Privatizing Constraint Optimization

Rachel Greenstadt

AAAI/Sigart Doctoral Consortium

July 16, 2006
Problem: Privacy for DCOP

- Promise of DCOP
  - Coordination in optimal and automated ways

- Problems
  - Constraints may be personal and private
  - No evaluation of privacy in current systems
  - Current systems not designed for privacy
Approach

- Analyze existing DCOP algorithms
- Develop new metrics where appropriate
- Develop new algorithms with better privacy
- Analyze effect of continuous/dynamic DCOP on privacy
Example: Alice the Hairdresser

- Alice’s concerns
  - Doesn’t want clients to know how busy she is
  - Some clients preferred -- don’t want others to know

- Customer concerns (Bob and Carol)
  - Alice gossips and they don’t want their scheduling info spread around
Scheduling Domain: Entering the Mainstream

Microsoft Office Groove 2007

Beta is available

groove virtual office

Everything your team needs to share files, manage meetings and projects, track data and processes, and get work done as if you were all in the same location. Your team stays in sync. Your information and conversations stay secure.

Learn how Groove is used for
- IT Project Teams
- The Public Sector
- Managing a Small Business
- Consulting Practices

Try Groove

Current Groove Users
If you’re upgrading from an earlier version of Groove, Click here

Watch an Overview

Learn More About Groove Virtual Office

iCal

The desktop calendar, redefined.

Google Calendar

Yahoo! Calendar
Scheduling Process

Public Information: Meetings/appts, participants

Private Information: Preferences

Scheduler

Schedule of Meetings and Times
Centralized Models

- Central server optimizes subject to the constraints of individual preferences
- Privacy issues, unless trusted third party
- But are they trusted?
  - Your company’s IT department
  - Google/Microsoft (Big Brother?)
  - Data retention
DCOP Model

- Distributed Constraint Optimization Problem
- Distribute the problem for privacy/efficiency/autonomy/scalability
- Many algorithms
  - DPOP (Dynamic Programming Optimization)
  - SynchBB (Synchronous Branch and Bound)
  - Adopt (Asynchronous Distributed Optimization)
  - OptAPO (Optimal Asynchronous Partial Overlay)
- Metrics are important to distinguish privacy properties of these algorithms
Analysis of Existing Algorithms

- **Research Questions:**
  - Which algorithms are best in given situation?
  - Why do certain algorithms do better/worse?

- **Approach:**
  - Run experiments over many scheduling scenarios
  - Measure results with VPS metrics (AAMAS 05)

- **Results**
  - Distributed better than centralized
  - ADOPT & DPOP better than SynchID & SynchBB
  - Topology of agents has large impact on privacy
  - Asynchrony improves privacy
Existing Metrics: Valuations of Possible States (VPS)

- **Framework for quantitative metrics for privacy**
  - Assume agent A trying to infer private information about agent B
  - The relationship between $n$ and $p$ can be used to measure privacy

**Possible states**

- Before optimizing: $n = \{0,1,2,3,4,5\}$
- After optimizing: $p = \{0,3,4\}$
Can we do better than centralized?

• Yes!
• DPOP and ADOPT performed best.
• All were better than centralized!
Critique of VPS Metrics (1)

- No adversary/threat model
  - Who is the adversary?
  - What does the adversary do to gain information?
  - What if he does something unexpected?
Critique of VPS Metrics (2)

- Aggregation of partial information obscures actual losses
  - Metrics aggregate pairwise results
  - E.g. average privacy loss of all pairs of agents

- Example:
  - All agents lose half their data
  - Half agents lose all their data
  - Result is the same!
New Metric : D|A

- Consider only the Definitive harm D
  - Adversary gains concrete information w/ probability 1

- For a given Adversary A
  - Stronger adversaries might gain more information
Private Information

- For Alice, Bob, Carol
- How they value the meetings
- How they value their time
- 13 pieces total
- Corresponding to the 13 rows in the tables
How is information lost?

