Analysis of Best Response Dynamics in Potential Games

Stephane Durand

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ DE GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 Mai 2016

Analysis of Best Response Dynamics in Potential Games

Présentée par
Stéphane Durand

préparée au sein des laboratoires
LIG et **GIPSA-LAB**
dans **L'école doctorale Mathématiques, Sciences et technologies de l'information, Informatique (MSTII)**

Thèse soutenue publiquement le **11 décembre 2018**,
devant le jury composé de:

Giacomo Como, Rapporteur
Assistant Professor, Politecnico di Torino et Lund University
Jean Mairesse, Rapporteur
Directeur de Recherche CNRS , ESIEE
Olivier Brun, Examinateur
Directeur de recherche, CNRS , LAAS
Johanne Cohen, Examinatrice
Directrice de recherche, CNRS , LRI
Denis Trystram, Président
Professeur, Université Grenoble-Alpes
Federica GARIN, Co-directrice de thèse
Chargée de recherche, Inria , Gipsa-Lab
Bruno Gaujal, Co-directeur de thèse
Directeur de recherche, Inria , LIG

Contents

Chapter 1

Introduction

In game theory, Nash equilibria are in many ways solutions of a game. They are stable points, making them probable states when modeling a system or providing anti cheating guaranties when designing one. Unfortunately, they are also difficult to compute for general matrix games. It has been shown in [4] that the computation of a Nash equilibrium in a two player game is PPAD-complete. PPAD is a class of complexity known to be between P and NP.

Potential games (games admitting a potential function, as described in Chapter 2) have been introduced in [17]. They cover a wide class of problems, such as routing games ([3]), distributed optimization ([18]), congestion in transportation, either for road traffic or communication networks [8, 16, 22]. A particularly interesting subclass of potential games for computer science is *congestion games*: a set of users share resources in a competitive way. In fact congestion games and potential games are equivalent ([14]).

All that makes the study of Nash equilibria of potential games an interesting topic.

Finding an equilibria on a potential game is still difficult: it is a PLS-complete problem (Partially Local Search, a class of complexity between P and NP) ([7]). This means that any algorithm should take an exponential time in the number of players and strategies per player in the worst case, unless P=NP. Despite that, there was still a lot of efforts to develop algorithms to compute Nash equilibria. Among them, we will focus in this thesis on one of the earliest and most naive ones: the best response dynamics.

We will describe potential games, Nash equilibrium and the best response dynamics more precisely in Chapter 2.

The best response correspondance was first introduced by J. Nash in [15] as a tool of proof, in order to show the existence of at least one Nash equilibria in any matrix game. In this context. Best response can also be used as an algorithm (*best response dynamics*) to compute Nash equilibria. It was proved in [17] that when used on a potential game best response dynamics converge in finite time to a Nash equilibrium under the condition that players act one at a time.

We will show in Chapter 3 that this condition can be relaxed while still preserving convergence.

As predicted by the complexity class of the problem, the worst case complexity of the best response dynamics is exponential in the number of players. We will exhibit in Chapter 4 an example where the algorithm has to read every potential of the game before finding a Nash equilibrium, and introduce an alternate complexity criterion using average convergence time.

The definition we will use for average convergence time will be explained in Section 4.3. A computation of the average complexity of the dynamics in the simplest case, where players act one after the other in a round robin sequence, will be shown in Section 4.5 using a Markovian model presented in Section 4.4. The same model will be used again to analyze a few variants in Section 4.8 and will be reused in the distributed context in Chapters 5, 6 and 7.

In general, players are considered as independent actors. As such, one could expect them to also be isolated. Namely with limited knowledge of the others. This fits well with the context of distributed systems. This is the central theme of Chapter 5. We will consider a simple model of distributed games using Poisson activations. The chapter will present an upper bound on the complexity of the best response dynamics on distributed system without (Section 5.3) or with (Section 5.4) players acting simultaneously, followed by the proofs (Sections 5.7 and 5.6) respectively) of those bound using the Markovian model.
This chapter will also contain an alternative method to stop the dynamics using a small amount of communications between players in 5.8.

