Privacy-preserving and socially optimal resource allocation

FP7 Project INSIGHT (`insight-ict.eu`)

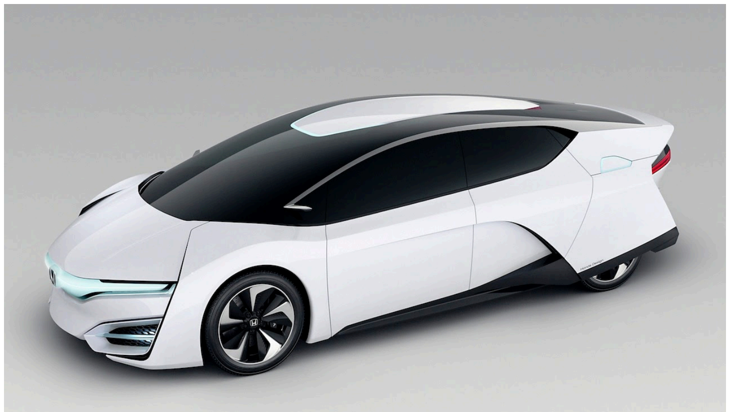
Jia Yuan Yu
with F. Wirth, S. Stuedli, M. Corless, and R. Shorten

IBM Research Ireland

October 20, 2014, IC1205 Computational Social Choice

Smarter Cities Technology Centre

Motivation: Internet of Things



What "Things?"

- smart meters,
- smart appliances (air conditioning),
- self-driving cars,

Why Internet? Real-time communication between Things.

Why communication between Things?

Coordinate agents and avoid congestion:

- everyone turning on AC at the same time,
- everyone driving at the same time,
- everyone downloading at the same time.
- everyone charging their electric car at the same time.

This boils down to a resource allocation problem.

What do the operator and agents want?

IBM

To allocate optimally, a central operator needs:

- know the temperature in your house, and what temperature you would like;
- to tell each agent what to do (e.g., controlling your AC).

What do agents want?

- Privacy: I don't want a central operator or my neighbours to know how badly I need to turn on my AC!

Can we make everyone happy?

IBM

Is it possible to satisfy these objectives?

- Allocate optimally,
- Keep privacy.

Yes, and more:

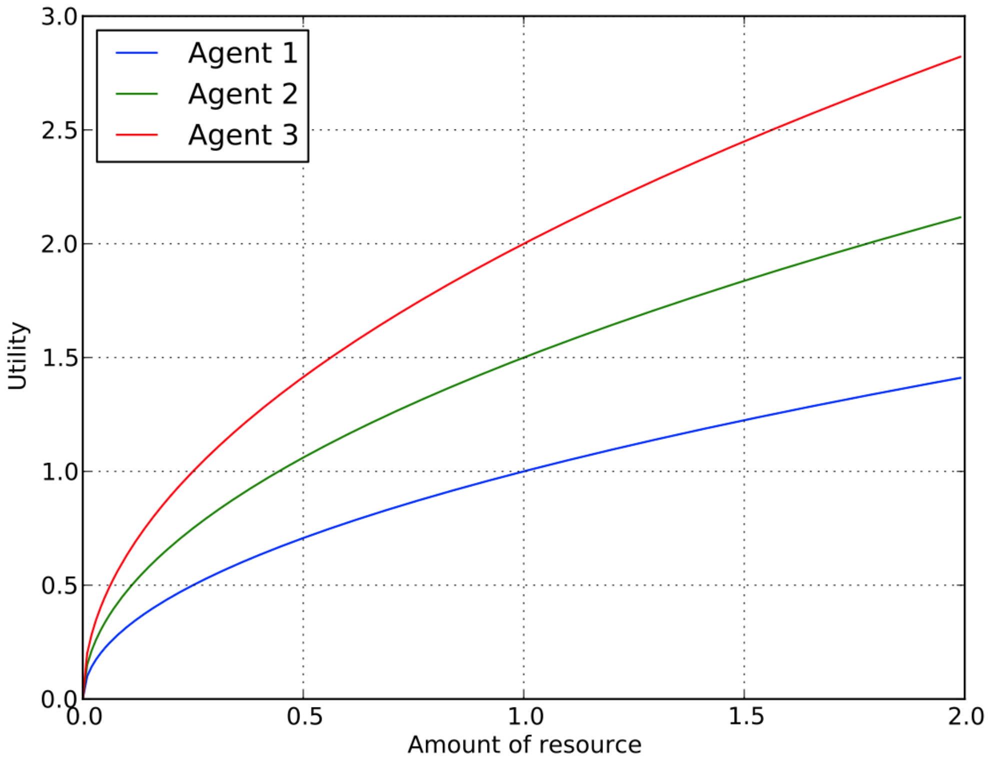
- Agents do not send anything.
- Agents do not receive any signal from other agents.
- Central operator sends (broadcasts) the same signal to everyone.

