Impact of Problem Centralization on Distributed Constraint Optimization Algorithms

John P. Davin and Pragnesh Jay Modi
Carnegie Mellon University
School of Computer Science
{jdavin, pmodi}@cs.cmu.edu

Autonomous Agents and Multiagent Systems 2005

July 29, 2005
DCOP

- DCOP - Distributed Constraint Optimization Problem
  - Provides a model for many multi-agent optimization problems (scheduling, sensor nets, military planning).
  - More expressive than Distributed Constraint Satisfaction.
  - Computationally challenging (NP Complete).
Defining DCOP Centralization

- **Definition:** Centralization – aggregating information about the problem into a single agent, where:
  - this information was initially distributed among multiple agents, and
  - this aggregation results in a larger local search space.

For example, constraints on external variables can be centralized.
Motivation

- Two recent DCOP algorithms - Adopt and OptAPO:
  - Adopt does no centralization.
  - OptAPO does partial centralization.

- Prior work [Mailler & Lesser, AAMAS 2004] has shown that OptAPO completes in fewer cycles than Adopt for graph coloring problems at density $2n$ and $3n$.

- But, cycles do not capture performance differences caused by different levels of centralization.
Key Questions in this Talk

- How do we measure performance of DCOP algorithms that differ in their level of centralization?
- How do Adopt and OptAPO compare when we use such a measure?
Distributed Constraint Optimization Problems (DCOP)

- Agents: \( A = \{A_1, A_2 \ldots A_N\} \)
- Variables: \( V = \{x_1, x_2, \ldots x_n\} \)
- Domains: \( D = \{D_1, D_2, \ldots D_n\} \)
- Cost functions: \( f = \{f_1, \ldots f_k\} \)
- Objective function \( F: F = \sum_{x_i, x_j \in V} f_{ij}(d_i, d_j) \)

Goal: Find values for variables that minimize the sum cost over all cost functions (constraints).
DCOP Algorithms

• ADOPT [Modi et al., AIJ 2005]
  – Agents communicate *variable values* and *costs* (lower bounds, upper bounds).
  – Each agent’s search space remains constant.

  – *cooperative mediation* – a mediator agent collects constraints for a subset of the problem, applies centralized search.
  – Mediator’s search space grows.
Example

Adopt: \[ A = \{ A_1, A_2, A_3 \}, \quad D = \{ 0, 1 \}, \quad \text{Constraints} = \{ A_1 \neq A_2, \quad A_2 \neq A_3, \quad A_1 \neq A_3 \} \]
Example

OptAPO: $A = \{A_1, A_2, A_3\}$, $D = \{0, 1\}$, Constraints = $\{A_1 \neq A_2, A_2 \neq A_3, A_1 \neq A_3\}$
Key Questions

- How do we measure performance of DCOP algorithms that differ in their level of centralization?

- How do Adopt and OptAPO compare when we use such a measure?
Previous Metric: Cycles

- **Cycle** = one unit of algorithm progress in which all agents process incoming messages, perform computation, and send outgoing messages.
  - Independent of machine speed, network conditions, etc.
  - Used in prior work [Yokoo et al., 1998], [Mailler et al., 2004]
Previous Metric: Constraint Checks

- **Constraint check** = the act of evaluating a constraint between N variables.
  - Standard measure of computation used in centralized algorithms.

- **Concurrent constraint checks (CCC)** = maximum constraint checks from the agents during a cycle.
  - Used in prior work for distributed algorithms [Meisels et al., 2002].
Problems with Previous Metrics

• Cycles do not measure computational time (they don’t reflect the length of a cycle).
• Constraint checks alone do not measure communication.

We need a metric that measures both communication and computation time.

We introduce a new metric, Cycle- Based Runtime (CBR), to address this.
Cycle-Based Runtime

- $T =$ time of one constraint check.
  - Let $T = 1$, since constraint checks are the shortest operation that we are interested in.

- $L =$ communication latency (time to communicate in each cycle).

  - Let $L$ be defined in terms of $T$
    - $L = 10$ indicates communication is 10 times slower than a constraint check.

  - $L$ can be varied based on the communication medium.
    - Eg., $L = 1, 10, 100, 1000$. 

**CBR: Cycle-Based Runtime**

**total runtime of** \( m \text{ cycles} = CBR(m) = L \times m + ccc(m) \times t \)

Define \( ccc(m) \) as the total constraint checks:

\[
ccc(m) = \sum_{k=0}^{m} \max_{x_i \in V} cc(x_i, k)
\]
Results

- Tested on graph coloring problems, $|D|=3$ (3-coloring).
- Variables = 8, 12, 16, 20, with link density = 2n or 3n.
- 50 randomly generated problems for each size.

### Cycles:
- OptAPO takes fewer cycles, but more constraint checks.

### CCC:
How do Adopt and OptAPO compare using CBR?

$$CBR(m) = L \times m + ccc(m) \times t$$
How do Adopt and OptAPO compare using CBR?

\[ CBR(m) = L \times m + ccc(m) \times t \]

Density 3

For \( L \) values < 1000, Adopt has a lower CBR than OptAPO.

OptAPO’s high number of constraint checks outweigh its lower number of cycles.
How much centralization occurs in OptAPO?

- OptAPO sometimes centralizes all of the problem.
How does the distribution of computation differ?

- We measure the distribution of computation during a cycle as:

\[
\text{load} (k) = \frac{\max_{x_i \in \text{Agents}} cc(x_i, k)}{\sum_{x_i \in \text{Agents}} cc(x_i, k)}
\]

This is the ratio of the maximum computing agent to the total computation during a cycle.
- A value of 1.0 indicates one agent did all the computation.
- Lower values indicate more evenly distributed load.
How does the distribution of computation differ?

- Load was measured during the execution of one representative graph coloring problem with 8 variables, density 2:

- OptAPO has varying load, because one agent (the mediator) does all of the search within each cycle.
- Adopt’s load is evenly balanced.
Communication Tradeoffs of Centralization

- How do the algorithms perform under a range of communication latencies?

- Centralization performs best relative to non-centralized approaches at higher communication latencies.

- At high density, centralized Branch & Bound outperforms OptAPO.
Conclusions

• Cycle-Based Runtime (CBR), a performance metric which accounts for both communication + computation in DCOP algorithms.

• Comparison of Adopt + OptAPO showing Adopt has a lower CBR at several communication latencies.

• Future Work
  – Compare DCOP algorithms on a distributed system.