Intelligent Control and Automation, **2011, 2, 196-202**
doi:10.4236/ica.2011.23024 Published Online August 2011 (http://www.SciRP.org/journal/ica)

Autonomous Target Interception Using Hybrid Sensor Networks*

Wenzhe Zhang[1], Yihuai Wang[1], Ying Li[1], Minglu Li[2]
[1]*School of Computer Science and Technology, Soochow University, Soochow, China*
[2]*Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai, China*
E-mail: wzzhang@suda.edu.cn

Received December 2, 2010; *revised May* 20, 2011; *accepted May* 27,2011

Abstract

Hybrid sensor networks (HSNs) comprise of mobile and static sensor nodes setup for purpose of collaboratively performing tasks like sensing a phenomenon or monitoring a region. In this paper, we present target interception as a novel application using mobile sensor nodes as executor. Static sensor nodes sense, compute and communicate with each other for navigation. Mobile nodes are guided to intercept target by the static nodes nearby. Our approach does not require any prior maps of the environment thus, cutting down the cost of the overall energy consumption. As to multi-targets multi-mobile nodes case, we present a PMB algorithm for task assignment. Simulation results have verified the feasibility and effectiveness of our approach proposed.

Keywords: Target, Hybrid Sensor Networks, Interception, CoS, Intercepting Strategy, PMB Task Assignment

1. Introduction

A networked system of hybrid sensor networks opens new frontiers in variety of civilian and military applications and in some scientific disciplines. A mixture of networked mobile robots and static sensors reduce the cost but preserves the flexibility and advantageous capabilities of a multi-robot system.

We are broadly interested in the mutually beneficial collaboration between mobile and static sensor networks. The underlying principle in this hybrid network between the mobile and static nodes is that: the static nodes serve as the sensing, computation and communication medium, whereas the robots provide action and finish the task of blocking with the help of static sensors. In this work we describe results from such a system which accurately and reliably solves the problem of target interception. Some properties of our approach are summarized below:

1) The sensor network is pre-deployed into the environment deterministically or randomly for the full coverage of the region.

2) After deployment, static sensor nodes can sense the condition of its environment. By the multi-hop communication, static nodes compute the distributions of local environment and evaluate appearance of the target.

3) The nodes of the sensor network are synchronized in time (high precision is not required).

4) The mobile node is made up of a static sensor and a robot, so it can communicate with any static nodes nearby.

5) The environment is not required to be static.

Sensor networks are deployed in the field of interest. They are expected to monitor the field and intercept any intruding target as soon as possible in an unmanned manner[1]. The concept of artificial potential fields for the purpose of obstacle avoidance was presented in [2-6]. The concept of Vector Field Histograms (VFH) based on locally constructed polar histograms for robot navigation was presented in [7]. It may be noted that a parallel approach for the construction of a navigation field has been proposed in sensor network. It uses potential fields and the hop count to compute the magnitude of the directional vectors. In our paper, we only use the static node nearby to guide the mobile node, which cut down the complication of computation greatly.

In this paper we propose a landmark-based pursuit

*This paper is supported by the Natural Science Foundation of China, No. 61070169, National Basic Research Program of China, No. 2006 CB303000 and Jiangsu Natural Science Foundation No. 10KJB520017.

strategy to address the problem of target interception. *i.e.* When the target intrudes the sensory field, static sensor nodes that detect the events collectively elect one head that has sensed the largest intensity of signal. After data computing, it will generate a sensing report and flood over the network by multi-hops. We call the sensing report data stimulus and the head node as center of stimulus. Mobile sensor node acquires the position of target and decides its optimal pursuit direction with the help of sensors nearby. When the target moves, center of stimulus is re-elected and static sensor node keeps the proper position of the target. Only if the update of the network is enough, can the mobile node select its optimal strategy to intercept the target. Instead of a mobile node wandering randomly, in sign-based approach each static node decides the direction to guide the mobile node. As a result, our approach with refreshing stimulus packets adapts to the mobile target, and can guide the mobile node to block the target as soon as possible.

