Distributed computation of optimal allocations using potential games

Pierre Coucheney, Corinne Touati, Bruno Gaujal

INRIA Alcatel-Lucent Common Lab.

Alge(co)Fail

Motivation: Distributed computations using equilibria in games

From the beginning, game theory has been concerned with equilibria in distributed systems.

Computational issues were often ignored up to the point that a "new" field has emerged: algorithmic game theory.

Motivation: Distributed computations using equilibria in games

From the beginning, game theory has been concerned with equilibria in distributed systems.

Computational issues were often ignored up to the point that a "new" field has emerged: algorithmic game theory.

The same kind of questions are now arising in population games where it may not be enough to provide a dynamical system converging to equilibria.

In this talk, I will present an effective distributed algorithm for a class of games solving an optimal allocation problems in wifi networks.

Outline and Main Result

1: Model.

- Model an optimization problem as a potential game.

2: Algorithm.

We provide a distributed algorithm "following" the replicator dynamics and show that:

- it converges to a pure strategy.
- converges to a local maximum of the objective function.

3: Experimental results and several extensions.

Simulation of the algorithm.

Model with general throughput

We consider a set $\mathcal{N}$ of users that can connect to a fixed set of *base stations* (BS), of various technologies (WiFi, WiMAX, UMTS, LTE...).
The set of BSs that user n can connect to is denoted by $\mathcal{I}_n$. An allocation s_n for user n is the choice of a BS $i \in \mathcal{I}_n$. The state of a BS (absence of presence of every user) is a binary vector ℓ. The thoughput of user n under allocation s is denoted $u_n(\ell(s))$
Only assumption:

$$\forall \ell \in \{0;1\}^N, \forall n \in \mathcal{N}, \quad U_{\min} \le u_n(\ell) \le U_{\max}. \tag{1}$$

Global Objective

Objective

Find an allocation s of users to BSs that maximises the α fair throughput.:

$$\max_s \sum_{n \in \mathcal{N}} u_n^\alpha(\ell(s))$$

The α-modified throughput is $u_n^\alpha(\ell) \stackrel{\text{def}}{=} G_\alpha(u_n(\ell))$ with $G_\alpha(x) \stackrel{\text{def}}{=} \dfrac{x^{1-\alpha}}{1-\alpha}$.

The best allocation s must be computed by a fully distributed algorithm where BSs don't see each other and users can only exchange information with their currently attached BS.

Global Objective

Objective

Find an allocation s of users to BSs that maximises the α fair throughput.:

$$\max_s \sum_{n \in \mathcal{N}} u_n^\alpha(\ell(s))$$

The α-modified throughput is $u_n^\alpha(\ell) \stackrel{\text{def}}{=} G_\alpha(u_n(\ell))$ with $G_\alpha(x) \stackrel{\text{def}}{=} \frac{x^{1-\alpha}}{1-\alpha}$.

The best allocation s must be computed by a fully distributed algorithm where BSs don't see each other and users can only exchange information with their currently attached BS.

This rather general optimization problem can be solved using potential games.

Population Game

We model the user-network association problem by a game in which each user is seen as a player.

For user n, the choice s_n is the type of BS (or equivalently, network) that user n chooses to connect to.
We denote by $q_{n,i}$ the probability for user n to choose network i:

$$q_{n,i} = P(S_n = i).$$

Population Game: repercussion payoffs

the set of *repercussion payoffs* is

$$r_n^\alpha(\ell^{s_n}(s)) \stackrel{\text{def}}{=} u_n^\alpha(\ell^{s_n}(s)) - \sum_{m \neq n : s_m = s_n} \left(u_m^\alpha(\ell^{s_m}(s) - e_n) - u_m^\alpha(\ell^{s_m}(s))\right),$$

With no loss of generality, the repercussion payoffs are assumed to be positive (by adding a constant C_α to all throughputs, depending on the upper and lower bounds).
The game with mixed strategies has expected payoff of a packet from user n and type i : $f_{n,i}(q) \stackrel{\text{def}}{=} \mathbb{E}[r_n^\alpha(\ell^i(S))|S_n = i]$.
Mean Payoff over all BS $\overline{f}_n(q) \stackrel{\text{def}}{=} \sum_{i \in \mathcal{I}_n} q_{n,i} f_{n,i}(q)$.

($f_{n,i}(q)$ only depends on $(q_{m,i})_{m \neq n}$, multi-linear function of $(q_{m,i})_{m \neq n}$).