Continuing the idea of limiting information to players, Chapter 6 develops a similar analysis when players have to sample their strategies instead of being able to compute their utilities.

In actual problems using game theory, some players wont be affected by some others. As an example, this could be caused by physical distance: when studying traffic as a routing game over a huge network, travelers staying on one side of the network will not interact much with other staying on the other side. In fact, in such networks, each player will only interact with a few neighbors.
The last chapter of this main part takes this information into account to shorten greatly the computation time. Section 7.1 contains the definition and effect of indifferent players on a game. Section 7.2 applies this on a well coordinated network where concurrent processing is possible and Section 7.3 applies this on a distributed network. Section 7.4 is more numerical. it shows that when the hypothesis of uniformly chosen games, that we needed in the analysis, is absent the results are not heavily affected.

In addition to the work on the best response dynamics, this thesis overview a few additional works on perfect sampling of Markov chains from random walks in graphs in Chapter 9, on the subject of perfect sampling (9.1) and on grid graph with forbidden edges (9.2).

Chapter 2

Potential Games, Best Response

This part consist on the definitions of potential games and of the best response dynamics.

We consider a finite game $\mathfrak{G} \stackrel{\text{def}}{=} (\mathcal{N}, \mathcal{A}, u)$ consisting of

- a finite set of *players* $\mathcal{N} = \{1, \ldots, n\}$;
- a finite set $\mathcal{A}_k$ of *actions* (or *pure strategies*) for each player $k \in \mathcal{N}$; The set of (action) *profiles* or *states* of the game is $\mathcal{A} \stackrel{\text{def}}{=} \prod_k \mathcal{A}_k$;
- the players' *payoff functions* $u_k : \mathcal{A} \to \mathbb{R}$ that players seek to maximize.

In the case of congestion games on network, the players are most often the streams of packet, their choices are their possible paths and their utilities the opposite of the delay.

We define the classical *best response correspondence* $\mathrm{BR}_k(x)$ as the set of all actions that maximizes the payoff for player k under profile x:

$$\mathrm{BR}_k(x) \stackrel{\text{def}}{=} \left\{ \underset{\alpha \in \mathcal{A}_k}{\operatorname{argmax}}\, u_k(\alpha; x_{-k}) \right\}. \tag{2.1}$$

A *Nash equilibrium* (NE) is a fixed point of the correspondence, i.e. a profile x^* such that $x_k^* \in \mathrm{BR}_k(x^*)$ for every player k.

Iteratively playing a best response may not converge in general. We consider here the specific class of potential games for which convergence is ensured.

When used as variables in computations we will write the number of players n, the number of strategies of each players A (we assume they have all the same number of strategies, which does not change much). In the following, in order to simplify notations we will also write the number $A - 1$ or alternatives choices as a.

Definition 1 (**Potential games and variants).** A game is an (exact) *potential game* [13] if it admits a function (called the potential) $F : \mathcal{A} \to \mathbb{R}$ such that for any player k and any *unilateral* deviation of k from action profile x to x'

$$u_k(x) - u_k(x') = F(x) - F(x'). \tag{2.2}$$

The existence of one such function is equivalent to the property that for any cycle, the sum of differences of utility along the cycle is null, which is also equivalent to the sum of difference of utility along any four-cycle is null

If all players share a common goal and valuation, this valuation is both their utility and the potential. In the general case, this function global maximum do not coincide with global maximum on the sum of utility. The local extrema, on the other hand coincide and in many cases, have property bounding the distance to the optimum.
For example, in a congestion game with affine costs, we know that the cost of any Nash equilibrium is at worst $\frac{4}{3}$ of the cost of the optimum .

A game is a *generalized ordinal potential game* [13] (or G-potential game for short) if there is a function $F : \mathcal{A} \to \mathbb{R}$ such that for any player k and any *strictly profitable unilateral* deviation of k from action profile x to x', $F(x') > F(x)$.

