Pseudo-tree Construction Heuristics for DCOPs and Evaluations on the ns-2 Network Simulator

Atena M. Tabakhi
William Yeoh

Reza Tourani
Francisco Natividad
Satyajayant Misra
Outline

• Motivating Application
• Problem Definition
• Background (DCOPs)
• Pseudo-tree Construction Heuristics
• Empirical Evaluations
• Conclusions
Customer-driven Microgrids (CDMGs)

Motivating Application  Problem Definition  Background  Pseudo-tree Heuristics  Results  Conclusions

[S. Gupta, P. Jain, W. Yeoh, S. Ranade, E. Pontelli, DCR 2013]
Customer Driven Micro Grids (CDMG)
Challenges

• Identical communication times between all pairs of agents in typical Distributed Constraint Optimization Problem (DCOP) algorithms

• Unrealistic assumption in many real-world applications

Investigating the impact of variable communication times between pair of agents on the performance of DCOP algorithms
Distributed Constraint Optimization

\(<X, D, F, A, \alpha>:\)

- **X**: Set of variables.
- **D**: Set of finite domains for each variable.
- **F**: Set of constraints between variables.
- **A**: Set of agents, controlling the variables in X.
- **\(\alpha\)**: Mapping of variables to agents.

- **GOAL**: Find a maximum utility assignment.

\[ x^* = \arg \max_x F(x) \]

\[ = \arg \max_x \sum_{f \in F} f(x|_{\text{scope}(f)}) \]
Distributed Constraint Optimization

\(<\mathcal{X}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \alpha>:\)

- \(\mathcal{X}:\) Set of variables.
- \(\mathcal{D}:\) Set of finite domains for each variable.
- \(\mathcal{F}:\) Set of constraints between variables.
- \(\mathcal{A}:\) Set of agents, controlling the variables in \(\mathcal{X}\).
- \(\alpha:\) Mapping of variables to agents.

Constraint graph

<table>
<thead>
<tr>
<th></th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td>(\infty)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

Constraint (utility table)
DCOP: Assumptions

- Agents coordinate an assignment for their variables.
- Agents operate distributedly.

Communication:
- By exchanging messages.
- Restricted to agent’s local neighbors.
- Identical communication times

Knowledge:
- Restricted to agent’s sub-problem.
- Privacy preserving.

DCOP Constraint graph
NS-2: A Discrete Event Simulator

Generate Scenario

Initialize Event List

Fetch Next Event From Event List

Update Simulation Time base on Event Time

Process Event

Update Event List

Update Statistics

More Event?

Yes

No

Finish

Motivating Application

Problem Definition

Background

Pseudo-tree Heuristics

Results

Conclusions
DPOP Algorithm

- Distributed Pseudo-tree Optimization Procedure (DPOP)
  - Pseudo-tree Construction Phase
  - UTIL Propagation Phase
  - VALUE Propagation Phase

A DCOP Constraint Graph $\mathcal{G}$

A pseudo-tree of $\mathcal{G}$

[Petcu et al., IJCAI 2005]
DPOP Algorithm

- Distributed Pseudo-tree Optimization Procedure
  - Pseudo-tree Construction Phase
  - UTIL Propagation Phase
  - VALUE Propagation Phase

A DCOP Constraint Graph $G$:

A pseudo-tree of $G$:

[Petcu et al., IJCAI 2005]
DPOP Algorithm

- Distributed Pseudo-tree Optimization Procedure
  - Pseudo-tree Construction Phase
  - UTIL Propagation Phase
  - VALUE Propagation Phase

A pseudo-tree of $G$

A DCOP Constraint Graph $\mathcal{G}$

[Petcu et al., IJCAI 2005]
Variable Communication Times

Extend the DCOP model to $<X, D, F, A, C, \alpha>$, where, $C$ is a set of communication times $c_i \in C$ of a constraint $f_i \in F$

$$c_i = C' \cdot d_i$$

$d_i$ is a physical distance between agents sampled from $\mathcal{N}(\mu, \delta)$