Potential Game

Theorem 1.

The repercussion game is a potential game, i.e. $\forall n, \forall i, f_{n,i}(q) = \dfrac{\partial F}{\partial q_{n,i}}(q)$, where F is its associated potential function, and:

$$F(q) = \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}_n} q_{n,i} \mathbb{E}[u_n^\alpha(\ell^i(S)) | S_n = i].$$

This implies that every local maximizer of the potential is an ESS [Sandholm 2001].
Also, since the potential is multilinear over a convex set, at least one local optimum is pure.

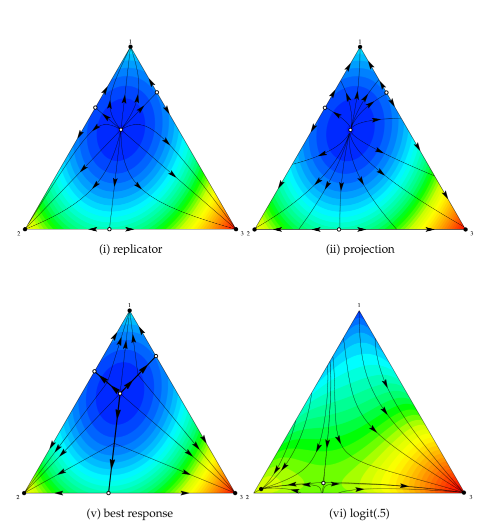
Equilibrium points of potential games have been shown to be rest points of dynamical systems [Sandholm 01]:

$$\dot{q} = G(q).$$

- Replicator: $\dot{q}_i = q_i(f_i - \overline{f})$
- Projection: $\dot{q}_i = Proj_{\Delta}(f)_i$
- Best Response: $\dot{q}_i = BR_i(q) - q_i$
- Loggit: $\dot{q}_i = \dfrac{e^{f_i/K}}{\sum_j e^{f_j/K}} - q_i$

Dynamics II

The rest points of these dynamics - if they exist - are (perturbed) equilibria of the game. From [Sandhlom, 2001],



Computation Issues

This is not the end of the story: how do you come up with an algorithm to compute the equilibria?
A numerical integration of the differential equation is not always good enough.

Computation Issues

This is not the end of the story: how do you come up with an algorithm to compute the equilibria?
A numerical integration of the differential equation is not always good enough.
One may want:

- Select Lyapounov stable points.
- Select good points.
- Resilience to small errors and/or to a small number of malicious individuals.
- A distributed computation done by the users (synchronized or not).
- An incentive for each user to execute the algorithm.

Replicator Dynamics

Recall that the replicator dynamics [weibull 97, hofbauer 03] is

$$\forall n \in \mathcal{N}, i \in \mathcal{I},\ \frac{dq_{n,i}}{dt} = q_{n,i}\left(f_{n,i}(q) - \overline{f_n}(q)\right).$$

Intuitively, this dynamics can be understood as an update mechanism where the masses associated to networks whose expected payoff are more than the average payoff will increase in time, while non profitable networks will gradually be abandoned.

Properties of the Replicator Dynamics

Theorem 2.

All the asymptotically stable sets of the replicator dynamics are faces of the domain Δ. These faces are sets of equilibrium points for the replicator dynamics.

This is because the replicator dynamics preserves a certain form of volume [Akin 83] so that no interior set can be an attractor.

Theorem 3.

[Coucheney, Gaujal, Touati, 2008] If an asymptotically stable face of the replicator dynamics is reduced to a single point, then it is an ESS, a Wardrop and a Nash equilibrium of the game.

Algorithm:

For all $n \in \mathcal{N}$:

- Choose initial strategy $q_n(0)$.
- At each time epoch t:
  - Choose s_n according to $q_n(t)$.
  - Update: $q_{n,i}(t+1) = q_{n,i}(t) + \epsilon\, r_n(\ell^{s_n}) \left(\mathbb{1}_{s_n=i} - q_{n,i}(t) \right)$.

repercussion utility

constant step size

$\begin{cases} 1 \text{ if } s_n = i \\ 0 \text{ otherwise} \end{cases}$

Simple computation for the mobile.

1) The algorithm is a stochastic approximation of the replicator dynamic differential equation with constant step size:
$q_{n,i}(t+1) = q_{n,i}(t) + \epsilon\, b(q_{n,i}(t), S_n(t))$.
$\mathbb{E}[b(q_{n,i}, S_n)] = q_{n,i}(f_{n,i}(q) - \overline{f}(q))$.