A game is a *best-response potential game* [21] (or BR-potential game for short) if there is a function $F : \mathcal{A} \to \mathbb{R}$ such that for any player k and action profile x

$$\mathrm{BR}_k(x) = \left\{ \underset{\alpha \in \mathcal{A}_k}{\operatorname{argmax}} F(\alpha, x_{-k}) \right\}. \tag{2.3}$$

As shown in [21], BR-potential games are characterized by the fact that any sequence of profiles generated by unilateral best response, and containing at least one strict improvement, is not a cycle. In particular, it can be seen that exact potential games are BR-potential games. G-potential games that are not BR-potential games also exist.
To avoid ties, and unless otherwise mentioned, we make the following assumption in the rest of the thesis.

Assumption 1 (**Uniqueness of Best Response).** We assume that the Best Response correspondence is a univalued function; for any player k and profile x:

$$\mathrm{BR}_k(x) = \underset{\alpha \in \mathcal{A}_k}{\operatorname{argmax}} u_k(\alpha; x_{-k}). \tag{2.4}$$

For example, this assumption holds when players face different payoffs at each state:

$$u_k(\alpha, x_{-k}) \neq u_k(\beta, x_{-k}), \tag{2.5}$$

whenever $\alpha \neq \beta$, for all k and all x. This can be imposed by perturbing the payoffs. Also, for general payoffs, breaking ties can be done by ranking players and actions using an arbitrary fixed order.

With this assumption t, any G-potential game is also a BR-potential game.

As mentioned earlier, a G-potential game that satisfies this assumption is necessarily a BR-potential game. Hence all the results will be stated for BR-potential games in the following.

Let us now focus on the *revision protocol*, that is the sequence of players that can revise their strategy. We first consider a version of the asynchronous *Best Response Dynamics* where the next player is selected according to a *revision sequence* (an infinite sequence of players).

Algorithm 1: Best Response Dynamics (BRD) under revision sequence R

Input: Game utilities $(u_k(\cdot))$; Initial state $(\mathbf{x} := \mathbf{x}(0))$; revision sequence R;
Initialize $t := 0$; List of satisfied customers $L := \emptyset$;
while *size(L)*$\neq n$ **do**
  Pick next player $k := R_t$; $t := t + 1$;
  if $x_k \notin BR_k(x)$ **then**
    Update strategy for player k to $x_k \in \mathrm{BR}_k(\mathbf{x})$;
    $L := \emptyset$;
  $L := L \cup \{k\}$;

When the revision sequence is driven by a random process, we denote by ρ the probability distribution, called *revision law*:

$$\forall k \in \mathcal{N}, x \in \mathcal{A},\ \mathbb{P}(\text{selected player} = k \,|\, \text{profile } x) = \rho_x(k).$$

We assume that the revision law is *communicating* meaning that the probability of choosing any player is strictly positive over time. Without this assumption, the algorithm may clearly not converge to a NE in general.

Algorithm 2: Best Response Algorithm (asyncBRA) with random selections

foreach *player* $k \in K$ **do**
  $stop_k := false$
repeat
  Pick player $k \in \mathcal{N}$ using law ρ_x
  Select new action $\alpha_k := \mathsf{BR}_k(x)$
  $stop_k := \mathbf{1}_{\{\alpha_k = x_k\}}$;
  $x_k := \alpha_k$;
until $stop_1 \wedge stop_2 \wedge \cdots \wedge stop_n$;

Theorem 2 (asyncBRA converges to NE [13]). *For any BR or G-potential game $\mathfrak{G}$, Algorithm 2 converges in finite time, almost surely, to a Nash Equilibrium of $\mathfrak{G}$.*

Proof. This is a well known result, we only provide a sketch of the proof.

Clearly, when a NE is reached, the algorithm stops when each player have had the opportunity to revise its strategy.

Otherwise, a player has incentive to change its strategy, which will lead to a strict improvement of the potential. By characterization of BR-potential games, this state will never be visited again. Since the number of states is finite, the algorithm will reach a NE in finite time, almost surely. □

Let us now consider the algorithm BRA under a general revision protocol ρ that allows several players to change their strategy simultaneously. In that case, the revision law ρ is a distribution over sets of players, that, in full generality, depends on the current profile x:

$$\forall K \subset \mathcal{N}, x \in \mathcal{A}, \quad \mathbb{P}(\text{set of selected players} = K \,|\, \text{profile } x) = \rho_x(K).$$

The sets whose probability is positive define the *support of* ρ_x, denoted by $\mathcal{S}(\rho_x)$. In the following, we will assume that the support does not depend on the profile, and that each player has a chance to revise its strategy.

