid repeated computation to improve the real-time performance. At last, a branch-and-bound pruning approach is exploited to prune the belief tree to avoid unnecessary computation. As discussed above, the time complexity of PBC is $O(|A||Z|\ k)^D$, the space complexity is $O(|A||Z|)^D$.

## V. RESULTS AND DISCUSSION

We evaluate the effectiveness of PBC algorithm from two points of view. The first experiment is the comparison of the tracking performance and energy consumption with Q-MDP [11]. The second experiment is the comparison of the target tracking accuracy and the lifetime with LEACH [19, 20-22]. In this paper, the simulation platform is Matlab R2010a, 32-bit Windows 7, Intel (R) Core (TM) : i3 CPU 3.07 GHZ, RAM: 4 GB.

### a. Comparison of the tracking performance and energy consumption

In our simulation environment, the target tracking happens in a 100×100 $m^2$ area of the wireless sensor network, and the speed of the moving target is 10 m/s. The number of sensor nodes is 100. The communication between the sensor nodes and the base station is single-hop. The other parameters used in our experiments are as follows: the sampling interval is $T_s$ =1 second; the noise is $\sigma_x = \sigma_y = g$ ( $g$ is the gravitational acceleration ); the discount factor $\gamma = 0.95$ ; the reward factor is $\eta = 28$, the largest depth of belief tree is $D = 6$ . The observation noise is Gaussian distribution with $N(0, 4.25)$ . We compare our algorithm (PBC) with Q-MDP shown in [11].

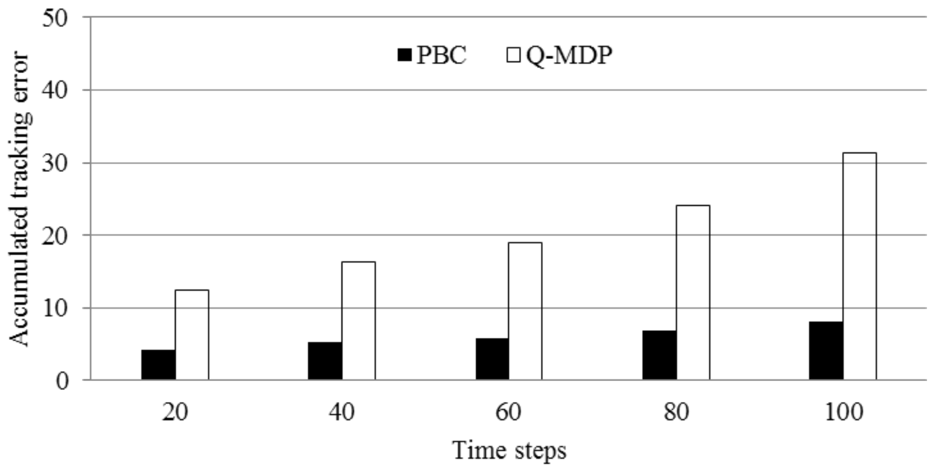

Figure 1. The comparison of the accumulated errors for PBC and Q-MDP

In Figure 1, the tracking accumulated error of PBC is small, which shows that PBC has better accuracy. When *t* = 20, the tracking accumulated error between PBC and Q-MDP is small. But as the growth of the time, the error accumulation of Q-MDP algorithm is bigger than that of PBC.