- Using DPOP algorithm
- Participants are organized into trees
- They send upward valuations for their subtrees
How is the Problem Solved?

- Bob builds a table with his utility for scheduling the meeting (AB) in each timeslot
- Bob’s valuation is \( V(AB) - V(\text{timeslot}) \)
- Row 2 is \( V(AB) - T_0 \) or \( 4 - 1 = 3 \)
- He sends the table to Alice and she optimizes
- We count as lost any data which the adversary determines with probability 1 (only one state remaining)
Results

- Bob and Carol lose all their information to Alice
- 8 pieces of personal information
- Out of 13 total pieces (Alice’s 5 valuations are not revealed)
- So privacy loss is 8/13
New Example: Chain Topology

- Only Carol loses her information
- Privacy loss is 4/13
Metric in VPS Framework

- This function sums up all pieces of personal information known by any other participant \( j \) about participant \( i \).

- We then add up the results for each participant and divide by the total amount of personal data.

\[
V_i(P_i(S_i)) = \frac{\sum_{s_i} |s_i| \sum_{j \neq i} \sum_{t \in S(t_j)} I_{P_i(t) = 1}}{\sum_{i} |s_i|}
\]
A for Adversary:

But we still don’t have a threat model

- Adversary could take many actions
  1. View messages sent to a single participant in the course of algorithm
  2. Run “StalkerPro” to do sophisticated inference
  3. Use outside domain knowledge
  4. Collude with other participants
  5. Actively manipulate the message stream

- Each action has a cost
If we can linearize this cost in risk/resources, we can get at the two dimensional plane

Linearization can be hard to do

But, it is important to do so to resolve the harm of partial information
D|A Conclusions

- Previous metrics don’t capture intuitive notions of privacy loss
- Or contextualize it in a threat model
- Proposed D|A metric
  - Good for broad notions of privacy
  - Helping us design new algorithms
  - Still needs work on A side
Current and Future work: New algorithms

- Approach:
  - Use analysis to identify key features of algorithms for privacy
    - Topology
    - Asynchrony
  - Design algorithms around these features
  - Import ideas from anonymity and trust management literature
Current and Future Work: Dynamic DCOP

- Scheduling is inherently dynamic
  - New events constantly arise
- Expectation: more privacy loss
- Research Goals
  - quantify privacy loss in dynamic DCOP
  - evaluate privacy impact of different approaches
Goals of Dissertation

- Develop techniques to evaluate privacy in DCOP
- Understand how well existing algorithms protect privacy
- Quantify design tradeoffs between privacy, efficiency and optimality
- Design algorithms with more privacy than status quo
Example Results

Comparison Results
- Averaged over 25 runs
- Results for other scenarios were similar
- Privacy loss between all pairs of agents is averaged
- Privacy loss considered per timeslot by EntropyTS metric (logarithmic scale), and scaled between 0 (no privacy loss) and 1 (total loss)

See poster, DCR paper, or AAAI paper for additional results
Example: Alice’s Schedule

Alice’s Schedule

08:00 - 08:30: Pay traffic ticket
09:00 - 11:00: Workgroup meeting
12:00 - 13:30: Lunch with clients
14:00 - 14:30: Call lawyer about divorce settlement
15:00 - 17:30: Job interview
17:30 - 18:00: Pick up kids from after school program
19:00 - 19:30: Pick up babysitter
20:00 - 00:00: Hot date!

Alice has a busy schedule and would like to optimize it but she doesn’t want all parties to know about all her time conflicts.
D|A in DPOP

- Leaf nodes lose all their information
  - 9/25 meeting valuations
  - 35/70 timeslot valuations
  - 45% privacy loss
- No single node can determine the internal nodes’ valuations for certain
Trees with greater depth and less breadth produce more privacy and less efficiency

Nodes near the bottom of the tree lost more privacy than nodes at the top.
Asynchrony improves privacy iff message origin is unknown

Use anonymity techniques to hide message origins better

– Without sacrificing too much efficiency?