Assumption 2 (Constant support). The support of the revision law ρ,

(i) is constant wrt the profile x (hence is just denoted $\mathcal{S}(\rho)$):

$$\forall x, y \in \mathcal{A}, \mathcal{S}(\rho_x) = \mathcal{S}(\rho_y); \tag{2.6}$$

(ii) covers all players:

$$\forall x \in \mathcal{A}, \cup_{K \in \mathcal{S}(\rho_x)} K = \mathcal{N} \tag{2.7}$$

Algorithm 3: Best Response Algorithm BRA with general revision protocol ρ

```
foreach player k ∈ N do
    stop_k := false
repeat
    Pick a set of players K ⊂ N using law ρ_x
    foreach player k ∈ K simultaneously do
        Select new action α_k := BR_k(x);
        stop_k := 1_{α_k = x_k};
    x := α;
until stop_1 ∧ stop_2 ∧ ··· ∧ stop_n;
```

To keep notations simple, for every set K of players, we will denote by $\mathrm{BR}_K(x)$ the action profile obtained by simultaneous best responses for each player in K from x. Hence, $x' = \mathrm{BR}_K(x)$ means that, for all $k \in K, x'_k = \mathrm{BR}_k(x)$ and for all $j \notin K$, $x'_j = x_j$.

Chapter 3

Convergence under Simultaneous Play

When several players move simultaneously, the potential may not be decreasing and then the convergence of BRA to a NE of the game is not guaranteed as shown in the following example.

Example 1 (**No convergence to NE for simultaneous revision)**. Let us consider a 2-player 2-action (a and b) potential game with the following (exact) potential:

$$F = \begin{array}{|c||c|c|}\hline 1\backslash 2 & a & b \\ \hline\hline a & 0 & 2 \\ \hline b & 3 & 1 \\ \hline\end{array}$$

If the revision protocol always makes the two players play simultaneously ($\rho(\{1,2\}) = 1$) and if Algorithm 3 starts with action profile (a,a), then, during the run, both players keep changing their strategy simultaneously from (a,a) to (b,b) and back, and they never reach the two NEs (a,b) and (b,a).

In this chapter we will give a characterization of the revision sequences that ensure convergence on all games.

It should be clear that this characterization only depends on the support of the revision protocol ρ (namely which sets have positive probability under ρ) rather than on the actual values of the probabilities. Indeed, since players are selected at each round independently of the previous rounds, Borel-Cantelli lemma implies that any sequence of revision sets with positive probability will occur infinitely often regardless of the probability values.

3.1 Separability of the Revision Protocol

To state a necessary and sufficient condition for convergence to a NE, we need to introduce several definitions. A family $\mathcal{F}$ of sets of player is a *set cover* of $\mathcal{N}$ if $\cup_{K\in\mathcal{F}} K = \mathcal{N}$.

Definition 3 (**Separable family).** Let $\mathcal{F}$ be a set cover, consider the following iterative elimination process: as long as there is a singleton (say $\{k\}$) in $\mathcal{F}$, remove player k from all sets in $\mathcal{F}$. Then, $\mathcal{F}$ is *separable* if the elimination process reduces $\mathcal{F}$ to the empty set. A revision law ρ is *separable* if its support is separable.

Another way to state the separability property is the following: $\mathcal{F}$ is separable if and only if there is a permutation $(k_1, k_2, \dots k_N)$ of $\mathcal{N}$ such that the following sets all belong to $\mathcal{F}$:

$$\begin{array}{lcl} K_1 & = & \{k_1\}, \\ K_{i+1} & = & \{k_{i+1}\} \cup L_i, \ \forall i \in \{1, \dots, N-1\} \end{array} \tag{3.1}$$

where L_i is any (possibly empty) set included in $\{k_1, \dots k_i\}$.

