Particle Swarm Optimization for clustering in Ad Hoc sensor networks

Jason Tillett, Raghuveer Rao and Ferat Sahin
Rochester Institute of Technology
Rochester NY 14623
jcteee@osfmail.rit.edu, mrreee@rit.edu, feseee@rit.edu

T. M. Rao
SUNY at Brockport
Brockport, NY
trao@brockport.edu

Rochester Institute of Technology – Kate Gleason College of Engineering, work partially funded by LACOMS
Group’s Current Network Thrust

- Self-Similar Traffic Modeling.
- Evolutionary programming and parallel computing approaches. Particle swarm optimization is an instance.
Overview

- Problem and motivation.
- The Particle Swarm Approach
- Our Algorithm
- Results
The Problem

- Form $M$ clusters from a group of $N$ nodes where $A$ nodes are available to take on the role of cluster head.

- Divide the nodes into clusters to equalize the number of nodes and candidate cluster-heads in each region.

- Do the divisions so that resulting clusters will contain nodes that are spatially nearby.

- Notion of clusterheads derived from LEACH (Heinzelman et al).
The PSO Approach

• A swarm of test solutions cooperate and interact to perform a “social” style search of parameter space for solutions to the proposed problem.

• Ingredient List.

  • particle construct which fully represents the solution
  • fitness calculator, need to be able to turn particles into fitness
  • dynamic response of the swarm
The PSO Particle

Choice of particle representation: binary or continuous?

For continuous variables, choice of fitness needs to be correlated with the variables.

The PSO Fitness

Given a PSO particle, the particle must be able to evaluate “how good” its test solution is. The particle in particular must know when another neighboring particle has a better test solution.
The PSO Dynamical Swarm Response

\[ v(t + 1) = v(t) + \phi_1 (x - x_p) + \phi_2 (x - x_n) \]

\[ x(t + 1) = x(t) + v(t + 1) \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v )</td>
<td>The particle velocity.</td>
</tr>
<tr>
<td>( x )</td>
<td>The particle position (test solution).</td>
</tr>
<tr>
<td>( t )</td>
<td>time</td>
</tr>
<tr>
<td>( \phi_1 )</td>
<td>A uniform random variable usually distributes over [0,2].</td>
</tr>
<tr>
<td>( \phi_2 )</td>
<td>A uniform random variable usually distributes over [0,2].</td>
</tr>
<tr>
<td>( x_p )</td>
<td>The particle’s position (previous) that resulted in the best fitness so far.</td>
</tr>
<tr>
<td>( x_n )</td>
<td>The neighborhood position that resulted in the best fitness so far.</td>
</tr>
</tbody>
</table>
Our Algorithm

Suboptimize using splitting. Split the nodes in half (M even). Simple.

Keep splitting resulting collections of nodes.

Stop when we have M clusters.

From each final cluster, select the most central node as cluster-head from the available cluster-heads.
Dividing the Sensors

1. Division 1: split into 2 regions. In one region, 3 clusters are required, in the other, 2 clusters are required.
2. Division 2: split the ‘2 cluster’ region.
3. Division 3: split the ‘3 cluster’ region. In one region, 2 clusters are required, in the other, 1 cluster is required.
4. Division 4: split the remaining ‘2 cluster’ region into 2 ‘1 cluster’ regions
Our line represents the line dividing the region into 2 smaller regions.

\[ \vec{p}_i = \{ x_i, y_i, \theta_i \} \]
Our Fitness

\[ \text{Fitness} = (a_1 - f_1 A)^2 + (a_2 - f_2 A)^2 + (c_1 - f_1 N)^2 + (c_2 - f_2 N)^2 \]

Our fitness gets better if the number of nodes AND available cluster-heads is the “same” in each region.

We only accept a division if there are adequate cluster-heads in the remaining regions to perform subsequent divisions that are required to reach \(M\) clusters.

\[ f_i = \frac{M_i}{M} \]

explain the variables please…
Results

• Natural Cluster Identification
• Clustering for a Uniform Node Distribution
• Comparison to k-means
Natural Cluster Identification – Numerical Results

Optimum configuration found in 80% of 26 trials.

Correctly identified (put a cluster-head in) natural clusters in 87% of the 26 trials.

Only 13% of runs resulted in 2 cluster-heads being assigned to the same natural cluster.

On less than 1% of the runs, more that 2 cluster-heads were assigned to a single natural cluster.
Uniformly Distributed Nodes – Numerical Results

Division of 100 Node Network

Convergence of PSO clustering for different set of sensors
## Comparison to k-means Algorithm

<table>
<thead>
<tr>
<th></th>
<th>PSO</th>
<th></th>
<th>k-means</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;d&gt;</td>
<td>&lt;sig&gt;</td>
<td>&lt;d&gt;</td>
<td>&lt;sig&gt;</td>
</tr>
<tr>
<td>set 1</td>
<td>18.4%</td>
<td>9.2%</td>
<td>15.8%</td>
<td>10.68%</td>
</tr>
<tr>
<td>set 2</td>
<td>18.4%</td>
<td>9.4%</td>
<td>16.32%</td>
<td>10.6%</td>
</tr>
<tr>
<td>set 3</td>
<td>19.32%</td>
<td>9.12%</td>
<td>16.96%</td>
<td>11.2%</td>
</tr>
</tbody>
</table>
Future Work

- Use ns to simulate the performance of a sensor network where the cluster-heads are selected using our algorithm

- Bayesian Network Space Searching Using Particle Swarm Optimization

- Run PSO optimizations on a computer cluster (next slide)

thank you…
PSO Network Building

Ported a scaled down version of portions of the code to SUN.

Wrote MPI “Master / Slave” type application.

Able to execute network building using a 34 node SUN Blade Cluster in the Center for Imaging Sciences.
A plug

  • http://spie.org/conferences/calls/03/or/
  • Abstracts still being accepted.