We rely on the communication network to establish the navigation paths. Also, in our approach the mobile robots only need a local sense of target in order to move toward the correct direction. The obvious merit is that the mobile node does not need a pre-decided environment map, or a compass.

The rest of the paper is organized as follows: We present sensing model, stimulus election and target localization in Section 2 Mobile node navigation is presented in Section 3. In Section 4 we extend the case to multiple targets and mobile nodes and design the task assignment algorithms. Section 5 presents the simulation and evaluates its performance. Section 6 summarizes the work and sketches out our future plan.

2. Sensor Network Surveillance

2.1. Sensing Model

Let N_s be the number of static senor nodes, deployed over the surveillance region $R \subset \Re^2$. Let $s_i \in R$ be the location of the i-th sensor node and let $S = \{s_i : 1 \le i \le N_s\}$. Let $G = (S, E)$ be a communication graph such that $(s_i, s_j) \in E$ if and only if node i can communicate with node j. Let $N_m \ll N_s$ be the number of mobile nodes (for simplicity, $N_m = 1$) and Let $R_s \in \Re$ be the sensing range of each static/mobile sensor node. If there is a target at $x \in R$, the sensor node can detect the presence of the target. Each sensor records the sensor's signal strength,

$$S_i = \begin{cases} \dfrac{\beta}{1+\gamma\left\|s_i - x\right\|^{\alpha}} + w_i & if \left\|s_i - x\right\| \le R_s \\ w_i & if \left\|s_i - x\right\| > R_s \end{cases} \quad (1)$$

where α, β and γ are constants specific to the sensor type and they are normalized such that w_i has the standard Gaussian distribution. This signal-strength based sensor model is general for sensors available in sensor networks, such as acoustic and magnetic sensors, and has been used frequently [8-10]. For each sensor i, if $S_i \ge \eta$, where η is a threshold set for appropriate values of detection and false-positive probabilities, the node is activated and join in the mesh as shown in **Figure 1**. It will participate in the stimulus election in the next section and may transmit its S_i to its neighboring nodes if necessary.

2.2. Stimulus Election and Its Update

The election of a source follows the mechanism bellow. We want only one node to generate the report since it would be a waste of resources if every node detecting the target sends a report. The target creates a field of sensing signal strength, the nearer the sensor is, the signal strength it collects is larger.

Each node broadcasts a message indicating its signal strength and Cartesian coordinate (with some random delay to avoid collision). A node rebroadcasts its signal strength whenever it hears a neighbor's message with a weaker signal, but stops broadcasting when it hears a stronger one. In this way, messages roll throughout the whole network of the signal strength field. Finally the node with the strongest signal is elected as the Center of Stimulus (CoS) and generates the sensing reports.

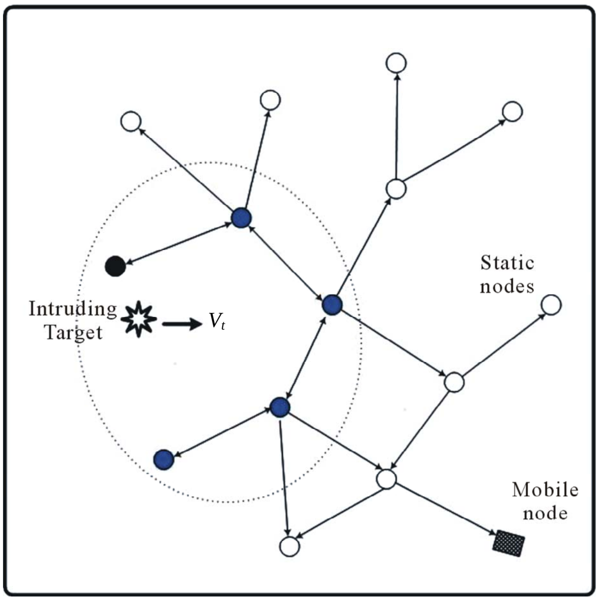

Figure 1. The stimulus starts from a source and ends at the mobile node. The black nodes forward the packet to the source collectively. Notice that some nodes outside of the mesh also receive the packet but do not forward it.