What's the catch?

Agents have to behave nicely:

- Follow a predefined policy,
- Listen to the signal from the central operator.

Mathematical formulation



- N agents.
- An amount C of resource (e.g., power).
- Each f_i is private to agent i. Not known to anybody else!

Mathematical formulation

- N agents.
- An amount C of resource (e.g., power).
- Each f_i is private to agent i. Not known to anybody else!

Problem

Optimize social utility (sum over all agents).

Maximize

$$\sum_{i=1}^{N} f_i(x_i)$$

subject to

$$\sum_{i=1}^{N} x_i = C\,, \quad x_i \geq 0.$$

SmarterCities Technology Centre

Optimization is easy, is it not?

Maximize

$$\sum_{i=1}^{N} f_i(x_i)$$

subject to

$$\sum_{i=1}^{N} x_i = C\,, \quad x_i \geq 0.$$

If the f_i are strictly concave, this is an easy optimization problem

The conditions for optimality are simply, that in the optimum x^* satisfies for all i and j:

$$f_i'\left(x_i^*\right) = f_j'\left(x_j^*\right).$$

Optimization is easy, is it not?

Maximize

$$\sum_{i=1}^{N} f_i(x_i)$$

subject to

$$\sum_{i=1}^{N} x_i = C \,, \quad x_i \geq 0.$$

If the f_i are strictly concave, this is an easy optimization problem, **if we can use the full force of optimization theory**.

The conditions for optimality are simply, that in the optimum x^* satisfies for all i and j:

$$f_i'\left(x_i^*\right) = f_j'\left(x_j^*\right) .$$

Solving optimization problem by AIMD

Define:

- $x_i(t)$: amount of "demand" by agent i at time t,
- $\bar{x}_i(t)$: average over the past, i.e.

$$\bar{x}_i(t) = \frac{1}{t+1} \sum_{s=0}^{t} x_i(s)\,.$$

Solving optimization problem by AIMD

Initialize: α and β are fixed.

At time $t = 1, 2, \ldots$:

- Each agent receives one-bit warning signal $1_{[W_t < C]}$, where $W_t = \sum_j x_j(t)$. This is usally from central operator, but can sometimes be computed locally (e.g., measuring pollution to estimate power demand).
- Each agent i computes $\lambda_i(\bar{x}_i(t))$ (knowing only f_i, details later).
- Each agent i updates its own demand in an recursive fashion:

$$x_i(t+1) = \left\{x_i(t) + \alpha\right\} 1_{[W_t < C]} + \left\{B_i(t) x_i(t)\right\} 1_{[W_t \geq C]},$$

$$\begin{aligned} B_i(t) &= 1 \text{ with probability } 1 - \lambda_i(\bar{x}_i(t)) \\ &= \beta \text{ with probability } \lambda_i(\bar{x}_i(t)). \end{aligned}$$

This means back-off with probability $\lambda_i(\bar{x}_i(t))$.

Good news

- To compute the warning signals, you only need to observe the sum $\sum_j x_j(t)$.
- A simpler version of AIMD is already used in TCP.
- Each agent i only needs to know its own f_i.

Theorem

Suppose that $\{f_i\}$ are strictly concave. We can design the probability functions λ_i so that for every agent i:

$$x_i(t) \to x_i^*, \quad \text{as } t \to \infty.$$

How do we update λ_i?

For fixed probabilities λ_i, a well-known property of AIMD algorithm is:

Theorem

$$\lim_{t\to\infty} x_i(t) = \frac{\alpha}{\lambda_i(1-\beta)}, \quad \text{almost surely.}$$

Leap of Faith

Set λ_i as follows for all i:

$$\boxed{\lambda_i(z) := \frac{f_i'(z)}{z}}$$

The Magic Formula

$$\boxed{\lambda_i(x_i(t)) := \frac{f_i'(x_i(t))}{x_i(t)}}$$

The Magic Formula

$$\boxed{\lambda_i(x_i(t)) := \frac{f_i'(x_i(t))}{x_i(t)}}$$

If the algorithm converges ($x(t) \to \tilde{x}$) , then for all i:

$$\lim_{t\to\infty} \lambda_i(x_i(t)) = \frac{f_i'(\tilde{x}_i)}{\tilde{x}_i}$$

(by the property of AIMD) $\lim_{t\to\infty} x_i(t) = \tilde{x}_i = \dfrac{\alpha}{\frac{f_i'(\tilde{x}_i)}{\tilde{x}_i}(1-\beta)}$;

