FROM ARTIFICIAL INTELLIGENCE TO GAME THEORY

Boi Faltings
Artificial Intelligence
Artificial Intelligence

• People are good in the natural world:
  • irregular shapes
  • reactive
• Modern world is artificial:
  • precision machinery
  • schedules and interdependencies
• AI: helps people work in the artificial world
My Motivation

• Use computers to solve a problem that people could not solve otherwise.
Early Days: mechanisms

- People find it hard to predict interactions of shapes.
- Even harder to design them.
- Can computers help?
Success, but...

- Computers can invent devices:

- Market for such ideas is very small!
Lesson 1

- Work on a significant problem.
Sharing Networks

- Routing demands in circuit-switched communication networks.
- Shortest path in blocking island graph rather than network => 40% more capacity.
**Iconomic Systems V1**

- **Goal:** market Icononet = routing tool that increases capacity by 40%
- **Problem:** nobody wanted to save capacity.

Monday, November 15, 2010
Lesson 2

• Work on a marketable problem.
Smart Clients

• Internet -> Java -> Applets
• Idea: offer intelligent functionality as applet + data
• Example: travel planning
Sophisticated Search

1. Drop-down menus allow fine tuning of preferences.
2. Preferences are stored and can be reused at any time.

Monday, November 15, 2010
Architecture

- Constraint Programming: simple search and propagation algorithms implement intelligent behavior.
- Developed OpenSource package (Java Constraint Library).
- Java Program + Interface can be packaged into small applet and run on user’s PC.
**SmartClient Advantage**

- Applet size = 4 web pages.
- With application: 10KB data suffices.
Java Wars

- Battle Sun vs. Microsoft: Microsoft removed Java from IE.
- Most companies did not allow Java installation on corporate PCs.
- This killed a lot of the potential. (Fortunately, not all).
Lesson 3

• Be a good sport.
A bigger Challenge

- Is it possible to make a distributed organization as efficient as a centralized one?

- Examples:
  - airport slot allocation
  - sharing truck capacity
  - project planning
Airport Slot Allocation

- Planes need to take off and land.
- => intended schedule only works if all slots are granted.
- otherwise airlines find another use...
- => huge inefficiencies.
Sharing Truck Capacity

• Most trucks drive home empty.

• Sharing capacity among operators could eliminate a lot of truck traffic and cost.

• So can sharing orders among competing companies.
Project Planning

• Construction project involves 50-100 contractors.

• All need to be scheduled.

• All plan independently and are self-interested.
Issues in Multi-Agent Coordination

- Efficient multi-agent optimization.
- Rationality: incentives to cooperate?
- Privacy: not revealing competitive information.
Constraint Optimization

- n discrete variables $x_1..x_n$ in $d_1..d_n$
- m constraints: $c_i = \{x_j, x_k, ..\}$
- M relations $r_i : c_i \rightarrow R$
- find $x_1 = v_1$, .., $x_n = v_n$ to minimize $\sum r_i$
- most general method: search
- complexity $O(d^n)$
Multi-Agent Optimization

• Information about constraints and preferences distributed among agents.
• Centralize all information:
  • insufficiently formalized.
  • lack of control.
  • privacy concerns.
• => distributed algorithms.
Progress on Distributed Constraint Optimization

Meeting Scheduling: problem size solvable with complete algorithm

Sensor Configuration

- Sensors track moving targets in their vicinity.
- Neighbouring sensors need to coordinate who tracks what target.
- DARPA benchmark problem.
Sensor Configuration

- Grid-like sensor layout
- Width = 3
- Complexity of solving: $O(d^3)$
- not $O(d^n)$
DFS orderings

- Tree and back edges.
- Back edges to ancestors.
- Separator $S(x)$ isolates subtree below $x$.
- Cost of subtree = message with one dimension for each variable in $S(x)$, size $d \| S(x) \|$
Minimizing Information Exchange

- DFS ordering -> separators.
- Complete information: $d^{|S|}$ values.
- Bounds, heuristics -> focus on a subset.
- How small can we make this?
Open Constraint Satisfaction

• What if it is consistent to choose the highest utility values in every relation?
• => would not need to search!
• Exploit this by searching in best-first order.
• Domains can be unbounded: open constraint optimization.
• Reduce information exchange dramatically.