2.3. Target Localization and Velocity Estimation

Computation of target's information is performed by the node CoS. Let x_i be the Cartesian coordinate of sensor i, the position of a target is estimated as

$$\hat{x}_i = \frac{\sum_{j=1}^{k} S_{i_j} * x_{i_j}}{\sum_{j=1}^{k} S_{i_j}} \quad (2)$$

then $\hat{x}_i$ is broadcasted as stimulus data all through the network. In this way, although each sensor cannot give an accurate estimate of target's position, as more sensors collaborate, the accuracy of estimates improves [11].

We assume mobile sensor node can move at some certain velocity, here we note it as v_m. Target may intrude into the monitoring region and move with the speed of maximum v_t.

Based on the target location, we can calculate the velocity of the moving target as well. For simplicity, suppose the estimated positions of target at time t_1 and t_2 are $\hat{x}_1 = (x_1, y_1)$ and $\hat{x}_2 = (x_2, y_2)$ respectively, the velocity of the target is computed as:

$$\hat{v}_t = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{|t_1 - t_2|} \quad (3)$$

And the direction of velocity $\theta_t : \hat{x}_1 \to \hat{x}_2$.

3. Mobile Node Navigation

3.1. Interception Strategy

We suppose the mobile node moves at a constant speed. In order to intercept the intruding target as soon as possible, we need select the optimal direction of movement θ_m according to the instant target. The target may intrude the field of interest at any time with the velocity of $v_t \left(\le v_t^{\max}\right)$. At first, we consider the case of static target, *i.e.* $v_t = 0$. As to the static target, the optimal strategy is intercepting towards the target as shown in **Figure 2**. So the intercept strategy is computed as:

$$\theta_m = \arccos\left(\frac{d_{SM}^2 + d_{MT}^2 - d_{TS}^2}{2 d_{SM} d_{MT}}\right) \quad (4)$$

where d_{SM}, d_{MT}, d_{TS} are Euclidean distances of Static node-Mobile node, Mobile node-Target and Target-Static node.

As to the mobile target, interception strategy needs modified according to the update of target's position and velocity. Based on the velocity estimation in Section 2C, the additional offset of interception direction is computed as:

$$\phi_m = \arcsin\left(\frac{v_t}{v_m} \sin\delta\right) \quad (5)$$

So the optimal strategy of mobile target interception is computed as shown in **Figure 3**:

$$\theta'_m = \theta_m + \phi_m \quad (6)$$

3.2. Target Intercepting

During the interception performance, there are four states for mobile sensor node, *i.e.* WAIT, LONG-DISTANCE: NAVIGATE, SHORT-DISTANCE: TRACKING and INTERCEPTED. In the beginning there is no target intruding into the region, so mobile node is waiting for a command, maybe wandering randomly. When the target appears in the region, CoS node injects the stimulus packet into the sensor network and activates the computation to the position and velocity of the target. Receiving the stimulus Message, mobile node come to the LONG-DISTANCE state and begins to navigate with the guidance of static node nearby. When the sensor of mobile node receives enough signal strength (above η), it

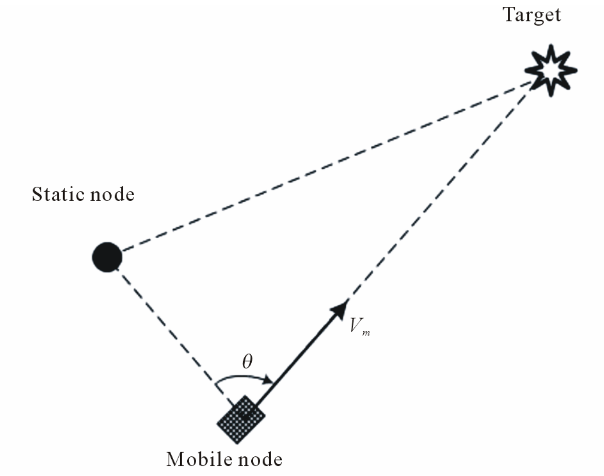

Figure 2. Intercepting static target.

