université PARIS DIDEROT PARIS 7

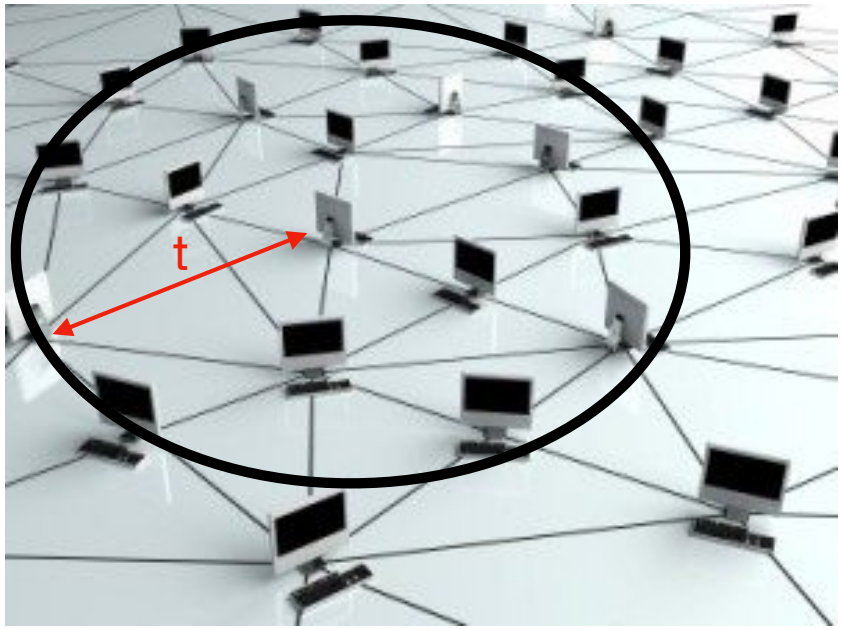
cnrs

Equilibria of Games in Networks for Local Tasks

Simon Collet Pierre Fraigniaud Paolo Penna

INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

Local Tasks



Tasks that can be solved locally in networks.

Every node outputs after having consulted information stored at nodes in its vicinity.

Ideally radius $t = O(1)$ or $t = O(\log^{O(1)} n)$

t = #rounds performed by the algorithm

Examples:

- Vertex-coloring (e.g., for frequency assignment)
- Independent set (e.g., for scheduling)
- Dominating set (e.g., to serve as cluster head)
- etc.

LCL Tasks

- Locally checkable labelings (LCL) are tasks whose solution can be *checked* locally.
- Coloring, MIS, dominating set, etc., are LCL tasks.
- An LCL is characterized by
  - a set $\mathcal{L}$ of node labels
  - a set $\mathcal{B}$ of labeled balls with radius r
- **Task:** Every node of network G must compute a label in $\mathcal{L}$ such that, for every node v, the ball $B_G(v,r)$ is in $\mathcal{B}$.

A generic randomized algorithm for LCL tasks

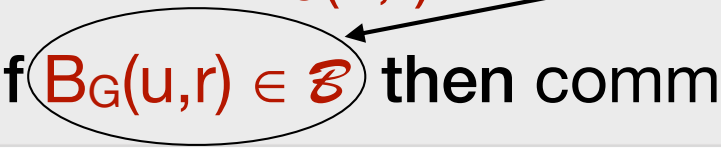
Algorithm of node $u \in V(G)$

Repeat

observe $B_G(u,r)$

select a label $\ell(u) \in \mathcal{L}$ at random according to $D(u)$

observe $B_G(u,r)$



if $B_G(u,r) \in \mathcal{B}$ **then** commit with label $\ell(u)$ and stop

The distribution $D(\cdot)$ can be uniform, but is often biased to increase the probability of constructing a good ball.

Also, $D(\cdot)$ can be different at different nodes, and may vary along with the execution of the algorithm.

Examples

1. Maximal Independent Set (MIS)

Luby's algorithm performs in $O(\log n)$ rounds w.h.p.

- Pr[u proposes itself to enter the MIS] ≈ $1/\deg(u)$

2. $(\Delta+1)$-coloring in max-degree-Δ networks

Barenboim & Elkin's algorithm performs in $O(\log n)$ rounds w.h.p.

