Multi-Agents systems for Distributed Optimization

Meritxell Vinyals [1]

[1] IIIA-CSIC

JMAS, Sevilla, 14/06/10

Outline

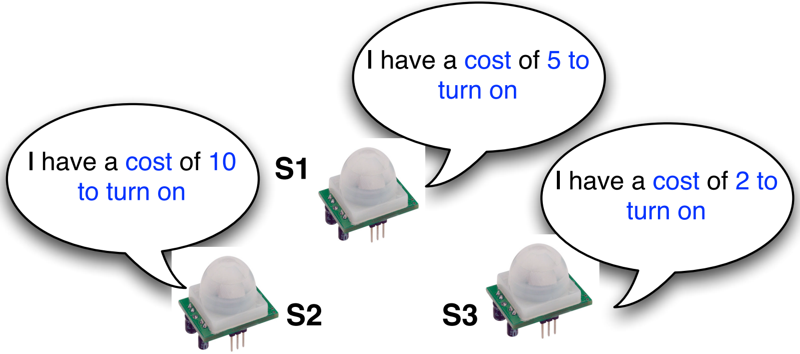
DCOPs by example: sleep/sense cycle problem

Distributed Constraint Optimization Problem (DCOP) helps modeling cooperative multi-agent decision making.

- Distributed bounded resources (sensors)
- Discrete sets of possible actions (ON/OFF)

DCOPs by example: sleep/sense cycle problem

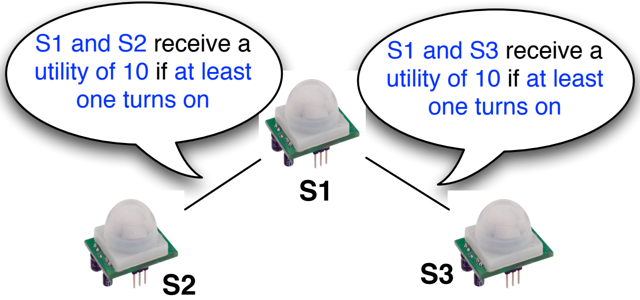
Sensors may have their own preferences:



Sensors may report a cost for sampling (e.g. which may vary depending on the remaining battery)

DCOPs by example: sleep/sense cycle problem

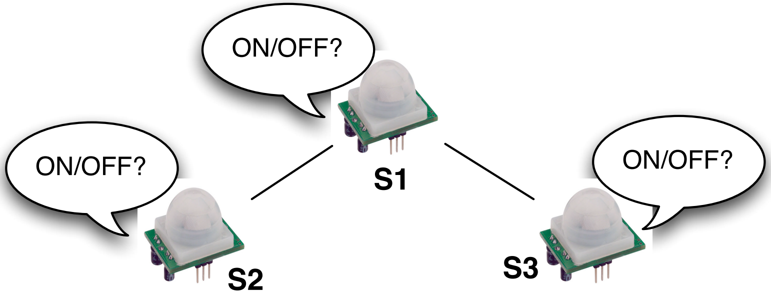
Locality of interactions (physical neighborhood)



There is a utility if a region is sampled by at least one sensor

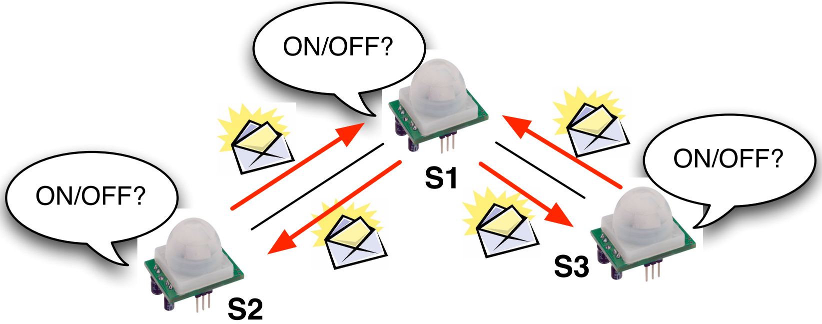
DCOPs by example: sleep/sense cycle problem

(Distributed Collaborative systems) The goal is to **distributedly** find the set of actions that **maximize the global social reward**



DCOPs by example: sleep/sense cycle problem

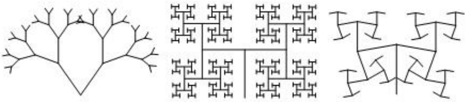
Sensors must solve the problem by exchanging messages with their closer neighbours about their local information



How hard is to solve a DCOP?

