Journées SMAI MODE 2018

Distributed Best Response in Random Potential Games

Stéphane Durand
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, GIPSA-Lab

Federica Garin
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, GIPSA-Lab

Bruno Gaujal
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG

Résumé. In this paper we design and analyze a distributed algorithm to compute a Nash equilibrium in potential games based on best response. This algorithm can be implemented in a distributed way using Poisson clocks and does not rely on the usual assumption that players take no time to compute their best response and change their strategy. We show that if one takes this time lag into account, the algorithm may suffer from *collisions* (one player starts to play while another player has not changed her strategy yet). We first show that collisions do not jeopardize convergence to a Nash equilibrium but our main result is to show that the average time complexity of the algorithm (time to reach a Nash equilibrium) is bounded by $\frac{2n\log n}{1-p} + o(n\log n)$, where n is the number of players and p is the probability that a player gets in a collision. Furthermore, exhaustive simulations suggest that the term $\log n$ is superfluous.

Mots-clefs : Potential Games, Best Response, Markov Chains, Distributed Algorithms

1 Potential Games

A game $\mathfrak{G} \stackrel{\text{def}}{=} \mathfrak{G}(\mathcal{N}, \mathcal{A}, u)$ is a triplet consisting of: A finite set of *players* $\mathcal{N} = \{1, \dots, n\}$; A finite set $\mathcal{A}$ of *actions* (or *pure strategies*) ; The set of *actions profiles* or *states* of the game is $\mathcal{A}^n$. The players' *payoff functions* $u_k : \mathcal{A}^n \to \mathbb{R}$, for each $k \in \mathcal{N}$.
The *best response correspondence* $\mathcal{BR}_k(x)$ is the set of actions maximizing the payoff for player k under state $\mathbf{x}$:

$$\mathcal{BR}_k(\mathbf{x}) \stackrel{\text{def}}{=} \left\{ \arg\max_{\alpha_k \in \mathcal{A}} u_k(\alpha_k; \mathbf{x}_{-k}) \right\}.$$

A *Nash equilibrium* (NE) is a fixed point of this correspondence, i.e., a profile x^* such that $x_k^* \in \mathcal{BR}_k(x^*)$ for every player k.
A game is a *(best response) potential game* if there is a function $F : \mathcal{A}^n \to \mathbb{R}$ such that for any player k and action profile $\mathbf{x}$

$$\mathcal{BR}_k(\mathbf{x}) = \left\{ \arg\max_{\alpha_k \in \mathcal{A}} F(\alpha_k, \mathbf{x}_{-k}) \right\}.$$

It is well-known that the Best Response Algorithm (BRA) converges to a pure Nash equilibrium in potential games [2]. BRA suffers from two main drawbacks when used in a distributed context. Firstly, the impact of the revision sequence (order of play) on the performance is still unknown. Secondly, BRA requires that players play one at a time and this is hard to guarantee in a distributed context. In this work, we answer to both points together by proposing a distributed version of the Best Response Algorithm where each player plays according to independent Poisson clocks and runs an explicit termination test. Our model of a distributed system is quite general and uses the classical CREW-PRAM model (Concurrent Read Exclusive Write, Parallel Random Access Memory).

2 Distributed Algorithm with Termination Test

Algorithm 1: Distributed BRA with termination test

Function MAIN ALGORITHM
  Input: Game utilities $(u_k(\cdot))$; Initial state $(\mathbf{x} := \mathbf{x}(0))$;
  Local clock, ticking w.r.t. a Poisson process with rate λ/n;
  repeat
    On *each tick of the Poisson clock*
      If $x_i \notin \mathcal{BR}_i(\mathbf{x})$ **Then** Update strategy to $x_i \in \mathcal{BR}_i(\mathbf{x})$; With probability q:
      Call Termination Test Sender;
    On Reception of *Stop* **Call** Termination Test Receiver;
  until End *sent or received*;
Function TERMINATION TEST, SENDER
  Stop Clock;
  Send(Stop) to all players;
  wait until n acks received;
  Send(Test) to all players;
  wait until n messages received;
  if n 'Stable' messages received **then** **Send** End;
  Else Restart Clock;
Function TERMINATION TEST, RECEIVER
  Stop Clock;
  Send(Ack) to sender;
  wait until Test received;
  If $\mathcal{BR}(\mathbf{x}) = x_i$ **Send**(Stable) to sender;
  else **Send**(Unstable) to sender;
  Restart Clock;

