Communication-Oriented Distributed Particle Swarm Optimization

Logan Perreault, Mike P. Wittie, John Sheppard
Montana State University
Particle Swarm Optimization

State Update:

- A particle’s state represents a potential solution.
- $x_i = v_i + x_i$

Velocity Update:

$$v_i = \omega v_i + U(0, \phi_1) \otimes (p_i - x_i) + U(0, \phi_2) \otimes (p_g - x_i)$$
Particle Swarm Optimization

State Update:

- A particle’s state represents a potential solution.
- $x_i = v_i + x_i$

Velocity Update:

$$v_i = \omega v_i + U(0, \phi_1) \otimes (p_i - x_i) + U(0, \phi_2) \otimes (p_g - x_i)$$

$$v_i = \omega v_i + \Phi_1 \text{Cognitive}() + \Phi_2 \text{Social}()$$
Distributed PSO

**PSO for Robotic Swarms (dPSO)**

Instead of virtual particles, use physical robots.
- unreliable update
- propagation of global best

**Previous Work**

J. M. Hereford, 2006

J. Pugh and A. Martinoli, 2008
Motivation

Previous Work Did Not Consider Mobile Targets

Previous experiments only attempted to track static targets.

- How well can dPSO track a moving target?
- What adaptations need to be made to achieve this?

Previous Work Ignores Server Communication

Standard dPSO communicates global best to other particles.

- Only useful if you can retrieve the information
- Transmit solutions back to a server
Problem Example
Goal: Periodically relay the current best solution to a central server.

Introduce a Communication Goal

The approach we take is to integrate a communication goal into the fitness function.

- Cognitive and social terms draw particles toward solution
- Communication term draws particles toward server
Notation

\( c \)

Target number of timesteps before communication with the server is restored
\( c \)

Target number of timesteps before communication with the server is restored

\( \theta_i \)

Unique offset for each particle

\[
\theta_i = R_{id} \times \frac{c}{N}
\]
\( c \)
Target number of timesteps before communication with the server is restored

\( \theta_i \)
Unique offset for each particle
\[
\theta_i = R_{id} \times c / N
\]

\( t_c \)
The last timestep where successful communication with the server was made
Communication-Oriented Velocity Update Rule

\[ v_i = \omega v_i + \left( 1 - \min \left( 1, \left\lfloor \frac{t - t_c}{c + \theta_i} \right\rfloor \right) \right) \Phi_1 \text{Cognitive()} \]

\[ + \left( 1 - \min \left( 1, \left\lfloor \frac{t - t_c}{c + \theta_i} \right\rfloor \right) \right) \Phi_2 \text{Social()} \]

\[ + \min \left( 1, \left\lfloor \frac{t - t_c}{c + \theta_i} \right\rfloor \right) \Phi_3 \text{Communication()} \]
The new communication term requires a new fitness function.

- Defined by the number of hops $H$ required to reach the server.
- Exponential decrease in fitness as hops increase.

$$fitness_c = \frac{1}{d^H}$$
Our Approach to Tracking Dynamic Targets

Goal: Find and track targets that are in motion.

**Decay**

We decay the three fitness values for each particle as follows:

- $f_i \leftarrow \beta_1 f_i$
- $f_g \leftarrow \beta_2 f_g$
- $f_c \leftarrow \beta_3 f_c$
Experimental Design

We implemented \textit{dPSO} and \textit{C-dPSO} in a simulated environment.

- communication range limited to 25\% of search space
- notion of global best must be propagated

We conducted 9 experiments to test our contributions.

- 8 particles, 1 server, 1 target
- 500 noisy points with fitness from 0\% to 5\% of target
- 1000 randomized iterations per experiment
- Parameters were tuned experimentally ($\beta_k$ was insensitive)

Measurements were taken in terms of server error.
Fitness Evaluation for the Communication Goal

Summary of Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic 1</td>
<td>Mobile</td>
</tr>
<tr>
<td>Dynamic 2</td>
<td>Decay</td>
</tr>
<tr>
<td>Dynamic 3</td>
<td>Fast Decay</td>
</tr>
<tr>
<td>Dynamic 4</td>
<td>Mobile Fast Decay</td>
</tr>
<tr>
<td>Comm 0</td>
<td>Static</td>
</tr>
<tr>
<td>Comm 1</td>
<td>Mobile</td>
</tr>
<tr>
<td>Comm 2</td>
<td>Decay</td>
</tr>
<tr>
<td>Comm 3</td>
<td>Fast Decay</td>
</tr>
<tr>
<td>Comm 4</td>
<td>Mobile Fast Decay</td>
</tr>
</tbody>
</table>
## Fitness Evaluation for the Communication Goal

Dynamic Results in Euclidean Distance

<table>
<thead>
<tr>
<th>Experiment</th>
<th>C-dPSO</th>
<th>C-dPSO Decayed</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile</td>
<td>53.435</td>
<td>1.263</td>
<td>&lt; 0.00001</td>
</tr>
<tr>
<td>Decay</td>
<td>8.823</td>
<td>6.879</td>
<td>0.09863</td>
</tr>
<tr>
<td>Fast Decay</td>
<td>12.071</td>
<td>9.419</td>
<td>0.04702</td>
</tr>
<tr>
<td>Mobile Fast Decay</td>
<td>53.586</td>
<td>27.936</td>
<td>&lt; 0.00001</td>
</tr>
</tbody>
</table>
## Communication Results in Euclidean Distance

<table>
<thead>
<tr>
<th>Experiment</th>
<th>dPSO</th>
<th>C-dPSO</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>8.308</td>
<td>4.338</td>
<td>0.00008</td>
</tr>
<tr>
<td>Mobile</td>
<td>5.351</td>
<td>1.263</td>
<td>&lt; 0.00001</td>
</tr>
<tr>
<td>Decay</td>
<td>10.159</td>
<td>6.879</td>
<td>0.00499</td>
</tr>
<tr>
<td>Fast Decay</td>
<td>15.160</td>
<td>9.419</td>
<td>0.00005</td>
</tr>
<tr>
<td>Mobile Fast Decay</td>
<td>30.072</td>
<td>27.936</td>
<td>0.00309</td>
</tr>
</tbody>
</table>
C-dPSO Tracking a Mobile Target
Conclusion

*C-dPSO*

We proposed *C-dPSO* as an alternative to standard *dPSO*.

- Particles relay information to servers with more consistency.
- Fitness value decay helps to track mobile targets.

Future Work

- Examine problems with mobile servers.
- Investigate effects of a server with larger communication range.
- Implementation on physical robots.
Thank You!