Solving heterogeneous coverage problem in Wireless Multimedia Sensor Networks in a dynamic environment using Evolutionary Strategies

Hossein Fayyazi Mohammad Sabokrou Mojtaba Hosseini Ali Sabokrou
Dept. of ICT Dept. of ICT Dept. of Computer Engineering Dept. of Control Engineering
Malek-Ashtar University of Technology Malek-Ashtar University of Technology Amirkabir University of Technology Islamic Azad University-Gonabad
Tehran, Iran Tehran, Iran Tehran, Iran Gonabad, Iran
e-mail: fayyazi_hosseim@yahoo.com e-mail: sabokro@gmail.com e-mail: mojtabahoseini@aut.ac.ir e-mail: a_sabokroo@yahoo.com

Abstract—Wireless sensor networks have become a rapidly developing research area that offers some ways of monitoring the environment and bridging the gap between the physical and virtual world. Visual Sensor Networks consist of a large number of low-power camera nodes to monitor a general environment with some targets in it. One of the most important problems in this context is energy conservation and covering the entire targets. This paper uses Evolutionary Strategy as a method based on population gradual adjustment by environmental conditions to solve the heterogeneous coverage problem in Wireless Multimedia Sensor Networks (WMSNs) in a dynamic environment by mobile targets with minimum energy consumption.

Keywords—Evolutionary Strategy; Wireless Sensor Networks; coverage problem; mobile targets; energy consumption.

I. INTRODUCTION

Advances in wireless communication and Micro Electro Mechanical Systems (MEMS) have enabled the development of low-cost, low-power, multi-functional, tiny sensor nodes which can sense the environment, perform data processing and communicate with each other over short distances [1].

The main and most important reason for Wireless Sensor Networks (WSNs) development was for continuous monitoring of environments where are difficult or impossible for human being to access or stay for a long time; Monitoring of environments like the head of an active volcano, difficult terrain border lands, bridges, battlefields, roads, sluices etc. So, normally, there is often a low possibility to replace or recharge the dead nodes as well. The other important requirement is that in most applications of WSNs, we need a continuous monitoring, so the lifetime and network coverage of these networks are our great concerns since the performance of WSNs severely depends on their lifetime. Therefore, energy conservation is a serious and critical issue in designing of WSNs with longevity [2].

Power consumption is a fundamental concern in Wireless Multimeda Sensor Networks, even more than in traditional WSNs. In fact, sensors are battery-constrained devices, while multimedia applications produce high volumes of data, which require high transmission rates, and extensive processing. While the energy consumption of traditional sensor nodes is known to be dominated by the communication functionalities, this may not necessarily be true in WMSNs [3].

In the 1960s, Rechenberg introduced "evolution strategies", a method he used to optimize real-valued parameters for devices such as airfoils. The field of Evolutionary Strategies has remained an active area of research, mostly developing independently from the field of genetic algorithms [4].

This paper uses Evolutionary strategy to solve the heterogeneous coverage problem in WMSNs in a dynamic environment by mobile targets to minimize total energy consumption of sensors. In heterogeneous coverage problem, each target should be covered by at least k sensors that k are variable for different targets and determined by user demand or application. We also assume that each sensor has the PAN and zoom capability.

The remainder of the paper is organized as follows. We will start by presenting some related works in section II. Section III provides a more detailed introduction of heterogeneous coverage problem in WMSNs in a dynamic environment by mobile targets. Section IV introduces Evolutionary Strategies as a special instance of Evolutionary Algorithms. Section V gives a description of our solution, and experimental results and conclusion are provided in VI and VII sections, respectively.

II. RELATED WORKS

In the past few years, there have been a number of proposed algorithms for coverage preservation in Wireless Sensor Networks. The most commonly used approach to handle this problem is to determine the redundant nodes (sensors) and put them to sleep [5].

A survey on the coverage problem in Video-based WSNs and WMSNs are presented in [6] and [3], respectively. A target coverage scheduling scheme based on genetic algorithms that can find the optimal cover sets to extend the network lifetime while monitoring all targets in Directional Sensor Networks (DSNs) is proposed in [7].
and another coverage control scheme based on multi-objective genetic algorithm with the goal of energy conservation is introduced in [8]. In [9], the maximization of cover sets was modeled to extend the network lifetime of WSNs and then optimized by genetic algorithms. A solution for the problem of finding the maximum number of covers using memetic algorithm is proposed in [10]. Reference [11] solves the coverage problem in Wireless Camera-based Sensor Networks using genetic algorithm.