2) It is fully distributed.

Properties of the algorithm

Theorem 4.

[Couchenev, Gaujal, Touati, 2008] The values of q computed by our algorithm weakly converge to a set of pure Nash equilibria of the allocation game with repercussion utilities, that locally maximize the global α-fair throughput.

The proof uses the fact that q is a martingale over stable faces converging to pure points. Out of stable faces, the behavior of q is close to the behavior of the martingale with a high probability (using a coupling argument).

Distributed algorithms for other dynamics

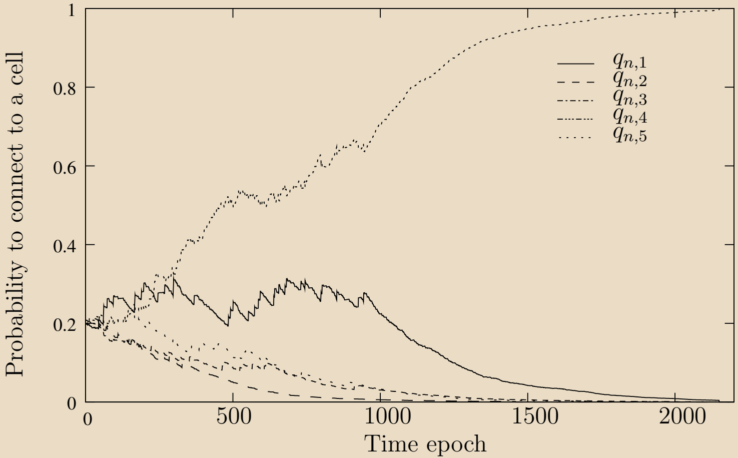
Consider a dynamics of type $\dot{q} = G(q)$.
To construct a distributed stochastic approximation, one has to find a function H such that
$q_{n,i}(t+1) = q_{n,i}(t) + \epsilon H(q_{n,i}(t), S_n(t))$
such that
$\mathbb{E}[H(q_{n,i}, S_n)] = G_i(q)$.

Distributed algorithms for other dynamics

Consider a dynamics of type $\dot{q} = G(q)$.
To construct a distributed stochastic approximation, one has to find a function H such that
$q_{n,i}(t+1) = q_{n,i}(t) + \epsilon H(q_{n,i}(t), S_n(t))$
such that
$\mathbb{E}[H(q_{n,i}, S_n)] = G_i(q)$.

When G is linear (as with replicator) this is an easy task. For non-linear dynamics (such as Proj, BR, Logit) , this can be very difficult or even impossible to get in closed form.

Convergence to Fixed Association for User n



Evolution of one user's strategy that can connect to 5 cells.

Is the local optimum a global one?

Suppose we initialize the algorithm with

$$\forall n \in \mathcal{N}, i \in \mathcal{I}_n, \quad q_{n,i}(0) = \frac{1}{|\mathcal{I}_n|}.$$

In the case of multiple local equilibria, will the algorithm converge to the global maximum?

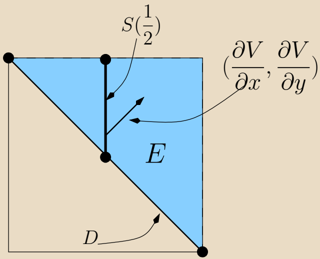
Two player, two action game

Theorem

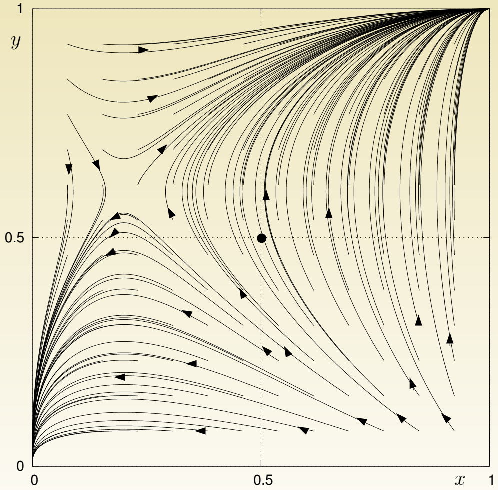
In a two player, two action allocation game with repercussion utilities, the initial point of the algorithm is in the basin of attraction of the global maximum.