or equivalently

$$f_i'(\tilde{x}_i) = \frac{\alpha}{1-\beta}, \qquad \text{for all } i\,.$$

The Magic Formula

IBM

$$\boxed{\lambda_i(x_i(t)) := \frac{f_i'(x_i(t))}{x_i(t)}}$$

If the algorithm converges ($x(t) \to \tilde{x}$) , then for all i:

$$\lim_{t\to\infty} \lambda_i(x_i(t)) = \frac{f_i'(\tilde{x}_i)}{\tilde{x}_i}$$

(by the property of AIMD) $$\lim_{t\to\infty} x_i(t) = \tilde{x}_i = \frac{\alpha}{\frac{f_i'(\tilde{x}_i)}{\tilde{x}_i}(1-\beta)};$$

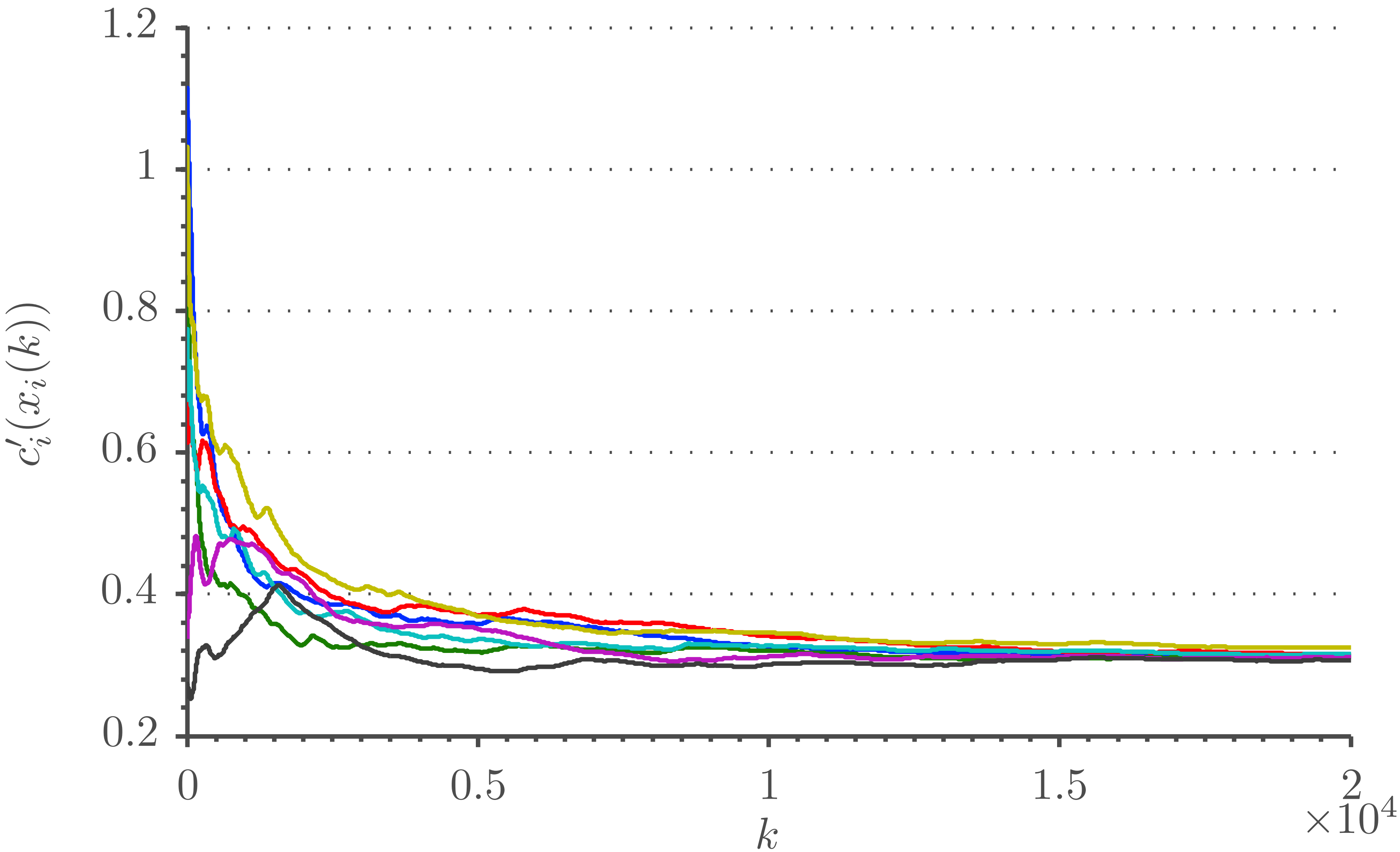
or equivalently

$$f_i'(\tilde{x}_i) = \frac{\alpha}{1-\beta}, \qquad \text{for all } i\,.$$

Remains to show convergence indeed occurs . . .

