An Energy-Efficient Tracking Algorithm based on Gene Expression Programming in Wireless Sensor Networks

Shucheng Dai 1, Changjie Tang 1, Shaojie Qiao 2, Yue Wang 1, Hongjun Li 1, 3, Chuan Li 1

1School of Computer Science, Sichuan University, Chengdu, China
2School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China
3School of Computer Science, Southwest University of Science and Technology, Mianyang, China

{daihucheng, cjtang, lcharles}@scu.edu.cn
{qiaoshaojie, hkharryking, achilee.lihj}gmail.com

Abstract—Wireless Sensor Networks (WSNs) are widely used in detecting, locating and tracking moving objects. The cheap, low-powered and energy-limited sensors that are set up in large areas may consume large portion of energy and disable the whole network. In this paper, a new energy-efficient method based on Distributed Incremental Gene Expression Programming is proposed to discover the moving patterns of moving objects in order to turn on/off some sensor nodes at certain time to save energy. The main contributions include: a) Distributed GEP methods are used to perform collaborative mining the patterns of moving objects, b) adjustable sliding window are adopted to balance the trade-off of the high accuracy and low energy consumption, c) simulation results show that the proposed GEP-based motion prediction algorithm can greatly improve the tracking efficiency, increase the lifetime of the network by around 25% compared to other tracking algorithms, i.e., EKF and ECPA.

Keywords—Gene expression programming; Location prediction; Energy efficiency; Wireless sensor network; Target tracking

I. INTRODUCTION

Target tracking is an important application in terms of wireless sensor networks (WSNs) [1]. Bayesian Network and kalman filtering are two classical methods for achieving this task. One possible solution is as follows. The system state includes the position, direction and velocity of the target. At each step, sensors near to the target form clusters and select a leader to perform the kalman filtering, and the updated state is forwarded to cluster leaders chosen from the next step. The kalman filtering implementation is straightforward in a centralized environment. But it is difficulty in the extremely distributed settings such as WSNs due to the energy-efficiency and lower computation capability of sensor nodes.

This study proposes a new scheme based on Gene Expression Programming (GEP) [2]. GEP works well in modeling the moving patterns of targets in WSNs as well as targets behaviors without aprior knowledge. Based on the historical location information of the target, GEP automatically evolves the trajectory of a moving object. To handle the problem, this paper makes the following contributions:

1) A new algorithm named distributed GEP is proposed to discover moving patterns of targets. The basic idea is that distributed GEP runs at multiple collaborative sensor nodes until they discover a good path or stopping criteria are satisfied.

2) A adjustable sliding window is adopted to ensure that distributed GEP can quickly train the latest collected location data. When new location data are received, old location data are discarded when the prediction error exceeds a certain threshold to ensure succeeding evolutions to find newly moving patterns.

3) Extensive simulations are conducted on OMNet++, a discrete event simulator, to show that new algorithms effectively prolong the network lifetime by 25% averagely when compared to other algorithms, i.e., EKF and ECPA.

II. RELATED WORK

There are many research efforts on target detection and tracking in terms of WSNs, which describes several aspects of collaborative signal processing [1, 3, 4] and real-time application for biologists to find the presence of individuals [5]. A set of approaches presented in [6-8] were proposed recently to solve the target localization and tracking problem with proximity binary sensors which transmit only one bit information to indicate whether a target is present. The information transmitted among sensor nodes was greatly reduced, while the localization error was increased. Shrivastava et al. [7] proved that the achievable resolution in localizing a target trajectory is with the order of \( \frac{1}{\rho R} \), where \( R \) is the sensing radius and \( \rho \) is the sensor density. Energy-efficient algorithms, protocols, node hardware and software designing technologies can prolong the lifetime of the network. Several approaches have been proposed at hardware and software levels to design energy efficient CPU, OS, algorithms and communication protocols [9]. Dynamic power management (DPM) schemes have been proposed in [10-12] to reduce the power consumption by selectively turning off idle components, such as radio frequency (RF) transmitter, RF receiver, sensing device, A/D converter, and the sensor node.