proof

$V(x, y) = |1 - x| + |1 - y|$ is a Lyapunov function on the upper right triangle

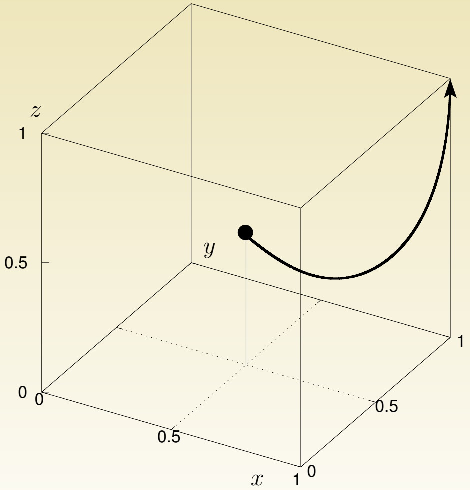


Two player, two action game (contd.)



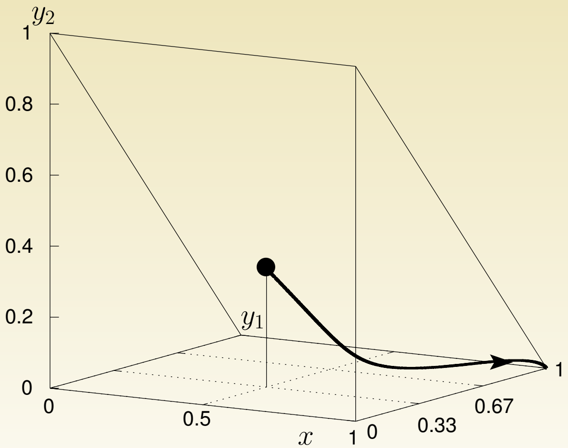
2 maxima: the point $(\frac{1}{2}, \frac{1}{2})$ is inside the attracting basin of the global maximum.

Extension to more than two players



3 players, with 2 choices each. The dynamics converges to the point $(1, 1, 1)$ whereas the global maximum is $(0, 0, 0)$.

Extension to more than two choices

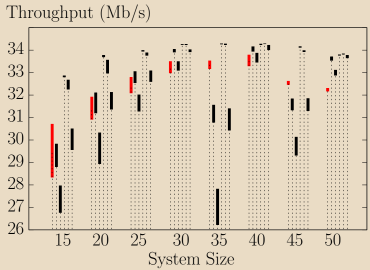


Example with 2 players, one has 3 choices and the other has 2. The dynamics starting in $(1/2, 1/3, 1/3)$ converges to $(1, 1, 0)$ whereas the global maximum is $(0, 0, 0)$.

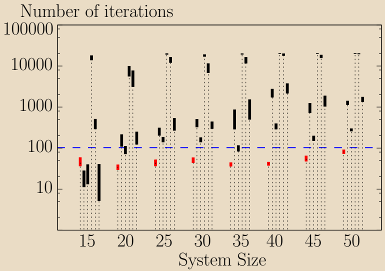
Convergence Speed: Adapt Step Size ϵ

6 heuristics for the choice of $\boxed{\epsilon_n(t)}$

$$q_{n,i}(t+1) = q_{n,i}(t) + \boxed{\epsilon_n(t)}\, r_n(\ell^{s_n})\,(1_{s_n=i} - q_{n,i}(t)).$$



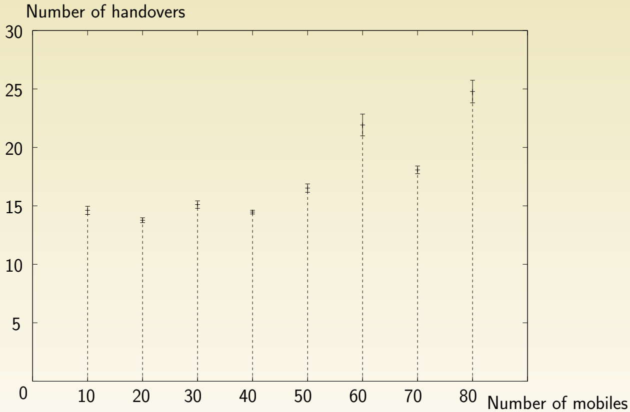
Average performance.



Average number of iterations (log. scale).