• With good engineering, can probably solve many interesting applications.
Self-interest

- Agents are self-interested and could manipulate the outcome.
- Truthfulness: agents truthfully report their preferences.
- Faithfulness: agents correctly execute the protocol.
Mechanism Design

• Mechanism implements a function f(s):
  • self-interested agents behave so that outcome is given by f(s)
  • Commonly: agents declare types t(i), algorithm computes o(t), payments p(t).
• Well-studied in game theory.
  (Economics Nobel prize 2007)
Examples

• Implementable: affine maximizers, median of types, exchanging houses.
• Not implementable: Nash bargaining solution, average of types.
• Restricting inputs creates opportunities.
Truthful Mechanisms

- Best strategy for self-interested agents: declare types truthfully.
- Revelation principle: always exists a truthful version.
- Constraint optimization: counter manipulation by charging a (VCG) tax.
- Analogous to auctions.
Faithfulness

- Truthfulness \neq Faithfulness
- However, VCG tax -> incentive-compatibility -> faithfulness.
- See MDPOP (2008) for efficient implementation.
Examples

• Auctions: determine the winner while keeping the bids private.

• Airport slot allocation: assign best combination of slots without revealing airlines’ marketing plans.

• Meeting scheduling: schedule meetings without revealing participants’ preferences.
**Homomorphic Computation**

- Classical architecture: trusted server

  - Idea: compute with encrypted inputs
  - Only result is revealed.

\[
\begin{align*}
x & \xrightarrow{e(x)} e(f(x,y,z)) \\
y & \xrightarrow{e(y)} e(z) \\
z & \xrightarrow{e(z)}
\end{align*}
\]
Constraint Satisfaction Through homomorphic Encryption

- For every possible solution $s_k$:
  - $A_i$ computes number of constraint violations $p_i(s_k)$ and encrypts it.
  - All $p_i$ are summed up and decrypted.
  - If $\text{sum}(p_i) = 0$, $s_k$ is a solution.
- Search/dynamic programming more efficient, but requires minimization.
A CLOSER LOOK AT PRIVACY

- Privacy of preferences and constraints is not the only thing.
- Other aspects of privacy:
  - agents: who else is participating?
  - topology: who has constraints with whom?
  - decision: what are the decisions?
Privacy and SMC

- SMC guarantees constraint privacy, but not agent, topology or decision privacy!
- These can be achieved through distributed computation ([Leaute & Faltings, 2009]):
  - only know about others if there is a constraint; hide identities through codenames.
  - every agent computes its own decisions.
Solved...

- Optimization without central authority for hundreds of variables.
- Payment-based schemes for faithfulness.
- Maintaining privacy (including agent and constraint privacy)
Open Issues

• Complete optimization algorithms with low information exchange.
• Faithful optimization without payments.
• Efficient optimization while maintaining privacy.
Multi-Agent Learning

• Optimal coordination => communication complexity too high.
• What can be done with very little coordination?
• Example: *learn* to coordinate on a single joint signal.
Automated Mechanism Design

- Choose desired function $f(t)$.
- Mechanism design = payments $p(t)$.
- Constraint satisfaction with $p(t)$ as variables, $f$ as constraint.
- Space $t$ where $f$ is implementable = $t$ where AMD has a solution.
Incentivize truthful information revelation

- Judge by comparing reports.
- Variables = payments depending on combinations of reports.
- Constraints: truthful reports have highest payoffs, payoffs > 0, etc.
- Found first mechanisms where truthful reporting is the best equilibrium.
CS and Society

- What will the future economy look like?
  - Centralized planning?
  - Distributed planning?
  - Markets?
  - Mechanisms?
- Essentially, this is an issue of coordination - Computer Science can have a huge impact on society.
Acknowledgements

- Kefeng Hua
- Kun Sun
- Ian Smith
- Berthe Choueiry
- Jamila Sam
- Beat Liver
- Monika Lundell
- Eric Sauthier
- Rainer Weigel
- Esther Gelle
- Claudio Lottaz
- Christian Frei
- Steven Willmott
- Monique Calisti
- Omar Belakhdar
- Pedro Bados
- Marius Silaghi
- Marc Torrens
- Michael Schumacher
- Walter Binder
- Carlos Eisenberg
- Nicoleta Neagu
- Santiago Macho-Gonzalez
- Ion Constantinescu
- Paolo Viappiani
- Vincent Schickel-Zuber
- Adrian Petcu
- Radu Jurca
- Quang-Huy Nguyen
- Radoslaw Szymanek
- Thomas Léauté
- Brammert Ottens
- Florent Garcin
- Li Pu
- Ludek Cigler
- Immanuel Trummer