$C' = 1$ millisecond per meter

Constraint graph
Definitions

- Generalized depth: Largest sum of $c_i$ across all constraints
- Simulated runtime: Time it takes for all agents to finish (assumes identical communication times between all pairs of agents)
- Actual runtime: Computed via ns-2 simulator to precisely measure variable communication times between agents

![Constraint graph](attachment:image1.png)

Generalized depth: 15
Measuring Actual Runtime

Variables’ x-y coordinates
Sampled from a distribution

Generate DCOP

Solve by DPOP
Algorithm

Generate
ns Scenario

Run
NS-2 Simulator

Output DPOP Actual runtime

Motivating Application  Problem Definition  Background  Pseudo-tree Heuristics  Results  Conclusions
Theoretical Results

- DPOP simulated runtime is exponential to the max width of its pseudo-tree
- DPOP actual runtime is exponential to the max width of its pseudo-tree and its depth
- Positive correlation between simulated and actual runtimes
- Positive correlation between both runtimes and pseudo-trees depth

<table>
<thead>
<tr>
<th>Constraint Density $p_1$</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation of “Actual” and Simulated Runtimes</td>
<td>0.74</td>
<td>0.84</td>
<td>0.97</td>
<td>0.91</td>
<td>0.94</td>
<td>0.95</td>
</tr>
<tr>
<td>Correlation of Depth and Simulated Runtime</td>
<td>0.99</td>
<td>0.97</td>
<td>0.82</td>
<td>0.66</td>
<td>0.69</td>
<td>0.67</td>
</tr>
<tr>
<td>Correlation of Depth and “Actual” Runtime</td>
<td>0.74</td>
<td>0.74</td>
<td>0.74</td>
<td>0.62</td>
<td>0.63</td>
<td>0.59</td>
</tr>
</tbody>
</table>
Pseudo-tree Construction Heuristics

The max-weighted-sum (mws) heuristic $h_{mws}$

A DCOP Constraint Graph $\mathcal{G}$

Generalized depth: 35
Pseudo-tree Construction Heuristics

The max-weighted-average (mwa) heuristic $h_{mwa}$

A DCOP Constraint Graph $G$

Generalized depth: 40
Pseudo-tree Construction Heuristics

The max-unweighted-sum (mus) heuristic $h_{\text{mus}}$

A DCOP Constraint Graph $\mathcal{G}$

Generalized depth: 25
The heuristic that used to select the root of the pseudo-tree can be different than the heuristic to select the non-root variables:

<table>
<thead>
<tr>
<th>Non-Root Variable</th>
<th>Root Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>h_{mws}</td>
<td>h_{mws}</td>
</tr>
<tr>
<td>h_{mwa}</td>
<td>h_{mwa}</td>
</tr>
<tr>
<td>h_{mus}</td>
<td>h_{mus}</td>
</tr>
</tbody>
</table>

Pseudo-tree Construction Heuristics
Random Graphs Setting

- Evaluate our nine heuristics against the max-degree heuristic on Random graphs
- Use the depths of pseudo-trees as the proxy for the runtimes of DPOP algorithm
- Averaged over 500 instances
- $|\mathcal{X}| = \{10, 20, 30, 40, 50, 60\}$ with $p_1 = 0.3$, $|\mathcal{D}| = 3$
- Sample $d_i$ from two truncated distributions Uniform and Gaussian $\mathcal{N}(50, 25)$
Random Graphs

Random graphs – Uniform Distribution

Random graphs – Gaussian Distribution
CDMGs Setting

• Sample neighborhood in U.S cities of Des Moines, Boston, San Francisco

• Randomly placed houses in 200m×200m grid

• Each house is constrained with its neighbors in 4 directions

• Greedily place aggregators with communication radius of 100m

• Each house is within the communication radius of at least 1 aggregator

• Averaged over 50 instances
Customer-driven Microgrids (CDMGs)

Motivating Application
Problem Definition
Background
Pseudo-tree Heuristics
Results
Conclusions
Conclusions

- Extended DCOP model to include communication times
- Incorporated communication times within the simulated runtime metric
- Measured communication times via ns-2 simulations to compute actual runtimes
- Proposed pseudo-tree construction heuristics
- Our heuristics exploit variable communication times
- Find pseudo-trees up to 20% shorter than the default tree
Thank You!