To design this algorithm, let us consider that each player is equipped with a Poisson clock that ticks at rate λ/n. At each tick of her clock, a player computes her best response under the current state and updates her action. We denote by C the time taken by one player (say k) to compute her best response, $\mathcal{BR}_k(x)$, under state x and to update her action. We assume that C does not depend on the player nor on the current state. Note that when one player starts playing while the previous player has not finished (this is called a *collision*), then this can be seen as a *simultaneous play*, and the potential may decrease during a collision.

Let us also consider the following communication procedures named *Termination Test Sender* and *Termination Test Receiver* in Algorithm 1: At every tick of her clock, each player, with probability q, broadcasts a message to every other player. Upon reception of such a message, the receivers interrupt their clock and send an acknowledgement (ack). Once the initial sender gets all the acks, she sends a second message. Upon reception of this second message, each player tests if she needs to change her best response (a player is *stable* if no change is needed) and sends back her stable/unstable status before restarting her clock. The initial sender receives n confirmations of stability only if the current state is a Nash equilibrium.
This global communication operation interrupts the Poisson clock of all the players during two broadcasts and the clock of one player for four broadcasts. We denote the average overall interruption time by σ_n (it may depend on n, as discussed later).
It should be clear that this algorithm terminates with probability one in finite time and its final state is a pure Nash equilibrium for all potential games. This is quite straightforward and relies on the Borel-Cantelli Lemma. The worse-case complexity of BRA is known to be exponential; however, the average complexity (over all potential games) is much better, as shown in [1] in the centralized case, and as shown below for Algorithm 1.

Theorem 1 *Using the best value for q, the execution time T of Algorithm 1, satisfies*

$$\mathbb{E}[T] \quad \leq \quad \frac{2n\log n}{\lambda(1-p)} + 2\sqrt{\frac{2\sigma_n n\log n}{\lambda(1-p)}} + \sigma_n + \varepsilon(n).$$

where $p = 1-\exp(-\lambda C)$ (probability of collision), $\varepsilon(n)$ is negligible with respect to the previous terms and the expectation is taken w.r.t. the random potentials and w.r.t. the Poisson clocks.

If the broadcast time σ_n takes the usual form $\sigma_n = C_2\log(n)$, then the average complexity becomes $\mathbb{E}[T] \leq \frac{2n\log n}{\lambda(1-p)} + 2\sqrt{\frac{2C_2 n}{\lambda(1-p)}}\log n + C_2\log(n) + \varepsilon(n)$.
Comparing this with a centralized BRA with an IID revision sequence and no collisions [1] shows that the convergence time is only affected by a multiplicative factor $2/\lambda(1-p)$.
The proof is based on several ingredients. The first two are similar to what is done in [1].
1. Randomization of the potential can be done with no loss of generality by assuming that all profiles have IID potentials uniformly distributed in $[0,1]$.
2. Consider the approximation that during the execution of the algorithm, all states $(x_1,\ldots,x_n)$ are examined at most once: The algorithms with and without this approximation have the same asymptotic complexity.
3. under 1. and 2., the behavior of the algorithm can be seen as a Markov chain whose state space is a product of a discrete state (number of players currently playing their best response) and a continuous state (the current potential, a real number in $[0,1]$). Computing the expected hitting time of a NE is done by solving a system of differential equations whose solution has explicit asymptotics in n.

Références

[1] Stéphane Durand and Bruno Gaujal. Complexity and optimality of the best response algorithm in random potential games. Research Report RR-8925, Inria, June 2016.

[2] Dov Monderer and Lloyd Shapley. Potential games. *Games and economic behavior, Elsevier*, 14(1):124–143, 1996.