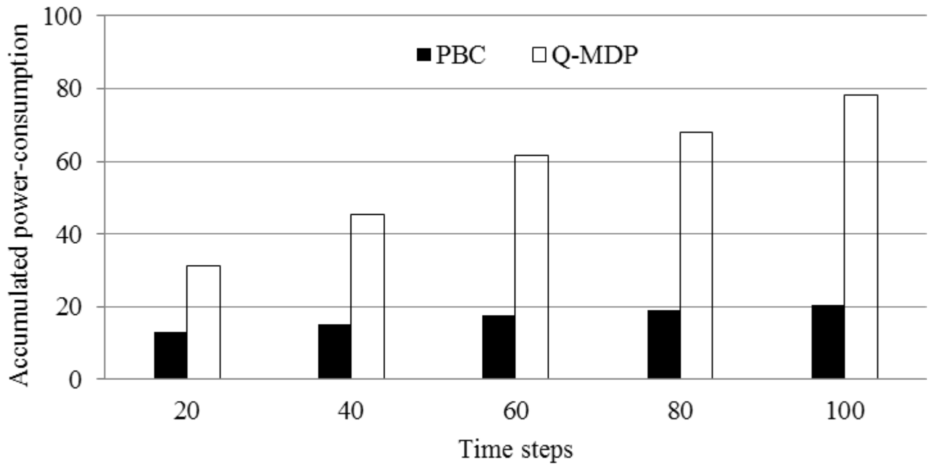

Figure 2. The comparison of the accumulated energy consumption for PBC and Q-MDP

From Figure 2, the energy consumption sum of PBC algorithm is smaller than that of Q-MDP algorithm. Because the clustering method is exploited in PBC, the energy consumption should not grow rapidly with the time growth. Figure 2 also reflects PBC algorithm has good stability.

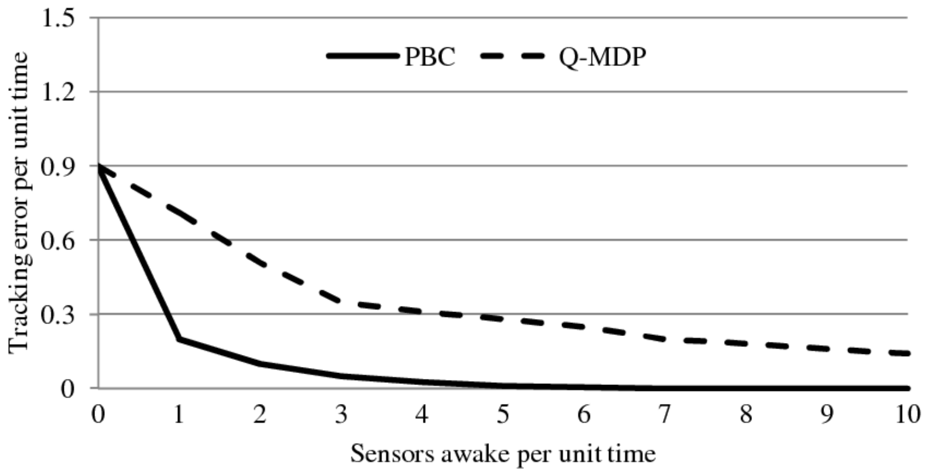

Figure 3. The tradeoff between tracking performance and energy consumption

Figure 3 describes the optimal tradeoff between tracking performance and energy consumption compared with PBC and Q-MDP in per unit time. The number of wakeup sensors is more, the tracking performance is better and the error is smaller. However, the energy consumption will be larger. From Figure 3, we can conclude that our algorithm balances the relationship between the tracking performance and the energy consumption better than Q-MDP. PBC is able to realize smaller energy consumption to gain better tracking performance.

b. Comparison of the target tracking accuracy and the lifetime

In this section, the simulation environment is a 100×100 $m^2$ area of the wireless sensor network. The speed of the moving target is 4 m/s, and the number of sensor nodes is 50. The largest depth of belief tree is $D$ = 4. The observation noise is Gaussian distribution with $N$ (0, 1.45). We compare our algorithm (PBC) with LEACH shown in [19, 20].

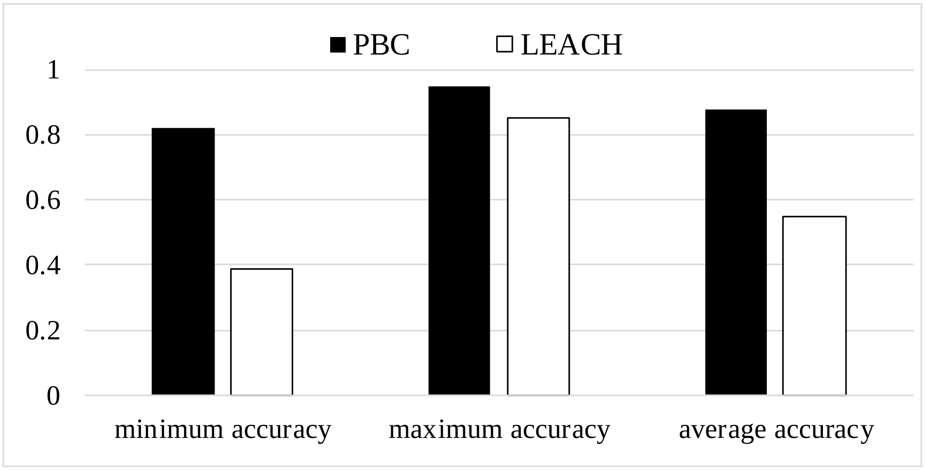

Figure 4. Comparison of target tracking accuracy

Figure 4 shows the target tracking accuracy between PBC and LEACH. LEACH does not consider the target moving factor when forming a cluster, its average accuracy is small, and the accuracy is rather changeable. The average accuracy of PBC is higher than LEACH, and the changeable range of accuracy is also smaller. The experimental results show that PBC not only can get higher accuracy, and better stability.