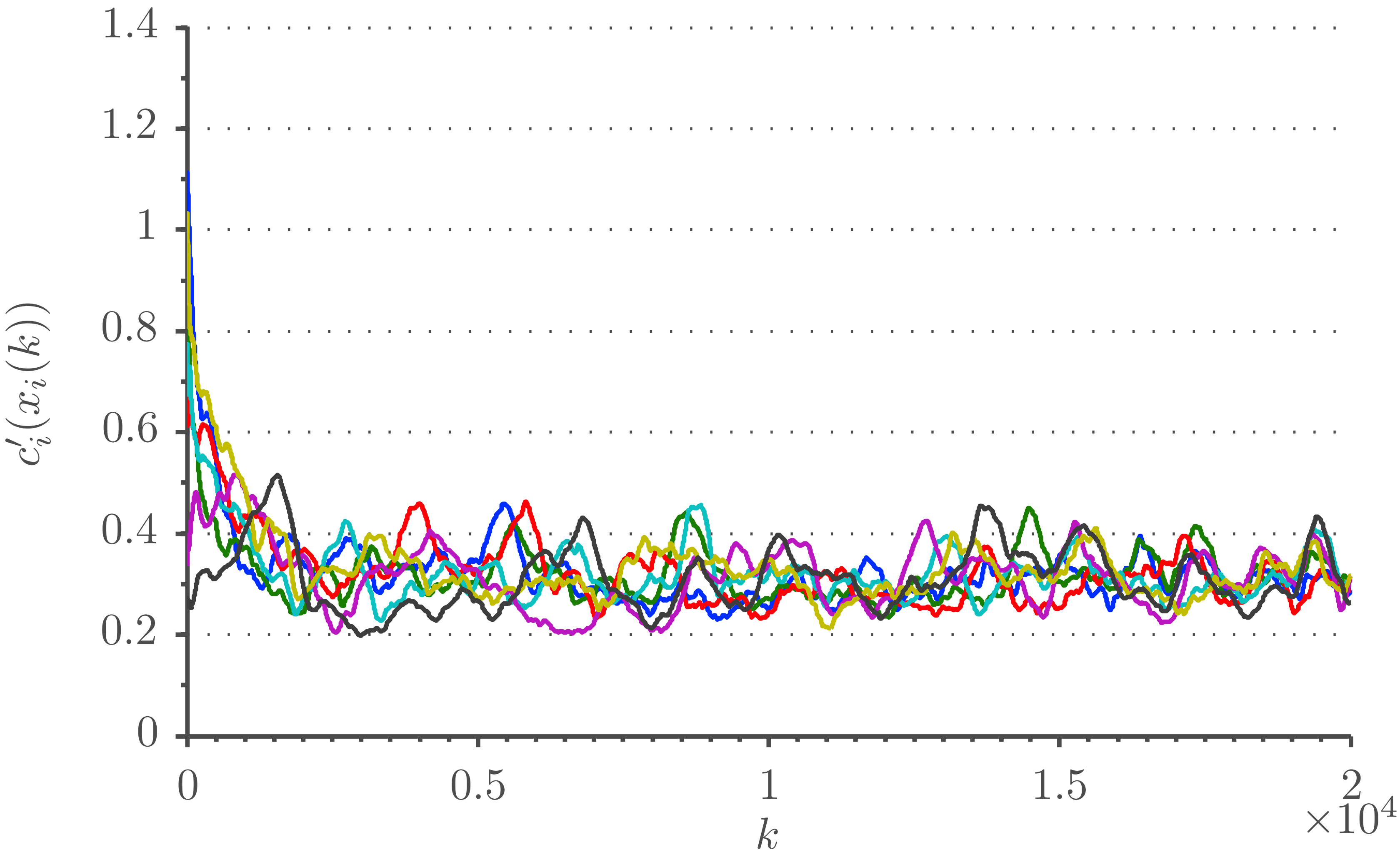
Optimization with infinite averaging

$\bar{x}_i(t) = \frac{1}{t+1} \sum_{s=0}^{t} x_i(s)$



Optimization with bounded window size

$$\bar{x}_i(t) = \frac{1}{\tau} \sum_{s=t-\tau}^{t} x_i(s)$$



Next Steps

- Other versions with adaptive α_i and β_i on top of λ_i.
- Making it robust: no agent has any incentive to deviate from the AIMD policy.

Paper:

`http://arxiv.org/abs/1404.5064`