DCOPs are in general NP-Hard problems. That means that with few exceptions, namely:

Tree-like networks **Binary submodular preferences**



In general agents will need an exponential amount of communication or computation to find the optimal set of actions for the whole network.

Outline

Planning: Scheduling elective surgery

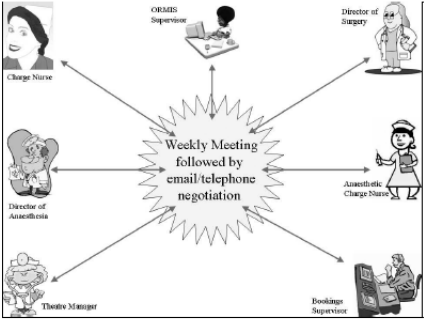
[Khanna et al., 2009]

Case of study: Princess Alexandra Hospital (Australia)

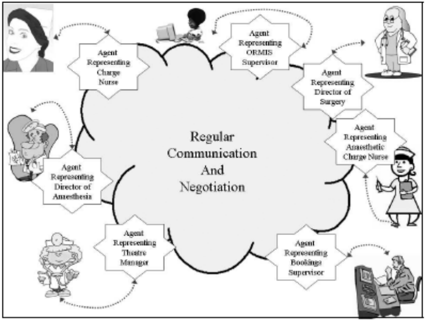
- Multiple departments, 21 operating theaters divided into slots of 3.5 hours each
- Each department books patients in consultation with the surgical team in a common system.
- Once a week the managers of the different departments meet and review bookings (conflicts worked out by negotiation)
- Further changes being often required (unexpected emergencies, variation in patients' health state, staff availability)

Planning: Scheduling elective surgery

[Khanna et al., 2009]



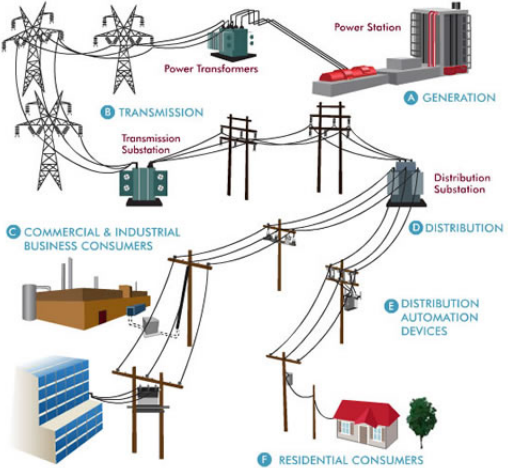
(a) Current Model



(b) Proposed Model

Outline

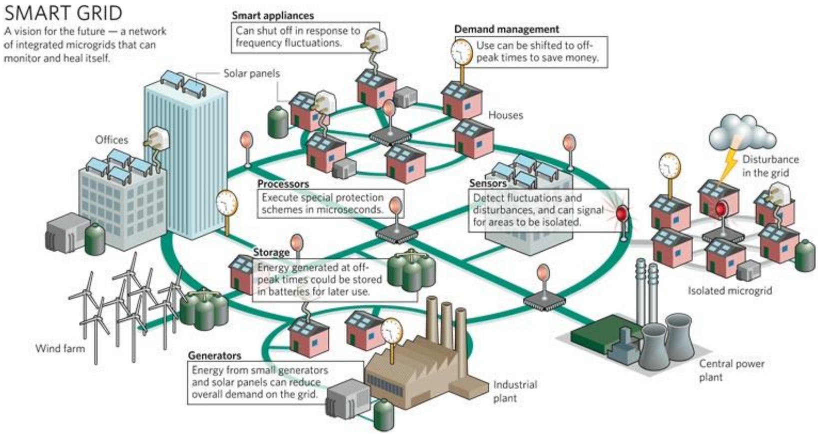
The electricity grid today



www.oncor.com/images/content/pathwaytopower.gif

- The current hierarchical, centrally-constrolled grid is obsolete.
- Problems on scalability, efficiency and integration of green energies
- Most of the decisions about the operation of a power system are made in a centralized fashion