Yet another characterization, similar to the previous one, but more compact, is that $\mathcal{F}$ contains N sets $K_1, \dots K_N$ such that, for all i,

$$K_i \setminus \cup_{j<i} K_j \text{ is a singleton.} \tag{3.2}$$

Obviously, if a family $\mathcal{F}$ is separable, then adding any set to $\mathcal{F}$ preserves separability. It should also be clear that if $\mathcal{F}$ is separable, then the family $\mathcal{F}'$, obtained by removing one player from all the sets in $\mathcal{F}$, is separable over the remaining $N-1$ players.

Example 2 (**Separable families and laws**).

- If $\mathcal{F}$ contains all the singletons, $\{k_1\}, \{k_2\}, \cdots, \{k_N\}$ then $\mathcal{F}$ is separable.
- If the family $\mathcal{F}$ contains the sets $\{k_1\}, \{k_1, k_2\}, \cdots, \{k_1, k_2, \dots, k_N\}$ then $\mathcal{F}$ is separable.
- The following revision law ρ is separable: Each player k_i chooses to play independently of the others, with some positive probability p_i. This revision law is separable because each singleton has a positive probability to be played. For all i,

  $$\rho(\{k_i\}) = p_i \prod_{j \neq i} (1 - p_j) > 0.$$

  This revision law is also fully distributed since it does not require any coordination between players.

We are now ready to state the main result of this section.

Theorem 4 (**Convergence to NE for separable revisions**). *Let $\mathcal{N}$ be a set of players and ρ be a revision law over $\mathcal{N}$. Algorithm BRA(ρ) converges a.s. to a NE for all BR-potential games $\mathfrak{G}$ over $\mathcal{N}$ if and only if ρ is separable.*

Proof. We prove the sufficient and necessary conditions separately.

Sufficient condition The sufficient condition is proved by contradiction.

Algorithm BRA(ρ) naturally induces a Markov process over $\mathcal{A}$ (up to adding a self-loop transition at each NE with probability one instead of stopping the algorithm). Let R be the set of recurrent action profiles, i.e. profiles that are visited infinitely often with probability one. All NE are recurrent since the algorithm stays there when a NE is reached. We are going to show that R is only made of NEs.

By contradiction, assume that there exists a BR-potential game together with a separable revision law ρ for which algorithm BRA(ρ) does not always converge to a NE. Let x be one recurrent action profile that is not a NE, with the highest potential. Starting from $X_0 = x$, let us consider a run of the algorithm using the sequence $K_1, K_2, K_3, \ldots, K_N$ of sets of players as defined in (3.1) (having positive probability by separability of ρ) which generates the sequence of profiles $X_1, X_2, \ldots, X_N$.

First notice that, if $X_i = x$ for all $i \in \{1, \ldots, j\}$, then players $k_1, \ldots, k_j$ already play a best response to x. In particular, j cannot be equal to N since this would imply that x is a NE.

So let i be the smallest index in $\{1, \ldots, N\}$ such that $X_i \neq x$. We have $X_i = \mathrm{BR}_{k_i}(x)$ since all other players in K_i already played a best response to x. This implies that the potential strictly increases along this deviation. Since X_i is in the same recurrent class as x and x is the recurrent non-NE with the highest potential, we deduce that X_i is a NE. But this contradicts the fact that x is recurrent since the algorithm will never visit x again.

Necessary condition The necessary part is proved by induction on the number of players.

First notice that if the game has a single player, the synchronous and asynchronous algorithms are identical, which implies the convergence in one step to the unique NE.

Let us assume that any revision law that guarantees a.s. convergence on any game with $n-1$ players is separable. This induction property is denoted $P(n-1)$.

We now consider a revision law ρ over a set $\mathcal{N}$ of players of size n such that for any game $\mathfrak{G}$ over $\mathcal{N}$, every execution of Algorithm BRA(ρ) converges a.s. to a NE. We will show that ρ is separable.

We first show that the support of ρ must contain a singleton. For that, let us construct a specific game over $\mathcal{N}$ with action set $\mathcal{A}_k = \{0, \ldots, p-1\}$ for every player k, where p is a prime number larger than n.