The comparison of the lifetime between PBC and LEACH is shown in Figure 5. The number of survival nodes changes with runtime (rounds). With the increasing of the death nodes, the death speed of nodes is in trend of accelerating. LEACH appears dead node firstly, and all of nodes have been dead in about 1200 rounds. PBC begins to appear dead node in about 600 rounds, and all of sensor nodes die in about 1800 rounds. Comparison results show that PBC can balance the node load and effectively prolong the network lifetime.

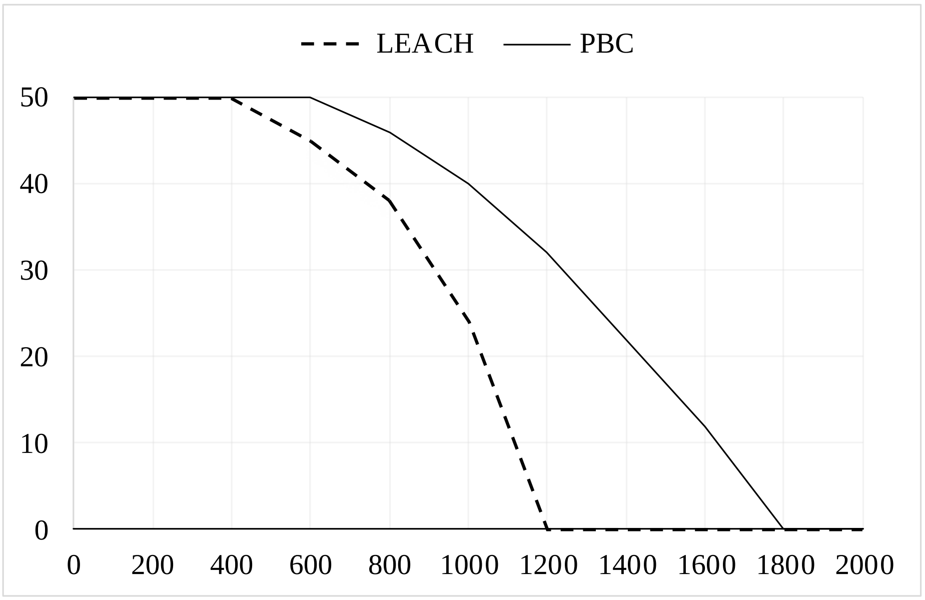

Figure 5. Comparison of network lifetime

## VI. CONCLUSIONS

In this paper, we cast the optimal energy-tracking tradeoff problem in WSNs as the optimal value function problem in POMDPs. A novel posterior belief clustering algorithm is proposed to solve the tradeoff between target tracking performance and sensors energy consumption in wireless sensor networks. Our numerical results illustrate the effectiveness of the approach.

## ACKNOWLEDGMENTS

This work was supported by the NNSF of China under grant 61074058 and 60874042, and also was supported by the Shenzhen Technology Innovation Program of China under grant JCYJ20120617134831736.

## REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey", Computer networks, vol. 38, No. 4, 2002, pp. 393-422.

[2] M. Martalo, C. Buratti, G. Ferrari and R. Verdone, "Clustered IEEE 802.15. 4 sensor networks with data aggregation: energy consumption and probability of error", IEEE Wireless Commutations Letter, vol. 2, No. 1, 2013, pp. 70-73.

[3] T. Cao, Y. H. Wang, X. M. Xiong and Y. Hao, "Cluster-based Routing Performance optimization Constraint of Energy, Delay and Connectivity Metrics in Wireless Sensor network", International Journal on Smart Sensing and Intelligent Systems, vol. 6, No. 5, 2013, pp. 2103-2118.

[4] Y. Liu, Y. H. Wang, S. Y. Chen, X. Li and Z. F. Rao, "A Hybrid MAC Mechanism for Multiple Load Intelligent Vehicle Transportation Network," International Journal on Smart Sensing and Intelligent System, vol. 4, No. 4, 2011, pp. 662-674.