Smart grid: the electricity grid of tomorrow



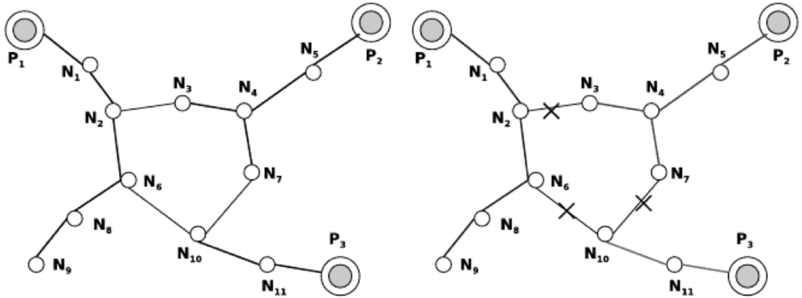
http://www.nature.com/news/2008/080730/images/454570a-6.jpg

Vision for smart grid 2030

- Centralized control is replaced with **decentralized control**:
  - **efficiency and scalability**
  - **complex control mechanisms** needed
- Introduction of intelligence at all levels, especially at lower levels, to **provide timely and accurate control responses.**
- The use of intelligent agents:
  - allow to respond to problems faster than a human operator.
  - **distributedly adapt to events and environments, and act** competitively or **cooperatively for the good of the entire system**.

DCOPs for smart grid: Configuration of power networks

[Petcu & Faltings, 2008]



(a) Simple network, with 3 power sources, 14 buses and 11 sinks.

(b) Optimal Configuration

How sinks configure the network by enabling transmission lines such that is (a) cycle-free and (b) the amount of power lost is minimized.

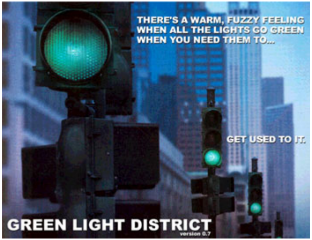
Outline

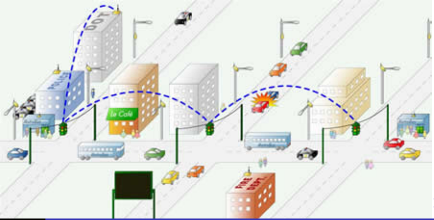
1. Introduction
2. Applications
   - Scheduling
   - Smart grid
   - Traffic light synchronization
3. Open problems
   - Trade-off solution quality/cost
   - Dynamic environments
   - Scalability

Traffic light synchronization

Early traffic lights

Future traffic light generation





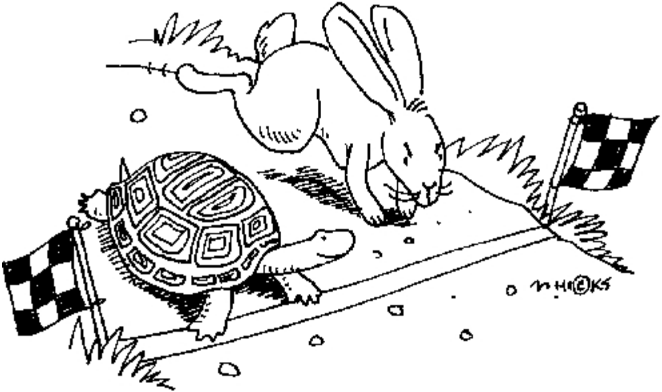
Traffic light synchronization

[R. Junges and A. L. C. Bazzan, 2010]

- Distributedly coordinate traffic lights so that vehicles can traverse an arterial in one traffic direction, keeping a specific speed without stopping (green waves)
- Each traffic light agent coordinates with its neighbours to select its best signal plan considering:
  - The current traffic light situation: The amount of traffic at intersection *j* that is coming from direction *i*
  - The sychronitzation and the coping with the volume of vehicles with their neighbours