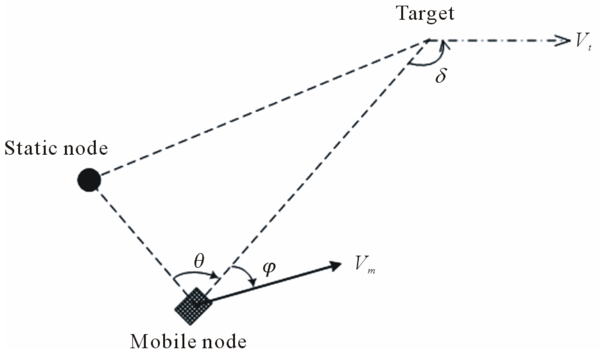

Figure 3. Intercepting mobile target.

will track the target in the current direction. Finally if the position and velocity of mobile node is the same with that of target, mission is completed. Note that when the signal strength received is less than the threshold η, mobile node's state will transit from SHOT-DISTANCE to LONG-DISTANCE, and come back to navigate with the help o f static nodes as shown in **Figure 4**.

3.3. Metrics of Interception

In the mission of target interception, several criteria may be chosen to evaluate the performance, such as minimization of mobile node's energy while guaranteeing capture of all targets or maximization of number of captured targets within a certain amount of time. In this work we focus on minimizing the Time-to-Interception (TI) of mobile node. Since target's motion is not known, exact TI is not known either; therefore we need to define a metric to estimate TI. We will use the following definition of TI:

Definition 1. Let $\left(p_t\left(t_0\right), v_t\left(t_0\right)\right) \in \Re^2 \times \Re^2$ the position and velocity of a target at time $t = t_0$, and $\left(p_m\left(t_0\right), v_m\left(t_0\right)\right) \in \Re^2 \times \Re^2$ the position and velocity of a mobile node at time $t = t_1 \ge t_0$. We define the minimum time TI necessary for the mobile node to reach the target with the same velocity, assuming that the target will keep moving with constant velocity, *i.e.*,

$$TI = \min_T \left\{ T \middle| p_t\left(t_1 + T\right) = p_m\left(t_1 + T\right), v_t\left(t_1 + T\right) = v_m\left(t_1 + T\right) \right\}$$

where

$$p_t\left(t_1 + T\right) = p\left(t_0\right) + \left(t_1 + T - t_0\right) * v\left(t_0\right), v_t\left(t_1 + T\right) = v\left(t_0\right).$$

This definition allows us to quantify TI in an unambiguous way. Although target can change trajectories over time, it is a more accurate estimate than, for example, some metric based on the distance between the target and the mobile node, since TI incorporates the dynamics of mobile node. Moreover, it is well-defined for any arbitrary time delay $t_d = t_1 - t_0$ in the estimate of target's position and velocity relative to current time t_1. Given this definition and the constraints on the dynamics of the mobile node, it is possible to calculate explicitly TI.

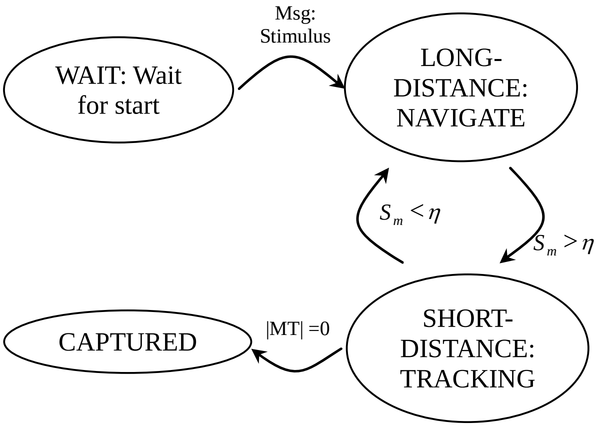

Figure 4. States transition diagram of mobile node.