The payoffs are taken identical for every players and equal to the potential $F(x) = -(\sum_k x_k \mod p)$. The maximum of the potential is 0, which is reached when e.g. all players choose action 0.

The best response of any player k in a state with potential $-h$ is to choose action $(x_k - h) \mod p$. Therefore, when m players play simultaneously in a state with potential $-h$, the potential becomes $-((h + m(-h)) \mod p) = -((1-m)h \mod p)$. Starting from a state with potential -1 and using a sequence of revision sets with respective sizes

$m_1, \ldots, m_\ell, \ldots$ the successive values of the potential are:

$$\begin{aligned}
&-((1-m_1) \mod p),\\
&-((1-m_1)(1-m_2) \mod p),\\
&\vdots\\
&-((1-m_1)(1-m_2)\cdots(1-m_\ell) \mod p),\\
&\vdots
\end{aligned}$$

Since BRA(ρ) converges a.s. to a NE, this sequence reaches value 0 in finite time. Since p is prime, the only possibility is that there exists a revision set with size 1 in the support of ρ, say $\{k\}$. Hence, any revision law that satisfies property $P(N)$ must contain a singleton.

We are now ready for the second step of the proof that uses the induction assumption. We first construct a revision law ρ_{-k} over the set of players $\mathcal{N} \setminus \{k\}$ whose support is obtained by removing k from all the revisions sets in $\mathcal{S}(\rho)$. Now, we can claim that $BRA(\rho_{-k})$ converges in all games with $N-1$ players: Indeed, from any game $\mathfrak{G}$ over $N-1$ players, one can construct a game $\mathfrak{G}^+$ over N players by adding a dummy player (say k) with a single action that does not affect the utilities of the other players. Since $BRA(\rho)$ converges on all games with N players, it converges on game $\mathfrak{G}^+$. Since the added player does not play any role in $\mathfrak{G}^+$, this implies that the revision law ρ_{-k} converges on $\mathfrak{G}$. By the induction assumption, $P(N-1)$, ρ_{-k} is separable. By definition of separability, ρ is also separable. □

The notion of separability is game-independent. Given a specific game $\mathfrak{G}$, it is possible to define a new version of the separability property (called $\mathfrak{G}$-separability in the following) that depends on $\mathfrak{G}$ or more precisely on the interaction of players in $\mathfrak{G}$.

More formally, players k and ℓ do not interact in game $\mathfrak{G} = (\mathcal{N}, \mathcal{A}, u)$ if for every profile x, functions $u_k(x_\ell, x_{-\ell})$ and $u_\ell(x_k, x_{-k})$ are constant w.r.t. to, respectively, variables x_ℓ and x_k. In the opposite case, we say that players k and ℓ are *neighbors*. We consider the graph whose vertices are the players, and the undirected edges are defined by the neighbor relation. The vertices of this graph can be colored such that no neighbors have the same color. Notice that the number of colors is not unique, for example one can use one color per player whatever the graph.

Definition 5 (**$\mathfrak{G}$-separability).** Let us consider that the interaction graph of game $\mathfrak{G}$ is colored with a set $\mathcal{C}$ of colors. Let $\mathcal{H}$ be a separable family over $\mathcal{C}$. For each set C in $\mathcal{H}$, consider the set $\mathcal{N}(C)$ of all the players whose color is in C. The collection of all such sets $\mathcal{N}(C)$ for all C in $\mathcal{H}$ is called a $\mathfrak{G}$-separable family over $\mathcal{N}$.

It should be clear that, if all players are neighbors in $\mathfrak{G}$, $\mathfrak{G}$-separability coincides with separability, because all players must have a different color. It is also clear that separability implies $\mathfrak{G}$-separability for all games by coloring all players with distinct colors.