Maxims for researchers

First takes all credit, second gets nothing



Maxims for researchers

Either you are the first or you are the best in the crowd

The moral

Identify open problems, preferably with few contributions

The most wanted list in distributed optimization

- Trade-off solution quality/cost
- Dynamic environments
- Scalability

Outline

1. Introduction
2. Applications
   - Scheduling
   - Smart grid
   - Traffic light synchronization
3. Open problems
   - Trade-off solution quality/cost
   - Dynamic environments
   - Scalability

Trade-off solution quality/cost

Bounded racionality: search for the best solution given the bounded resources (time, computation, communication).

Optimality: the idealistic (but usually impractical) term

Researchers proposed **optimal algorithms** to solve DCOPs **trying to minimize the communication and computation** needed by agents:

DPOP [A. Petcu & B. Faltings, 2005] **Adopt** [Modi et al., 2005]
OptAPO [R. Mailler & V. Lesser, 2004] **Action-GDL** [M. Vinyals et al., 2010]

However . . .

- All of them have an **exponential cost** (either in size/number of messages or computation)
- Very often in real problems the price of **optimality** is simply **unafforable**

Incomplete algorithms: low-cost at no guarantees

Researchers also proposed **low-cost incomplete algorithms**:

DSA [Yokoo & Hirayama, 1996] DBA [Fitzpatrick & Meeterns, 2003]

- Can return a solution at anytime
- Small amount of communication/computation per agent
- Usually good solutions in average

But not guarantee . . .

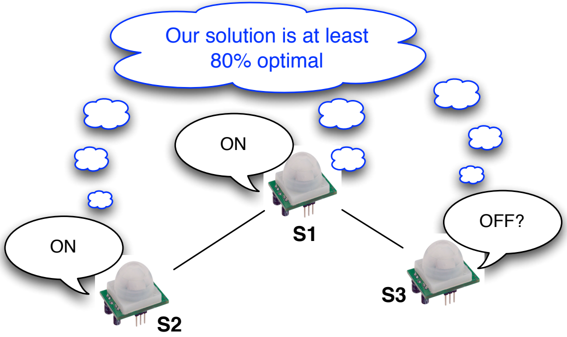
DSA: Distributed Stochastic Algorithm

Algorithm 2 Sketch of DSA, executed by an agent.

```
Randomly choose a value
while (no termination condition is met) do
    if (a new value is assigned) then
        send the new value to neighbors
    end if
    collect neighbors' values, if any; compute the best possible
    conflict reduction Δ
    if (Δ > 0) or (Δ = 0 but there is a conflict) then
        change to a value giving Δ with probability p
    end if
end while
```

What we understand for guarantees?

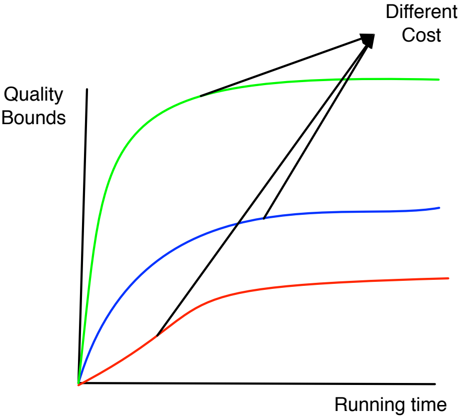
. Bounds the maximum error of the solution with respect to the optimal



Quality guarantees:

- Online guarantees
- Offline guarantees

Online guarantees



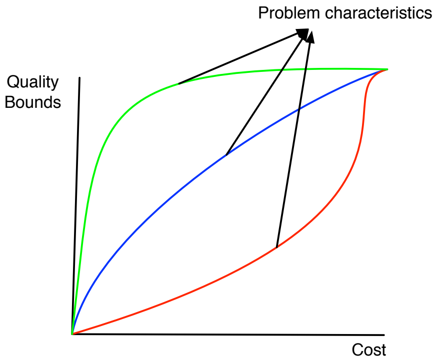
- Per-instance specific bounds
- Are known on runtime
- Vary along the execution of the algorithm

ADOPT [Modi et al., 2005],
Bounded Max-Sum [A. Farinelli et al., 2009],
DaC algorithms [M. Vinyals et al., 2010]

Allow agents to be aware of the goodness of their decisions, it is a criteria to trade-off quality vs cost.