However, above related works do not consider assumptions like target mobility, heterogeneous coverage and the PAN and zoom capability for a sensor that is closer to the real world, whereas we consider them.

III. PROBLEM DESCRIPTION

Consider a number of visual sensors that have been strewed randomly in an area of earth with random angles. There are some targets in the environment that are displaced randomly over each unit of time. Each target should be covered by k visual sensor that k represents a number between 1 to 5. Appointment of k is based on user demand or application. Each visual sensor can zoom and rotate 180 degrees by lateral.

The main objective of the WMSNs is to cover a set of mobile targets that need to be monitored by consuming minimum energy. The energy of each sensor is consumed by zooming and rotating.

Table I summarizes the notations introduced along with some minor additions.

Figs. 1 – 3 show some notations in Table I visually. Each sensor has a variable range and angle of view that zoom level specifies them.

The solid circle in Fig. 3 is a target that can be covered by the sensor that is placed in \((x_s, y_s)\) coordinates. The only parameter available for target is its coordinate in environmental space. Let the coordinate of target to be \((x_t, y_t)\).

<table>
<thead>
<tr>
<th>symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Number of sensors</td>
</tr>
<tr>
<td>T</td>
<td>Number of targets</td>
</tr>
<tr>
<td>(k_j)</td>
<td>Minimum number of sensors that should sense target j</td>
</tr>
<tr>
<td>(\theta_j)</td>
<td>Initial angle of sensor j with the positive x-axis</td>
</tr>
<tr>
<td>(a_j)</td>
<td>Rotation angle of sensor j</td>
</tr>
<tr>
<td>(R)</td>
<td>Initial range of sensors</td>
</tr>
<tr>
<td>(\beta)</td>
<td>Initial angle of view</td>
</tr>
<tr>
<td>(\delta)</td>
<td>Target’s angle</td>
</tr>
<tr>
<td>L</td>
<td>Number of zoom levels</td>
</tr>
<tr>
<td>(R_i)</td>
<td>Range of sensor in i’th zoom level</td>
</tr>
<tr>
<td>(i)</td>
<td>i’th zoom level</td>
</tr>
<tr>
<td>(\beta_i)</td>
<td>Angle of view in i’th zoom level</td>
</tr>
<tr>
<td>OS</td>
<td>Number of On sensors</td>
</tr>
<tr>
<td>ST_j</td>
<td>Number of sensors that sense target j</td>
</tr>
<tr>
<td>ZL_j</td>
<td>Zoom level of j’th sensor</td>
</tr>
</tbody>
</table>

The Euclidian distance between sensor and target and the angle of target with the positive x-axis can be calculated.

\[
\text{Euclidian Distance}(s,t) = \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2} \tag{1}
\]

\[
\delta = \tan^{-1} \frac{y_s - y_t}{x_s - x_t} \tag{2}
\]

The target is covered by the sensor, if at least for one zoom level of a sensor the (3),(4) be satisfied.

\[
\text{Euclidian Distance}(s,t) < R \tag{3}
\]

\[
(\theta + \alpha) - \beta / 2 < \delta < (\theta + \alpha) + \beta / 2 \tag{4}
\]

We suppose the following equations for range and angle of view of a sensor in different zoom levels.

\[
R_i = R \sqrt{1 + \frac{\beta_i^2}{\beta^2}} \tag{5}
\]
\[ \beta^i = \frac{\pi}{2} - 2 \tan^{-1} \frac{i}{i} \]

IV. EVOLUTIONARY STRATEGIES

This section provides a description of Evolutionary Strategies (ESs) as a special instance of evolutionary algorithms [12].

ES have a very useful feature in Evolutionary Computing, self-adaptation of strategy parameters. In general self-adaptivity means that some parameters of the algorithm are varied during a run in a specific manner: the parameters are included in the chromosomes and coevolve with the solutions [13].

The overall form of chromosomes in ES is considered as real-valued vectors. To create one child from two or more parents, recombination operator selects one of the parental values randomly (discrete recombination) or averages parental values (intermediary recombination). There is a strong emphasis on mutation for creating offspring in ES. The main mechanism of mutation operator is to change value by adding random noise drawn from normal distribution.