4. Task Assignment Algorithms

In the previous section, we presented mobile sensor node interception and its metric for an intercepting pair. In this section we consider the case of multiple targets and mobile nodes. Given positions and velocities of all targets and mobile nodes, it is possible to compute TI matrix $C = \left[c_{i,j}\right] \in \Re^{N_m \times N_t}$, where N_m and N_t are the total number of mobile nodes and targets, respectively, and the entry $c_{i,j}$ of the matrix C corresponds to the expected TI between mobile node i and target j. When coordinating multiple mobile nodes to intercept multiple targets, it is necessary to select an assignment. Our objective is to select an assignment that minimizes the expected TI of all mobile nodes.

Assume that we have the same number of mobile nodes and targets, *i.e.* $N_m = N_t$. An assignment can be represented as a matrix $X = \left[x_{i,j}\right] \in \Re^{N_m \times N_t}$, where the entry $x_{i,j}$ of the matrix X is equal to 1 if mobile node i is assigned to target j, and equal to 0 otherwise. The assignment problem can therefore be written formally as follows:

$$\begin{gathered} \min_{x_{i,j} \in \{0,1\}} \max_{i,j=1,\cdots,N} \left(c_{i,j} \times x_{i,j}\right) \\ \text{subject to } \sum_{i=1}^{N} x_{i,j} = 1, \ \sum_{j=1}^{N} x_{i,j} = 1. \end{gathered} \tag{7}$$

As formulated in the equation above, the assignment problem appears as a combinatorial problem.

One simple greedy assignment algorithm that tries to solve the optimization problem above is to look for the smallest TI entry in the matrix C, assign the corresponding interception pair, and remove the corresponding row and column from the matrix C which now becomes of dimension $\left(N-1\right) \times \left(N-1\right)$, and repeat the same process until each mobile node is assigned to a target. Although it is straightforward and easy to implement, this is a suboptimal algorithm, since there are cases when the greedy assignment gives the worst solution.

In this section we present a collaborative task assignment protocol as **Figure 5**. The protocol assigns the task based on probe message in a distributed manner. Previous work [11-13] gives a hierarchical multiple task assignment protocol for the similar problem. But all of them are not distributed and is not scalable well.

Besides, we describe a simple random task assignment algorithm as Randomly Task Assignment (RTA), *i.e.* in

```
Mobile sensor $M_j$
  a) $M_j$ received request messages for blocking targets,
     $T_1,\cdots,T_n$ ,
       Compute the distance from $S_j^m$ to each $T_i$, $D(S_j^m,T_i)$,
       $i=[1\cdots n]$ and sort them in ascending order as
       $D(S_j^m,T_{i_1}),\cdots,D(S_j^m,T_{i_n})$ .and $T_{i_1}=T_p$ .
       Select $T_p$ as the target candidate to block
  b) $M_j$ broadcast Probe Message to all mobile sensors
  c) $M_j$ upon receiving Probe Message $T_p$ from $M_k$
        If $M_j$ will block $T_p$, reply Reject Message
            else if $M_j$ received Probe Message from other
       $M_{k'}$, reply Reject Message
            else reply Accept Message, delete $T_p$ from $M_j$
  d) $M_j$ stay here for a period of time after sending Probe
     Message
       If $M_j$ received Reject Message, pick a larger
       Distance and select its target as candidate to block
            else if received Accept Message, begin to
       block it
            else send Probe Message again, go to b).
```

Figure 5. Pseudo-code of probe message based distributed task assignment.

multiple target and multiple mobile nodes, stimulus assigns every target to one mobile node randomly in spite of negotiation among nodes. We will compare PMB algorithm with RTA in the latter simulation program.