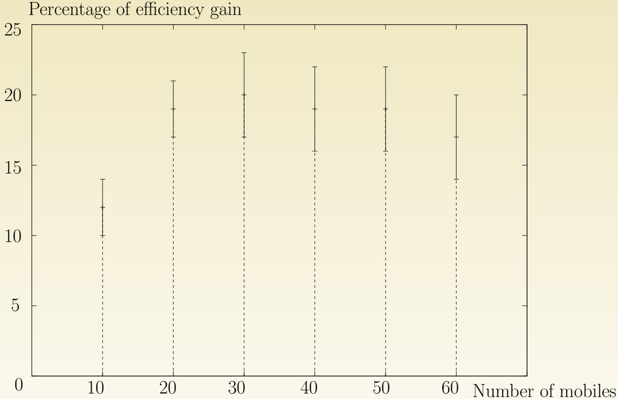
(with 5% confidence intervals).

Number of changes



Mean number of handovers for a user as a function of the total number of users.

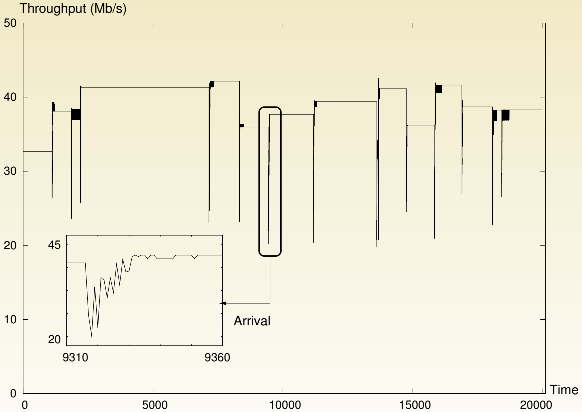
Improvement over naive allocations



Percentage of efficiency gain by using our algorithm in comparison to the fixed choice of the best cell BS for each user.

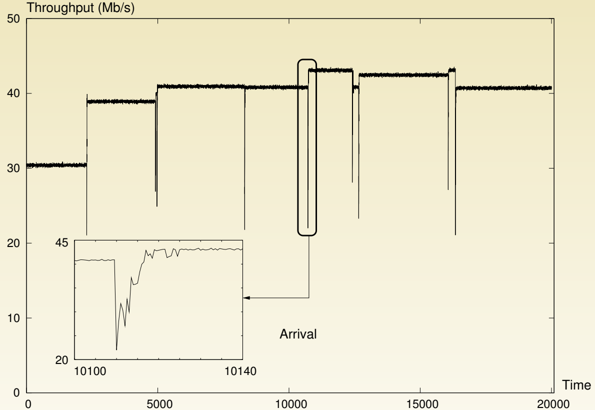
Extension 1: Mobility of users

When the set of users is not static but undergoes arrivals, departures and mobility, the association algorithm has to be run at every arrival or departure of a user.



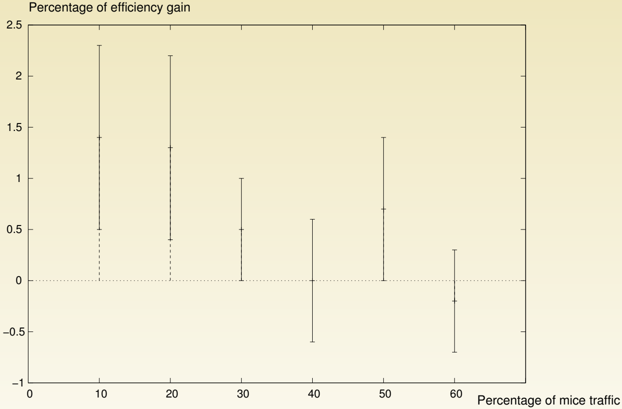
Adaptation to arrivals and departures: the algorithm smoothly and quickly reconverges after changes.

Extension 2: White Noise on the Measurements



Stability with respect to measurement errors: behavior of the algorithm when the throughput of all cells has a white Gaussian noise.

Extension 3: Mice and Elephant Traffic



Percentage of gain by running the algorithm for mice and elephants instead of running it only for elephants. The percentage of mice traffic vary, but the global traffic average is constant.

Conclusion

Conclusion

- Distributed algorithm.
- Convergence to a locally optimal fixed association.
- Very simple computation needed.
- Fast convergence (a few tens) with simple heuristics for the choice of step size ϵ.

Future works

- Analytically study convergence speed.
- Discuss the relevance of the throughput as a utility function for different kinds of applications (*e.g.* latency).
- Investigate links with optimal control under mobility conditions.