Example 3 (**$\mathfrak{G}$-separable, but not separable family**). Here is an example of a $\mathfrak{G}$-separable revision family that is not separable. Consider a game $\mathfrak{G}$ with 4 players such that 1 is a neighbor of 2, 2 is a neighbor of 3, and 3 is a neighbor of 4, with no other neighboring relations. The players can be colored with two colors, *Blue* for players 1 and 3 and *Red* for players 2 and 4. Consider the separable revision family $\mathcal{H}$ over the colors, made of two sets, $\{Blue\}, \{Blue, Red\}$. Now the family $\mathcal{F}$ is made of two sets $\{1, 3\}, \{1, 2, 3, 4\}$. By definition, $\mathcal{F}$ is $\mathfrak{G}$-separable but it is not separable because it does not contain any singleton.

The main (straightforward) property of games where two players (say k and ℓ) do not interact with one another is

$$\mathrm{BR}_{\{k,\ell\}}(\alpha) = \mathrm{BR}_k(\mathrm{BR}_\ell(\alpha)) = \mathrm{BR}_\ell(\mathrm{BR}_k(\alpha)). \tag{3.3}$$

In other words, letting k and ℓ play simultaneously or one after the other leads to the same state.

Corollary 6 (**$\mathfrak{G}$-separability implies convergence to NE**). *Let $\mathfrak{G}$ be a BR-potential game with N players and let ρ be $\mathfrak{G}$-separable. Then BRA(ρ) converges to a NE of the game.*

Proof. The proof is similar to the proof of Theorem 4.

Let us assume that there is a set of recurrent profiles R that does not contain any NE. Let x be a strategy with the largest potential in R. Starting from x, let us consider a trajectory of the algorithm using the sequence $K_1, K_2, \ldots, K_m$ of revision sets, ordered according to (3.1) for the associated separable family over the colors. The first time that the action profile becomes different from x (to a new profile y), the potential increases because of the order chosen on the sets. Since x is the non NE state with the largest potential in R, then $y \notin R$ unless it is a NE. Since x is recurrent, y cannot be a NE. And $y \in R$ because there is a path coming from the recurrent state x.

Therefore, applying the sequence of revision sets $K_1, \ldots, K_m$ starting in x, the algorithm remains in x. This implies that $x_k = BR_k(x)$ for all players $k \in \mathcal{N}$, so x is a NE of the game.

By contradiction, this implies convergence to NE of Algorithm 3. □

3.2 Extension to Smoothed Best Response

Theorem 4 says that Algorithm BRA(ρ) converges to a NE in potential games under the separability condition. This implies convergence to a local minimum of the potential. However its potential can be arbitrarily far from the global maximum.

To ensure convergence to an optimal NE (maximizing the potential), one can replace the greedy best response (used so far) by a smoothed best response allowing the algorithm to escape from local maxima.

We denote by $\mathbf{u}_k(x_{-k})$ the payoff vector of player k under action profile x: $\mathbf{u}_k(x_{-k}) = (u_k(\alpha, x_{-k}))_{\alpha \in \mathcal{A}_k}$.

A *random choice* Q is a random variable over $\mathcal{A}_k$, the actions of player k whose law only depends on the payoff vector $\mathbf{u}_k(x_{-k})$. This random choice is used to modify algorithm BRA(ρ) in the following way.

Algorithm 4: Smoothed BR algorithm with revision law ρ and random choice Q, SmoothBRA(ρ, Q)

```
repeat
    Pick a set of players K ⊂ 𝒩 according to ρ_x;
    foreach player k ∈ K simultaneously do
        Select action α_k := Q(u_k(x_{-k}));
    x := α;
until infinity;
```

Due to its random nature, and unlike BRA(ρ), this algorithm never ends. Its convergence properties will only be given in terms of its asymptotic distribution[1].

In the following, we will focus on the classical *logit choice*, parametrized by the inverse of the temperature θ. For each player k and profile x, the law is given by:

$$\mathbb{P}\left[Q(\mathbf{u}_k(x_{-k})) = \alpha\right] = \frac{\exp(\theta u_k(\alpha, x_{-k}))}{\sum_{\beta \in \mathcal{A}(k)} \exp(\theta u_k(\beta, x_{-k}))}.$$

Note that the logit choice is a close approximation of the BR mapping when θ goes to infinity.

Under this random choice, the sequence $(X_n)_{n\in\mathbb{N}}$ of action profiles computed by SmoothBRA(ρ, Q) forms a Markov chain whose transition matrix can be constructed as follows.