5. Experimental Results

In order to gain a better understanding of landmark-based target interception, we have performed a wide range of simulation studies. In this section, we present several interesting results and discuss their implications and possible applications.

The main simulation platform is written in C++. The visualization and user interface elements are currently implemented with Visual C++ and OpenGL libraries. Network Simulator (ns2) and CrossBow® MICAZ sensor nodes are also used to verify the sensing models and the qualitative performance of the exposure model in a realistic environment. The sensor field in our experiments is defined as a 500 * 500 square. 80 static nodes are randomly deployed in the region.

For simplicity, parameters θ_t and v_t is set to 0 and $v_t^{\max}$ respectively. Suppose a target intrudes from a point of the left edge and monitored by the networked sensors. The mobile node starts interception from the midpoint of the right edge. As shown in **Figure 6**, we calculate the intercepting path with v_m of 10 unit/s and 8 unit/s respectively. And the velocity of the target $v_t^{\max}$ is 6 unit/s.

If the target intrudes the field from different points, mobile node needs different time for interception. Intruding point of target has a great impact on the performance of algorithms. We conducted 50 independent trials and the outcome is averaged.

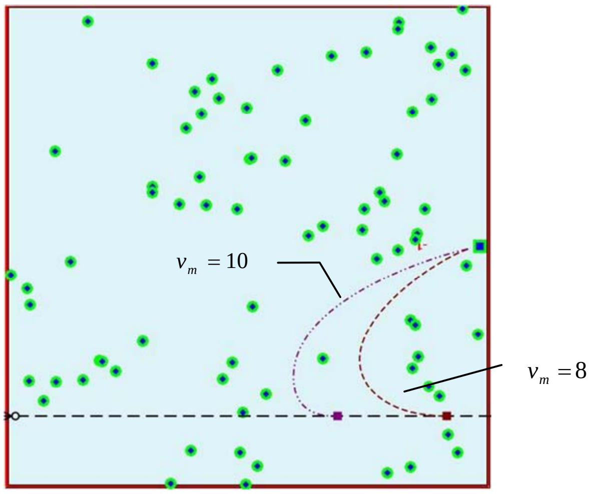

Figure 6. Intercepting paths with different speed v_m .

The mobile node is deployed over the field randomly. When the target intrudes the field of interest from the left edge with 0, 100, 200, 300, 400 and 500 units, TI for target interception is shown in **Figure 7**. From **Figure 7**, we can conclude that TI is largest when the target intrudes along the upper boundary (*i.e.* 0 unit) or bottom boundary (*i.e.* 500 units). This is because the mobile node is deployed randomly over the field of interest. When the target intrudes the field along the upper or bottom boundary, the relative distance between the target and the mobile node is the largest, so it needs the largest TI for interception. On the contrary, if the target intrudes from the midline point (*i.e.* 300 units), TI for interception is smaller.

The mobile node waits for the stimulus for interception from the original position and begins to intercept the target. The original position has important impact on the

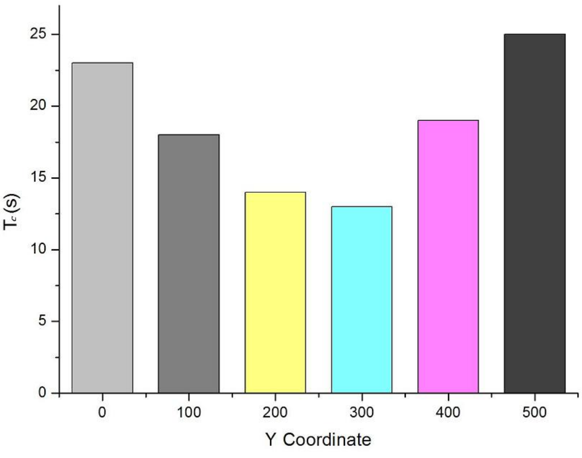

Figure 7. TI vs. different intruding point.