*Supported by the National Science Foundation of China under Grant No. 60773169 and the 11th Five Years Key Programs for Sci. &Tech. Development of China under grant No. 2006BAI05A01
To track moving targets energy-efficiently, Dylan et al.[13] proposed a solution based on CA (Cellular Automata) to reduce long distance communications among nodes, but a higher power consumption is introduced because it does not turn off those nodes that are far away from the moving object. Yang et al.[14] proposed ECPA (Enhanced Closest Point Approach) to predict the location of targets during the phase of moving, but the velocity and direction calculation algorithms with regard to the targets are overloaded for sensor nodes which often have low power and computation capability.

III. TARGET TRACKING MODEL

The tracking model is described in Fig. 1. The monitored region is covered with many sensor nodes that are distributed manually or randomly. A target moves along a trajectory that is unknown before and is detected by some sensor nodes that are depicted as black solid circle nodes. Suppose that the locations of the moving object are known at time $t_i, t_{i-1}, \ldots, t_0$. A question arises: where will it appear at time $t_{i+1}, t_{i+2}, \ldots$?

Once find the trajectory with moving object and express it as function, then the following tasks are straightforward:

- predict the locations that the moving object will appear at time $t_{i+1}, t_{i+2}, \ldots$
- Efficiently activate the necessary sensor nodes to save energy.

Target tracking applications needs careful consideration of the trade-off between tracking error and energy consumption. The tracking error is defined by the average target location estimation error of sensor nodes. The better trajectory leads to better target location estimation.

The trajectory of a moving object is treated as a sequence of time-stamped locations that are collected by sensor nodes around the target. It is described as follows.

**Definition 1:** (Trajectory) A trajectory of a moving object is a time sequence with time interval $\Delta t$:

$$P(t) = \{X(t), Y(t)\} = \{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$$

where for $\forall i \in [0, n]$, $t_i < t_{i+1}$, $t_i + \Delta t = t_{i+1}$, $(x_i, y_i)$ is 2D points, and $x(t_i) = x_i$. $y(t_i) = y_i$, $P(t) = [x_i, y_i]$.

$P(t)$ can be obtained by the trajectory mining algorithm. Sliding window prediction method (SWP) is adopted to load the latest historical data to train trajectories. The basic idea of SWP is given below.

Given the historical location data $P(0), P(1), \ldots, P(n)$ with length $n + 1$. The size of a sliding window is denoted as $h$ ($h \leq n+1$).

(a) Find a formula $\hat{P}(t) = [f(t), g(t)]$ from $h$ samples and predict the location at time instant $m$, ($m=n-1$) by Eq.(2). The Example 1 illustrates the phases of evolutions of trajectories.

$\hat{P}(m) = [\hat{X}(m), \hat{Y}(m)] = [f(t_m), g(t_m)]$ (2)

(b) During the evolving process, $h$ determines how many historical location data are used. The smaller $h$ leads to less energy consumption and faster convergence speed. $h$ is adjusted based on the location prediction error $\varepsilon$, geometric distance between the prediction value and the real measurement. $\varepsilon$ should be as small as possible and is calculated by Eq.(3).

$$\varepsilon = \sqrt{(\hat{x}(m) - x(m))^2 + (\hat{y}(m) - y(m))^2}$$

The trade-off between the energy consumption and the prediction accuracy is balanced by adjusting $\varepsilon$ to satisfy different application requirements.

**Example 1.** The historical locations are listed in Table 1.

<table>
<thead>
<tr>
<th>$t_i$</th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>19</td>
</tr>
</tbody>
</table>

Thus, $f$ may be $P(t) = [t+1, r^2+t+0.5]$, $P(t) = [t+2, r^2+2t+1]$ or $P(t) = [t+1, r^2+3t+1]$. All are suitable in environments with different prediction accuracy requirements.
V. DISTRIBUTED GEP SCHEME

A. Fitness evaluation of individual

Combining the fitness evaluation and prediction error, distributed GEP calculates the fitness of each individual in distinct populations by Eq.(4).

\[ E_i = \frac{1}{h} \sum_{j=1}^{h} (e_i^2) \quad (4) \]

Where \( e_i \) is the evaluation error of the \( i \)-th location data and \( E_i \) is the fitness value of the \( i \)-th individual.

B. Trajectory mining algorithm

The main idea of trajectory search algorithms is given below.

1) Sensor nodes that are activated based on the node scheduling algorithm.