Parents are selected by uniform random distribution. Survivor selection is applied after creating \( \lambda \) children from the \( \mu \) parents by mutation and recombination. Basis of selection is either the set of children only \((\mu, \lambda)\) selection or the set of parents and children \((\mu+\lambda)\) selection.

\((\mu+\lambda)\) selection is an elitist strategy, while \((\mu, \lambda)\) selection can “forget”. Often \((\mu, \lambda)\) selection is preferred because it is better in leaving local optima and following moving optima. In ES the number of children is much greater than the number of parents \((\lambda \approx 7 \times \mu \text{ is the common setting})\).

V. PROBLEM SOLVING USING EVOLUTIONARY STRATEGIES

Now we solve the introduced problem in section III using evolutionary strategies. As mentioned in previous sections, the most important feature of ES is gradual adjustment of evolutionary parameters. Therefore, these parameters must be placed in chromosomes in some way. In this paper mutation step size is placed in chromosome representation in two ways. Fig. 4 shows these forms, where \( \alpha_i \) is the direction of \( i \)’th sensor and \( \text{OnOff}_i \) shows that it is on or off.

At the first representation just one \( \sigma \) is considered for each chromosome, whereas \( \sigma \) is different for each sensor at the second one.

A. Recombination operator

As mentioned in section IV, two types of recombination can be done in ES, intermediary and discrete recombination. We use the intermediary recombination to solve our problem because: the averaging effect of intermediate recombination assures a more cautious adaptation of strategy parameters [13]. One of the vectors of introduced chromosome is of type boolean, and the round average of two binary digits cause the sensors to be on state of on (see Table II).

<table>
<thead>
<tr>
<th>\text{OnOff}_1</th>
<th>\text{OnOff}_2</th>
<th>\text{Intermediary(OnOff)}</th>
<th>\text{Round}</th>
<th>\text{XOR(OnOff)}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>((0 + 0)/2 = 0.0)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>((0 + 1)/2 = 0.5)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>((1 + 0)/2 = 0.5)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>((1 + 1)/2 = 1.0)</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Therefore, we use the Exclusive-OR of two boolean values to create the child. Other vectors of chromosome are real values and can be averaged normally.

B. Mutation operator

There are some special cases of mutation in ES that two types of them are used in this paper.

Case 1, Uncorrelated mutation with one step size [13]:

\[
\sigma' = \sigma \cdot \exp(\tau \cdot N(0,1))
\]

\[
x'_i = x_i + \sigma' \cdot N(0,1)
\]

\[
\tau \propto 1/n^\gamma
\]

\[
\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0
\]

Case 2, Uncorrelated mutation with n step sizes [13]:

\[
\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))
\]

\[
x'_i = x_i + \sigma'_i \cdot N_i(0,1)
\]

\[
\tau' \propto 1/(2n)^\gamma \text{ and } \tau \propto 1/(2n^\gamma)^\gamma
\]

\[
\sigma'_i < \varepsilon_0 \Rightarrow \sigma'_i = \varepsilon_0
\]

Where \( \sigma' \) is the new mutation step size, \( N(0,1) \) is a random variable drawn from a standard normal distribution. \( \tau \) is the learning rate and \( \tau' \) is the overall learning rate. Equations (10),(14) present boundary rules [13].

For mutation, a random number is generated and then compared with mutation step size to determine whether or not that chromosome is mutated. If the chromosome should be mutated, mutation operation is done according to the formula above.

C. Fitness evaluation

Whatever each target covered by more sensors and rotation angle and zoom level of sensors is less, energy consumption is less. Suppose that a target should be covered by \( k = 3 \) sensors and instead be covering by 6 sensors, half of the sensors that cover the target can be on state of idle in one unit of time and power on in another and so, energy consumption is decreased.

These criteria are used to determine fitness value of each chromosome in our solution (15).

\[
\text{Fitness} = \sum_{i=1}^{\text{OS}} \frac{x_i - n_i}{2} + 2 \cdot \sum_{j=1}^{\text{ST}} \frac{k_j}{k_j} - \frac{\text{ST}_j}{\text{ST}_j} + \sum_{i=1}^{\text{OS}} \frac{1-ZL_j}{1}
\]

The first expression of (15) applies the goal of PAN minimization while, the second wants to increase target coverage level. We have multiplied by two in the second expression because it has more importance for us. Finally we apply the third expression as zoom level minimization criteria.