TI. We build Cartesian coordinate in the monitored region with the upper-left point as original point. And we select 9 referred points as the start point of the mobile node (See **Table 1**). Intruding point of the target is selected randomly from the left boundary and we conducted 50 independent trials. The average of TI is shown in **Figure 8**. From **Figure 8**, we can see that the mobile node needs least time for interception when starting from the referred point 5, *i.e.* TI of the mobile node from the center point of the monitored region is the smallest. So we can conclude that the center point of the region is the best guard point for the mobile node and the referred points of 1 and 7 are the worst.

In order to validate the effectiveness of PMB multiple task assignment, we conducted simulation as follows. Suppose the number of the targets is equal to that of the mobile nodes, *i.e.* $N_m = N_t = N$. Velocity of all targets is 6 unit/s and velocity of all mobile nodes is 8 unit/s. All mobile nodes are deployed over the field randomly and all targets intrude the field from the random point of the left edge. We conducted 50 independent trials and the outcome is averaged. TI changes with the number of targets N as shown in **Figure 9**. From **Figure 9**, we can see that TI of PMB algorithm reduces with the number of targets/ mobile nodes N grows, while TI of RTA algorithm almost holds the line. Besides, the advantage of PMB algorithm appears more distinctly when N grows. With the number of targets N growing, TI of PMB algorithm decrease more slowly. It can be explained by the fact that TI is mainly determined by the distance between the entry point and escaping point of the target. Because the distance of entry-escaping points is fixed, TI of PMB algorithm is asymptotic with the expected time of interception between target and mobile node with the same Y coordinates.

6. Conclusions and Future Works

In this paper we consider target interception and propose a sign-based strategy to solve it using hybrid sensor networks. In our approach, static sensor nodes detect the target, compute the velocity and guide the mobile node. With the help of static node nearby, the mobile node transits between four states and manage the interception. In addition, we consider multiple targets and mobile nodes, and present a collaborative task assignment protocol to minimize the time of interception. Obstacles may appear in the field of interest, other barrier may be a trap for mobile nodes, e.g. pond. Hence, future work includes interception using mobile nodes with obstacle avoidance.

Table 1. Starting points of the mobile node.

Referred point	Coordinates (unit, unit)
1	(500,0)
2	(250,0)
3	(0,0)
4	(500,250)
5	(250,250)
6	(0,250)
7	(500,500)
8	(250,500)
9	(0,500)

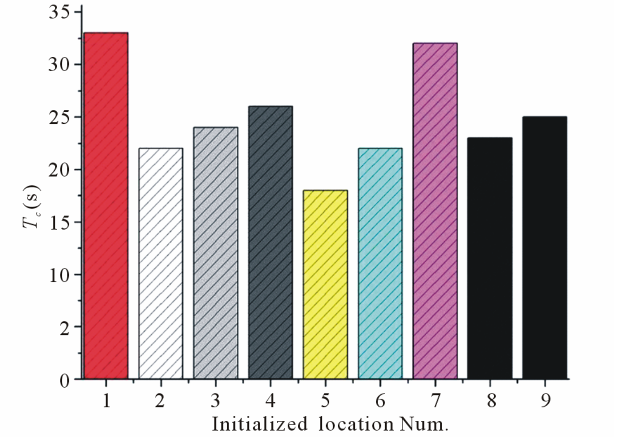

Figure 8. TI vs. different starting point of mobile node.