Offline guarantees

Allow to assess the bound for the solution on convergence considering problem properties before running the algorithm



- Characterize problems that are more favorable for the algorithm
- If the convergence is guaranteed in a number of steps => it quantifies the cost of obtaining β-solutions

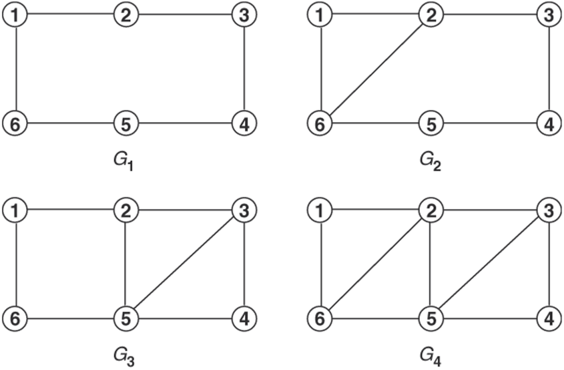
K-size optimal algorithms, [J.P. Pearce & M. Tambe, 2007]

T-size optimal algorithms, [C. Kiekintveld et al, 2010]

Outline

Dynamic environments

Agents' decision variables and their dependencies change/are added/are deleleted over time



Solution:

Self-stabilization [Z. Collin et al., 1999]: is the ability of a system responding to transient failures, eventually reaching a legal state and maintaining it afterwards.

A self-stabilizing algorithm is able to cope with dynamic environments.

Beyond self-stabilization: optimize the cost of reconfiguration

Small changes require small efforts to adapt.

- Optimize the amount of communication and computation to recover from a change in the environment
- Not start from scratch (to be completely re-run after every change in the environment), take advantatge of the work done.

S-DPOP [A.Petcu & B.Faltings, 2005],
Bounded Fast Max Sum [K Macarthur et al, 2010], . . .

Beyond self-stabilization: super-stabilization

- In **super-stabilizing systems** there are some conditions that are always satisfied even during the reconfiguration:
  - The previous assignment of variables is maintained until a new solution has been computed

S-DPOP [A.Petcu & B.Faltings, 2005],
Bounded Fast Max Sum [K Macarthur et al, 2010], . . .

Outline

Scalability

- Many real-world domains require multi-agent systems to **operate on large systems** (with a large number of agents)
- Although low-cost algorithms **scale to large systems in terms of cost**, often they do **not scale in terms of solution quality** (even in near-decomposable systems)

One promising solution: Organizations

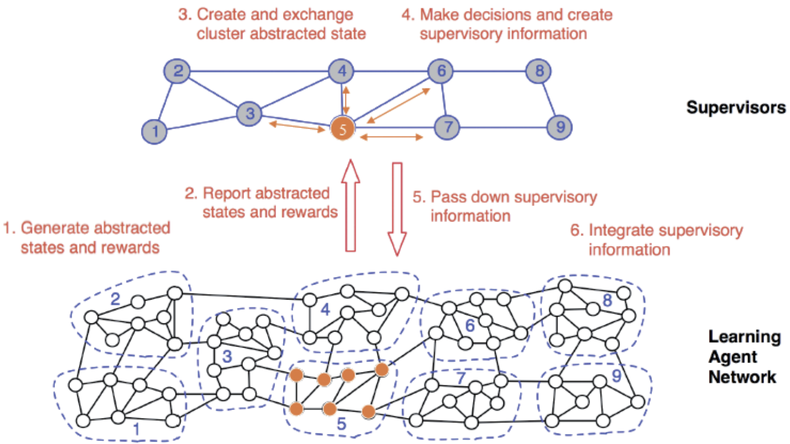
What do organitzations offer?

