Macro-Programming Support for Optimization in Wireless Sensor-Actuator Networks
Rahul Balani and Mani B. Srivastava
Networked and Embedded Systems Laboratory – http://nesl.ee.ucla.edu/

Introduction: Programming WSANs is hard, code is non-reusable across deployments

Optimization in WSANs
- Estimation and Optimal Control
  Find estimate of parameters or control inputs that optimize a cost function involving sensor data.

Variables that influence choice of algorithm and its implementation
- Deployment characteristics
  - Network topology
    - Communication modality
  - Application requirements
    - Latency, energy

Problem Description: Given opt. algorithm & variables, find most efficient implementation

Macroprogramming with MacroLab
- Matlab-like syntax and operations
- Macrovector: Data structure
  - Centralized, Distributed or Reflected storage
- Template-based decomposition
  - Matrix operations: find, addition, max
- Cost analysis: Messages, Latency
- But naïve decomposition results in inefficient implementation

Proposed Solution: Exploit properties of physical phenomena and prior research in distributed optimization to generate efficient implementations

Support for Convex Optimization
Macro-Operation: subgradient_descent()

Intelligent Light Control
\[
\tilde{I}^* = \arg \min_{I} \sum_{i=1}^{N} \left( S(i) + \phi - L_i \right) \\
\text{s.t. } S(i) = \sum_{j=1}^{M} A_{ij}, \forall i \in [1, N] \\
X = \{I_1, \ldots, I_N\} \mid 0 \leq I_j \leq I_{\text{max}}, \forall j \in [1, M] \}
\]

Exploiting physical properties of actuators and environment
- Effect of actuation is spatially limited
  - Light sources, sprinklers, etc.
- Fast incremental subgradient method
  - Approximate graph coloring

Prior research in distributed optimization
- Classical subgradient descent
  \[ \tilde{I}_{i,t} = P \left[ \tilde{I}_i - \alpha \sum_{j \neq i} S_j(\tilde{I}_j) \right] \]
- Incremental subgradient descent
  \[ \tilde{I}_{i,t} = P \left[ \tilde{I}_i - \alpha S_i(\tilde{I}_i) \right] \]
- Combination of consensus algorithms and subgradient descent
  \[ \tilde{I}_{i,t} = P \left[ \sum_{j \neq i} \left( \tilde{I}_j - \alpha S_j(\tilde{I}_j) \right) \right] \]

Fast consensus algorithm
\[
\tilde{I}_{i,t} = P \left[ \sum_{\nu \neq i} \left( \tilde{I}_\nu - \alpha S(\tilde{I}_\nu) \right) \right]
\]

<table>
<thead>
<tr>
<th>Network Config</th>
<th>Incremental subgradient</th>
<th>Consensus Algorithms</th>
<th>Fast Consensus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean Std</td>
<td>Mean Std</td>
<td>Mean Std</td>
</tr>
<tr>
<td>N=9, M=4</td>
<td>1099 376</td>
<td>9776 3311</td>
<td>489 164</td>
</tr>
<tr>
<td>N=25, M=9</td>
<td>3888 748</td>
<td>24150 4637</td>
<td>678 200</td>
</tr>
<tr>
<td>N=49, M=14</td>
<td>6316 1091</td>
<td>53624 5180</td>
<td>869 210</td>
</tr>
<tr>
<td>N=100, M=25</td>
<td>10920 1984</td>
<td>54800 5392</td>
<td>830 141</td>
</tr>
</tbody>
</table>