[5] C. L. Cheng, H. Wu, Z. H. Yu and D. Y. Zhang, "Outlier Detection Based on Similar Flocking Model in Wireless Sensor Networks," International Journal on Smart Sensing and Intelligent Systems, vol. 6, No. 1, 2013, pp. 19-37.

[6] H. M. Ammari and S. K. Das, "A Study of k-Coverage and Measures of Connectivity in 3D Wireless Sensor Networks," IEEE Transactions on Computers, vol. 59, No. 2, 2010, pp. 243-257.

[7] J. Kim, X. Lin, B. N. Shroff and P. Sinha, "Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks With Anycast," IEEE/ACM Transactions on Networking, vol. 18, No. 2, 2010, pp. 515-528.

[8] T. Nanayakkara, M. N. Halgamuge, P. Sridhar and A. M. Madni, "Intelligent sensing in dynamic environments using Markov decision process", Sensors, vol. 11, No. 1, 2011, pp. 1229-1242.

[9] Y. He, and K. Chong, "Sensor scheduling for target tracking in sensor networks", Proc. *CDC 2004*, pp. 743-748, Nassau, Bahamas, Dec. 17-17, 2004.

[10] X. Fei, A. Boukerche and R. Yu, "A POMDP based K-coverage dynamic scheduling protocol for wireless sensor networks", Proc. *GLOBECOM 2010*, pp. 1-5, Miami, FL, UAS, Dec. 6-10, 2010.

[11] J. A. Fuemmeler, G. K. Atia and V. V. Veeravalli, "Sleep control for tracking in sensor networks", IEEE Transactions on Signal Processing, vol. 59, No. 9, 2011, pp. 4354-4366.

[12] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes", Proc. *UAI 1998*, pp. 33-42, Madison, Wisconsin, UAS, July 24-26, 1998.

[13] B. Wu, H. Y. Zheng and Y. P. Feng, "Point-based online value iteration algorithm in large POMDP", Applied Intelligence, vol. 40, No. 3, 2014, pp. 546-555.

[14] X. Wang, J. J. Ma, S. Wang and D. W. Bi, "Cluster-based dynamic energy management for collaborative target tracking in wireless sensor networks". Sensors, vol. 7, No. 7, 2007, pp. 1193-1215.

[15] C. Kwok, D. Fox and M. Meila, "Real-time particle filters", Proceedings of the IEEE, vol. 92, 2004, pp. 469-484.

[16] R. Cohn, E. Durfee and S. Singh, "Planning Delayed-Response Queries and Transient Policies under Reward Uncertainty", Proc. *MSDM 2012*, pp. 17-23, Holetown, Barbados, April 14-18, 2012.

[17] S. Ross, J. Pineau, S. Paquet and B. Chaib-draa, “Online planning algorithms for POMDPs,” Journal of Artificial Intelligence Research, vol. 32, No.1, 2008, pp. 663-704.

[18] J. Pineau, G. Gordon and S. Thrun, “Anytime point-based approximations for large POMDPs,” Journal of Artificial Intelligence Research, vol. 27, No. 1, 2006, pp. 335-380.

[19] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” Proc. *HICSS 2000*, pp. 3005-3014, USA, Jan. 4-7, 2000.

[20] G. R. Mendez and S .C. Mukhopadhyay, “A Wi-Fi Based Smart Wireless Sensor Network for an Agricultural Environment”, Smart Sensors, Measurement and Instrumentation, Vol. 3,

Wireless Sensor Networks and Ecological Monitoring, ISBN 978-3-642-36364-1, Springer-Verlag, by S. C. Mukhopadhyay, and J. A. Jiang, 2013, pp. 247-268.

[21] X. N. Fan and Y. L. Song, "Improvement on LEACH Protocol of Wireless Sensor Network," Proc. *SENSORCOMM 2007*, pp. 260-264, Valencia, Spain, Oct. 14-20, 2007.

[22] N.K. Suryadevara, S.C. Mukhopadhyay, R. Wang, R.K. Rayudu, Forecasting the behavior of an elderly using wireless sensors data in a smart home, Engineering Applications of Artificial Intelligence, Volume 26, Issue 10, November 2013, Pages 2641-2652, ISSN 0952-1976, http://dx.doi.org/10.1016/j.engappai.2013.08.004.