Flexibility and adaptability of control at all levels Multiple levels of reasoning among decisions of multiple agents:

- Lower levels: more detailed information of the problem (local view)
- Higher levels: broader view but coarser (abstract view)

Agents in coarser levels take into account global dependencies that are not considered by agents at lower levels => scalability

Achieving scalability: organizations



More about MAS?

- **A concise introduction to multiagent systems and distributed artificial intelligence**. N. Vlassis.
- **Multiagent Systems Algorithmic, Game-Theoretic, and Logical Foundations**. Yoav Shoham and Kevin Leyton-Brown. Available at `http://www.masfoundations.org/download.html`
- Fundamentals of Multiagent Systems with NetLogo Examples. Jos Maria Vidal. `http://multiagent.com/`

More about a PhD life?

- **How to do Research At the MIT AI Lab**. *By a whole bunch of current, former, and honorary MIT AI Lab graduate students.* `http://ee.tongji.edu.cn/pages/forum/root/knowledgebase/howtodoresearch/mit.research.how.to.html`
- **How to Be a Good Graduate Student**. *Marie des Jardins.* `http://www.cs.indiana.edu/how.2b/how.2b.html`
- **`http://www.phdcomics.com/`**
- **To know the road ahead, ask those coming back**. *Chinese proverb*

Muchas gracias por vuestra atención

References I

- [Petcu & Faltings, 2008]. **Distributed Constraint Optimization Applications in Power Networks**. International Journal of Innovations in Energy Systems and Power, Vol. 3, no. 1 (April 2008)
- [Khanna et al, 2009]. **An Efficient Algorithm for Solving Dynamic Complex DCOP Problems**. WI-IAT '09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. Pages 339-346.
- [R. Junges & A. L. C. Bazzan, 2010]. **Evaluating the performance of DCOP algorithms in a real world, dynamic problem**. In AAMAS, pages 599Ð606, 2008.
- [Modi et al., 2005]. **ADOPT: Asynchronous distributed constraint optimization with quality guarantees**. In Artificial Intelligence Journal (161) 149Ð180
- [R. Mailler & V. Lesser, 2004]. **Solving distributed constraint optimization problems using cooperative mediation**. AAMAS (2004) 438-445.
- [A. Petcu & B. Faltings, 2005]. **DPOP: A scalable method for multiagent constraint optimization**. IJCAI (2005) 266-271
- [M. Vinyals et al., 2010] **Constructing a unifying theory of dynamic programming DCOP algorithms via the Generalized Distributive Law**, Journal of Autonomous Agents and Multi Agent Systems (JAAMAS).

References II

- [Yokoo & Hirayama, 1996]. **Distributed breakout algorithm for solving distributed constraint satisfaction and optimization problems**. In ICMAS.
- [Fitzpatrick & Meeterns, 2003]. **Distributed Coordination through Anarchic Optimization**. In: Distributed Sensor Networks A multiagent perspective. Kluwer Academic(2003) 257-293.
- [A. Farinelli et al., 2009]. Bounded Approximate Decentralised Coordination using the Max-Sum Algorithm. In: IJCAI-09 Workshop on Distributed Constraint Reasoning (DCR), 13th July 2009. pp. 46-59.
- [M. Vinyals et al., 2010]. Divide and Coordinate: solving DCOPs by agreement.AAMAS(2010). pages 149-156.
- [C. Kiekintveld et al, 2010]. Asynchronous Algorithms for Approximate Distributed Constraint Optimization with Quality Bounds. AAMAS(2010). pages 133-140.
- [J.P. Pearce & M. Tambe, 2007]. Quality Guarantees on k-Optimal Solutions for Distributed Constraint Optimization Problem. IJCAI, 2007. pages 1446-1451.
- [Z. Collin et al., 1999]. Self-stabilizing distributed constraint satisfaction. Chicago Journal of Theoretical Computer Science.
- [A.Petcu & B.Faltings, 2005] S-DPOP: Superstabilizing, fault-containing multi-agent combinatorial optimization. AAAI-05, pages 449Ð454.
- [K Macarthur et al, 2010] Efficient, Superstabilizing Decentralised Optimisation for Dynamic Task Allocation Environments. OPTMAS@AAMAS(2010).