- Pr[u participates in the phase] = $1/2$
- Pr[u proposes color c] ≈ uniform among available colors

Limits of the Generic Algorithm

- MIS or dominating sets can be used to construct a backbone in a radio network
  - being part of the backbone might be undesirable (e.g., because it causes high energy consumption)
- Vertex coloring can be used to assign radio frequencies to nodes in a radio network
  - some frequencies might be preferred (e.g., because they interfere with local transmitters)

➡ Some selfish nodes might be tempted to deviate from the algorithm, by not respecting the specification of the random distribution D governing the choice of the labels.

Framework

- Nodes *communicate honestly* their state, and *correctly transfer messages*, e.g., to avoid being caught.
- Selfish nodes may privately rationally cheat *about their choices* of the randomly selected labels.

Nodes want to solve the problem quickly because the solution provides some desirable service

Every node has preference for some of the solutions, and may wish to avoid undesirable solutions

The Game

- **Players:** the n nodes of a network $G=(V,E)$
- **Strategy** of node u: a distribution $D(u)$ over the good balls centered at u
- **Payoff** for node u:
  - $\text{pref}_u : \mathcal{B} \rightarrow [0,1]$
  - k = #rounds for the algorithm to terminate at u
  - Payoff:

$$\pi_u = \text{pref}_u(B) / 2^k$$

u aims at being the center of a ball that it prefers

u aims at terminating quickly

where $B = B_G(u,r)$ is the ball around u when the algorithm terminates at u

Question

What form of *equilibria* can be derived for LCL games?

Related Work (general)

	Strategic Games	**Extensive games with perfect information**	**Extensive games with imperfect information**
Finite games	[25] Nash equilibrium Mixed strategies	[28] Subgame-perfect equilibrium Pure strategies	[29] Trembling-hand perfect eq. Behavior strategies
Games with a finite action set		[12] Subgame-perfect equilibrium Pure strategies	[12] Sequential equilibrium Behavior strategies
Games with an infinite action set	[11] [14] Nash equilibrium Mixed strategies	[16] Subgame-perfect equilibrium Pure strategies	[10] Nash equilibrium Behavior strategies

Table 1 *A summary of results about the existence of equilibria*

Classical game theoretical results do not directly apply to LCL games because:

- the usual notion of imperfect information is *solely related to the fact that players play simultaneously*
- in LCL games, imperfect information also refers to the fact that each node is *not aware of the states of far away nodes* in the network.

Related Work (games in networks)

Distributed computing by rational agents:

- Abraham, Dolev, and Halpern [DISC 2013]
- Afek, Ginzberg, Feibish, and Sulamy [PODC 2014]
- Afek, Rafaeli and Sulamy [DISC 2018]

Framework:

- agents strategies define the algorithm itself, including which messages to send, which information to reveal
- the algorithms are “global” (they can take Ω(n) rounds)
- specific tasks are analyzed

Our result

A *trembling-hand perfect* equilibrium is a stronger form of Nash equilibrium.

- In Nash equilibria, players are assumed to play precisely as specified by the equilibrium.
- Trembling-hand perfect equilibria include the possibility of *off-the-equilibrium* play (players may, with small probabilities, choose unintended strategies).

Theorem
For any (greedily constructible) LCL task, the associated game has a symmetric trembling-hand perfect equilibrium.

Implications

For every LCL game, there is a distributed strategy from which the players have no incentive to deviate, in a robust sense (i.e., it supports small deviations).

➥ One can keep control of the system even in the presence of rational selfish players optimizing their own benefit.

Techniques

Lemma 1 Every infinite, continuous, measurable, well-rounded, extensive (symmetric) game with perfect recall and finite action set has a (symmetric) trembling-hand perfect equilibrium.

Lemma 2 LCL games are symmetric, infinite, continuous, measurable, well-rounded, extensive games with perfect recall and finite action set.

Conclusion and Open Problems

- We have proved that natural games occurring in the framework of local distributed network computing have trembling-hand perfect equilibria, a strong form of Nash equilibria.

What are the performances of the robust algorithms resulting from these equilibria?

- Note that determining the performances of iterative distributed construction algorithms such as the generic algorithm is non trivial, even if nodes follow the prescribed actions imposed by the algorithm (e.g., Luby's algorithm).

Thank you!