2) Communications occur among sensor nodes when one node succeeds in obtaining a trajectory and notifies other nodes.

3) Other nodes stop running their algorithms and obtain the trajectory to predict future location of the moving object.

Fig. 2 details the flowchart of distributed GEP and outputs \( \hat{P}(t) \).

One sensor node stops distributed GEP if one of these stopping criteria is satisfied:

1. The maximum number of generations is reached
2. Distributed GEP exceeds the specified runtime.
3. One or more other nodes send stopping signal to the node.
4. The node succeeds in obtaining a trajectory.

C. Location prediction and node scheduling

Once a trajectory is found, the model uses it to predict the location where the target will appear at time \( t_{i+j} \) by Eq.(2),

\[ f(t_{i+j}) - r_0 \leq x_i \leq f(t_{i+j}) + r_0, 0 < j \leq L \quad (5) \]

\[ g(t_{i+j}) - r_0 \leq y_i \leq g(t_{i+j}) + r_0, 0 < j \leq L \quad (6) \]

The node scheduling algorithm is given in Algorithm 1.

Algorithm 1. Node Scheduling

Input: \( \hat{P}(t) \) and prediction length \( L \).

Output: activated sensor nodes

1) for(k=0;k<\( L \); k++) {
2) for each(sensor node \( (x_j, y_j) \) in WSN) {
3) if \( (f(t_{i+k+1})-R \leq x_j \) and \( x_j \leq f(t_{i+k+1})+R \)
4) and \( g(t_{i+k+1})-R \leq y_j \) and \( y_j \leq g(t_{i+k+1})+R \))
5) \( S_i \) is scheduled awake at \( t_{i+k+1} \)
6) else
7) \( S_i \) is scheduled asleep at \( t_{i+k+1} \)\}

VI. EXPERIMENTS

A. Sensor Networks Setting

Suppose that hundreds of sensor nodes are uniformly distributed in a square of 100×100 meters. A target can present at a random place in the WSNs and sends signal with the strength of \( w_t \). The prediction accuracy \( \tau \) is 2 meters and \( h=7 \).

The sensing range \( r_0 \) of sensor nodes is set to 7 and \( r_1 \) is set to 10. The sensing energy \( e_s \) is 100uJ, and transmission (receiving and sending) energy for one packet is \( e_t=e_r=100uJ \).

The initial energy of each sensor node is 100mJ. Parameters used in distributed GEP are listed in Table 1.

<table>
<thead>
<tr>
<th>Parameters setting in distributed GEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Set</td>
</tr>
<tr>
<td>Basic Terminal Set</td>
</tr>
<tr>
<td>Number of Generations</td>
</tr>
<tr>
<td>Population Size</td>
</tr>
<tr>
<td>Head Length</td>
</tr>
<tr>
<td>Mutation Rate</td>
</tr>
<tr>
<td>Inversion Rate</td>
</tr>
<tr>
<td>IS Rate</td>
</tr>
<tr>
<td>RIS Rate</td>
</tr>
<tr>
<td>Length of Insertion Sequence</td>
</tr>
<tr>
<td>One-point Recombination Rate</td>
</tr>
<tr>
<td>Two-point Recombination Rate</td>
</tr>
<tr>
<td>Gene Recombination Rate</td>
</tr>
</tbody>
</table>

B. Network lifetime

This experiment analyses the impact of the number of sensor nodes on the network lifetime. The results are given in
Fig. 3. It shows that the three algorithms often obtain longer network lifetime when the number of sensor nodes gets bigger. The network lifetime of distributed GEP is averagely 35% longer than EKF and ECPA.

C. Energy consumption

In this experiment, four hundred sensor nodes are used to monitor the grid network. The sensor nodes are manually distributed at cross points in the grid network.

The results show that the total left energy decreases when the time passes. The total left energy cannot be zero because that all these algorithms are invalid when the network fails. Distributed GEP performs better than EKF and ECPA. This is because it consumes less energy than EKF and ECPA after running the same time.

VII. CONCLUSIONS

In order to track targets energy-efficiently in WSNs, we presented a distributed computing GEP used in wireless sensor networks; proposed sliding window policy for distributed GEP to improve evolution process; proposed a new target tracking model; give extensive experimental results to show the good performance of distributed GEP.

REFERENCES


