Multi-Agents systems for Distributed Optimization

Meritxell Vinyals $^1$

$^1$IIIA-CSIC

JMAS, Sevilla, 14/06/10
Outline

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

- I have a cost of 5 to turn on
- I have a cost of 10 to turn on
- I have a cost of 2 to turn on

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)

S1 and S2 receive a utility of 10 if at least one turns on

S1 and S3 receive a utility of 10 if at least one turns on

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

1 Introduction

2 Applications
   - Scheduling
   - Smart grid
   - Traffic light synchronization

3 Open problems
   - Trade-off solution quality/cost
   - Dynamic environments
   - Scalability
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]
Outline

1. Introduction

2. Applications
   - Scheduling
   - Smart grid
   - Traffic light synchronization

3. Open problems
   - Trade-off solution quality/cost
   - Dynamic environments
   - Scalability
The electricity grid today

- The current hierarchical, centrally-controlled 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

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.
(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 synchronization and the coping with the volume of vehicles with their neighbours
Maxims for researchers

*First takes all credit, second gets nothing*
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]
- **OptAPO** [R. Mailler & V. Lesser, 2004]
- **Adopt** [Modi et al., 2005]
- **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 unaffordable
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 . . .
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 $\Delta$
    if ($\Delta > 0$) or ($\Delta = 0$ but there is a conflict) then
        change to a value giving $\Delta$ with probability $p$
    end if
end while
What we understand for guarantees?

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

Our solution is at least 80% 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 $\beta$-solutions

K-size optimal algorithms, [J.P. Pearce & M. Tambe, 2007]
T-size optimal algorithms, [C. Kiekintveld et al, 2010]
Outline

1 Introduction

2 Applications
   • Scheduling
   • Smart grid
   • Traffic light synchronization

3 Open problems
   • Trade-off solution quality/cost
   • Dynamic environments
   • Scalability
Dynamic environments

Agents’ decision variables and their dependencies change/are added/are deleted over time
Solution:

Self-stabilization \cite{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 advantage of the work done.

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

1. Introduction

2. Applications
   - Scheduling
   - Smart grid
   - Traffic light synchronization

3. Open problems
   - Trade-off solution quality/cost
   - Dynamic environments
   - 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 organizations 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

1. Generate abstracted states and rewards
2. Report abstracted states and rewards
3. Create and exchange cluster abstracted state
4. Make decisions and create supervisory information
5. Pass down supervisory information
6. Integrate supervisory information

Supervisors

Learning Agent Network
More about MAS?

- **A concise introduction to multiagent systems and distributed artificial intelligence**. N. Vlassis.
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.  

- **How to Be a Good Graduate Student.** Marie des Jardins.  
  [http://www.cs.indiana.edu/how.2b/how.2b.html](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


[Yokoo & Hirayama, 1996]. Distributed breakout algorithm for solving distributed constraint satisfaction and optimization problems. In ICMAS.