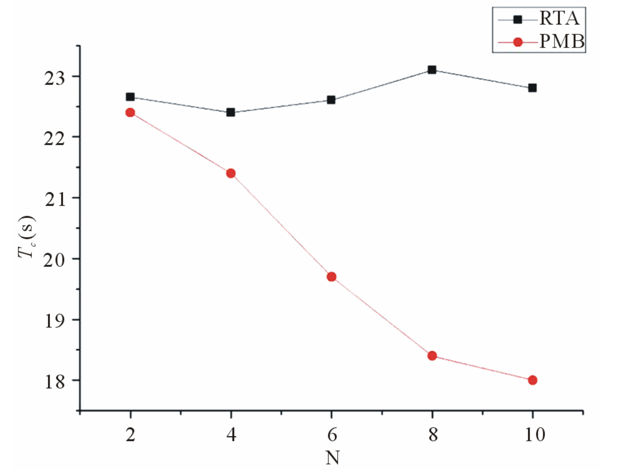

Figure 9. TI of two task allocation algorithms vs. N.

7. References

[1] E. Biagioni and K. Bridges, "The Application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species," *International Journal of High Performance Computing Applications*, Vol. 16, No. 3, 2002, pp. 315-324. doi:10.1177/10943420020160031001

[2] M. A. Batalin, G. S. Sukhatme and M. Hattig, "Mobile Robot Navigation Using a Sensor Network," *Proceedings*

of the IEEE International Conference on Robotics and Automation, New Orleans, April 2003, pp. 636-642.

[3] J. Borenstein and H. R. Everett, "Navigating Mobile Robots: Sensors and Techniques," A. K. Peters Ltd., Wellesley, 1992.

[4] Q. Li, M. D. Rosa and D. Rus, "Distributed Algorithms for Guiding Navigation across a Sensor Network," *Proceedings of the* 9*th Annual International Conference on Mobile Computing and Networking*, San Diego, 2003, pp. 313-325.

[5] A. J. Briggs, C. Detweiler, D. Scharstein and A. Vandenberg-Rodes, "Expected Shortest Paths for Landmark-Based Robot Navigation," *International Journal of Robotics Research*, Vol. 23, No. 7-8, July 2004, pp. 717-728. doi:10.1177/0278364904045467

[6] A. Verma, H. Sawant, J. Tan, "Selection and Navigation of Mobile Sensor Nodes Using a Sensor Network," *Proceedings of Third IEEE International Conference on Pervasive Computing and Communications*, Pisa, March 2005, pp. 41-50.

[7] J. Borenstein and Y. Koren, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," *IEEE Journal of Robotics and Automation*, Vol. 7, No. 3, 1991, pp. 278-288. doi:10.1109/70.88137

[8] J. Liu, J. Liu, M. Chu, J. E. Reich and F. Zhao, "Distributed State Representation for Tracking Problems in Sensor Networks," *Proceedings of third Workshop on Information Processing in Sensor Networks* (*IPSN*), New York, April 2004, pp. 234-242.doi:10.1145/984622.984657

[9] J. Liu, J. Liu, J. Reich, P. Cheung and F. Zhao, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *Proceedings of* 2*nd* *workshop on Information Processing in Sensor Networks (IPSN)*, Berlin, April 2003, pp. 113-128.

[10] S. Meguerdichian, F. Koushanfar, G. Qu and I. Potkonjak, "Exposure in Wireless Ad-Hoc Sensor Networks," *Proceedings of 7th Annual International Conference on Mobile Computing and Networking*, New York, July 2001, pp. 139-150.

[11] S. Oh, L. Schenato and S. Sastry, "A Hierarchical Multiple-Target Tracking Algorithm for Sensor Networks," *Proceedings of the International Conference on Robotics and Automation*, Barcelona, April 2005, pp. 2197- 2202.

[12] L. Schenato, S. Oh, P. Bose and S. Sastry, "Swarm Coordination for Pursuit Evasion Games Using Sensor Networks," *Proceedings of the International Conference on Robotics and Automation*, Padova, April 2005, pp. 2493-2498. doi:10.1109/ROBOT.2005.1570487

[13] J. P. Hespanha, H. J. Kim and S. S. Sastry, "Multiple-Agent Probabilistic Pursuit-Evasion Games," *Proceedings of the IEEE International Conference on Decision and Control*, Phoenix, 1999, pp. 2432-2437.