WIRELESS SENSORS NETWORK MODELLING AND OPTIMIZATION

A JAVA BASED MODIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM FOR WIRELESS SENSORS DEPLOYMENT

Introduction

There is an increasing application of wireless sensor networks (WSNs) to the sensing of physical, chemical and biological parameters around us. It replaces the use of wired sensor networks introducing cost savings in terms of cabling costs and ease of deployment and redeployment. Network area coverage using minimal sensor nodes is a major design objectives of WSNs. A modified particle swarm optimization algorithm combining virtual force model and adaptive parameter control is introduced to optimize the coverage of a dynamic WSN.

Aims and Objectives

The project aims and objective is summarized by figure 1 below.

Network Model

- The network area was divided into grid cells each of approximately 0.04% of the total area of the region of interest.
- The point coverage of the centers of the grid cells is assumed to be same for all points within a given grid cell.
- This point coverage, \( p(S_{ij}) = 1 - \prod_{k=1}^{N_{ij}} (1 - p(s_{ij})) \) where \( p(s_{ij}) \) is the probability of the point \( j \) detection by sensor \( i \), is obtained using Elfes probabilistic sensing model.
- \( N \) is the total number of sensors in the network.
- A threshold, \( Cth \) for effective point coverage of the grid cell centers is \( \sum_{i=1}^{N} p(s_{ij}) \geq Cth \) indicates that the grid cell center is effectively covered.

This threshold which depends on the end user requirements is specified during the network design stage. A threshold of 90% coverage was used in this project.

- The area coverage of the network is then obtained as

\[
\text{Area Coverage} = \frac{\text{Total Number of grid cells} - \text{Number of grid cells effectively covered}}{\text{Total Number of grid cells}}
\]

Modified Particle Swarm Optimization Algorithm

- A virtual force model and adaptive parameter modification to ensure uniform sensors distribution, avoid obstacles within the RoI and to escape local optimal solutions is incorporated in the classical particle swarm optimization by Kennedy and Eberhart[1].
- The virtual forces on the sensors, \( F = F_{ij} + F_{obsn} \) Where \( F_{ij} \) is the force of mutual interaction between sensor \( i \) and \( j \) which can be attractive or repulsive depending on the distance between the sensors, the communication range of the sensors and the threshold of the virtual force model. \( F_{obsn} \) is the repulsive force between sensor \( i \) and obstacle \( obsn \).
- The evolutionary factor is computed as according to Zhi-Hui Zhan et al in [2]:

\[
f = \frac{d_{max} - d_{min}}{d_{max} - d_{min}}
\]

and used to estimate the evolutionary state of the algorithm and effect the necessary control on the social and cognitive acceleration constants[2]. The inertia weight is adapted using:

\[
w(f) = \frac{1}{1 + e^\frac{-f}{1.5}} \in [0.4, 0.9] \quad \forall f \in [0, 1] \quad [2].
\]

Simulation and Results

- Fig. 2: Variation of Area Coverage with Iteration using Algorithm
- Fig. 3: Variation of Area Coverage with Iteration using Random deployment
- Fig. 4: Variation of Evolutionary Factor with Iteration
- Fig. 5: Variation of Inertia Weight with Iteration

Conclusion

As shown in fig.2 above, the improvement in the network area coverage between the green, blue and red coloured graphs as a result of an increase in the number of deployed sensors by ten in each case is far more profound than that obtained using random deployment as depicted in fig3. The evolutionary factor and hence inertia weight and acceleration constants are also shown to be adapted as shown in fig 4 and fig 5.

References