VI. Experimental results

In this section, we present the experimental results for evaluating the proposed algorithm. The solution presented in this paper is implemented by C# programming language. The algorithm has been executed with 200 generations and different number of sensors and population sizes. Table III represents average results of 5 implementations for each experiment.

In all cases, the width and length of environment are \( 100 \times 100 \). Initial range and angle of sensor are 10 and 90°, respectively and 10 levels of zooms are considered. There are 10 targets in the environment that 10 percent of them are displaced randomly in the eight possible directions over each unit of time. Each target should be covered by \( k \) visual sensor that \( k \) represents a number between 1 to 5. In our simulations, appointment of \( k \) is randomly.

Considering that the dynamic nature of environment, we have no information about an optimal fitness level to consider it as stopping criteria. Therefore, we use the number of generations as the termination condition. Fig. 5 shows the progress of the algorithm through the 200 generations.

VII. Conclusion

Simulation results showed that evolutionary strategies are powerful tools for continuous parameter optimization problems like ours. In other words, if the problem at hand given as an objective function then ES can solve it easily.

The motivation behind using n step sizes in chromosome representation is the wish to treat dimensions of chromosomes differently. We can see that uncorrelated mutation with one step size (Case 1) works better than n step sizes (Case 2) in this problem.

![Figure 5. Progress of the algorithm in 200 generations](image-url)
### Table III. Experimental Results

<table>
<thead>
<tr>
<th>Population size</th>
<th>Number of sensors</th>
<th>Mutation type</th>
<th>Number of found solutions</th>
<th>Max fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>300</td>
<td>Case 2</td>
<td>0</td>
<td>7.02</td>
</tr>
<tr>
<td>10</td>
<td>350</td>
<td>Case 2</td>
<td>1</td>
<td>8.34</td>
</tr>
<tr>
<td>10</td>
<td>400</td>
<td>Case 2</td>
<td>64</td>
<td>8.83</td>
</tr>
<tr>
<td>10</td>
<td>450</td>
<td>Case 2</td>
<td>51</td>
<td>9.72</td>
</tr>
<tr>
<td>10</td>
<td>500</td>
<td>Case 2</td>
<td>108</td>
<td>10.17</td>
</tr>
<tr>
<td>10</td>
<td>300</td>
<td>Case 1</td>
<td>4</td>
<td>8.56</td>
</tr>
<tr>
<td>10</td>
<td>350</td>
<td>Case 1</td>
<td>55</td>
<td>9.19</td>
</tr>
<tr>
<td>10</td>
<td>400</td>
<td>Case 1</td>
<td>80</td>
<td>9.94</td>
</tr>
<tr>
<td>10</td>
<td>450</td>
<td>Case 1</td>
<td>74</td>
<td>9.33</td>
</tr>
<tr>
<td>10</td>
<td>500</td>
<td>Case 1</td>
<td>170</td>
<td>10.58</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
<td>Case 2</td>
<td>0</td>
<td>7.23</td>
</tr>
<tr>
<td>5</td>
<td>350</td>
<td>Case 2</td>
<td>1</td>
<td>7.67</td>
</tr>
<tr>
<td>5</td>
<td>400</td>
<td>Case 2</td>
<td>3</td>
<td>7.87</td>
</tr>
<tr>
<td>5</td>
<td>450</td>
<td>Case 2</td>
<td>9</td>
<td>8.24</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>Case 2</td>
<td>19</td>
<td>8.70</td>
</tr>
<tr>
<td>5</td>
<td>300</td>
<td>Case 1</td>
<td>2</td>
<td>8.14</td>
</tr>
<tr>
<td>5</td>
<td>350</td>
<td>Case 1</td>
<td>31</td>
<td>8.21</td>
</tr>
<tr>
<td>5</td>
<td>400</td>
<td>Case 1</td>
<td>17</td>
<td>8.97</td>
</tr>
<tr>
<td>5</td>
<td>450</td>
<td>Case 1</td>
<td>15</td>
<td>9.05</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>Case 1</td>
<td>125</td>
<td>10.21</td>
</tr>
</tbody>
</table>

### References