Let $\mathrm{Diff}(x, y) \stackrel{\text{def}}{=} \{k : x_k \neq y_k\}$ be the set of players that must have played if the sequence $(X_n)_{n\in\mathbb{N}}$ jumps from $X_n = x$ to $X_{n+1} = y$ in one step. Of course, the set of players that actually played can be larger because some of them may have chosen to not change their action. This is the reason why we introduce the intermediate matrix P^V, defined for all sets V of players by:

$$P^V_{x,y} = \begin{cases} \displaystyle\prod_{k \in V} \frac{\exp(\theta u_k(y_k, x_{-k}))}{\sum_{\alpha \in A(k)} \exp(\theta u_k(\alpha, x_{-k}))} & \text{if } \mathrm{Diff}(x, y) \subseteq V \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

[1]For practical purpose, one needs to stop SmoothBRA(ρ, Q) after a finite number of iterations. Many heuristic stopping rules have been proposed (see [11] for example), but in practice there is a fair amount of "black magic" involved and no single stopping rule provides guarantees in the general case.

Then the transition matrix P is

$$P_{x,y} = \sum_{V \supseteq \text{Diff}(x,y)} \rho_x(V) P^V_{x,y}.$$

The asymptotic behavior of algorithm SmoothBRA(ρ, Q) is given by the stationary distribution π of this ergodic Markov chain on $\mathcal{A}$.

The well-known Markov chain tree theorem provides an explicit formula for π (up to a multiplicative factor), based on spanning trees over the Markov chain transition graph.

Theorem 7 (**Markov Chain Tree Theorem [10]**). *Let $\mathcal{T}_x$ be the set of spanning in-trees of the transition graph, with root in x. The stationary probability π_x is proportional to the sum of the probability weights of all the spanning trees T in $\mathcal{T}_x$:*

$$\pi_x \propto \sum_{T \in \mathcal{T}_x} \prod_{(y,z) \in T} P_{y,z}.$$

The stationary distribution π puts a positive probability on all action profiles x, meaning that all action profiles will be visited infinitely often during the execution of SmoothBRA(ρ, Q). However, when θ goes to ∞, the probability mass will concentrate on some profiles. Such profiles are called *stochastically stable* under ρ.

Stochastic stability can be asserted using the following lemma, based on the *orders* (w.r.t. θ) of the transition probabilities. When θ goes to ∞, $\varepsilon \stackrel{\text{def}}{=} \exp(-\theta)$ goes to 0.

Let us express the transition probabilities as a function of ε instead of θ:

$$P_{x,y} = \sum_{V \supseteq \text{Diff}(x,y)} \rho_x(V) \prod_{k \in V} \frac{\varepsilon^{-u_k(y_k, x_{-k})}}{\sum_{\alpha \in A(k)} \varepsilon^{-u_k(\alpha, x_{-k})}}. \tag{3.5}$$

which can be written under the following first order development w.r.t. ε:

$$P_{x,y} = c_{x,y} \varepsilon^{q_{x,y}} + o(\varepsilon^{q_{x,y}}),$$

where $q_{x,y}$ is called the *order* of $P_{x,y}$ w.r.t. ε (or, equivalently, w.r.t. θ).

Lemma 8 (**Stochastic stability characterization [2]**). *State x is stochastically stable if and only if the order of its minimal in-tree is the smallest, among all in-trees.*

Proof. This characterization has first been given in [2]. The following proof is similar to the original proof. It is helpful to detail it here to further highlight the notion of orders, used in the rest of the chapter.

Using Equation (3.5), for any pair of states x and y, one can compute the order $q_{x,y}$ as

$$q_{x,y} = \min_{V \supseteq \text{Diff}(x,y) \cap \mathcal{S}(\rho)} \left(\sum_{k \in V} \left(\max_{\alpha \in \mathcal{A}_k} u_k(\alpha, x_{-k}) - u_k(y_k, x_{-k}) \right) \right). \tag{3.6}$$

Note that the orders of the transitions do not depend on the values of ρ but only of its support. Therefore, the limit value of π only depends on $\mathcal{S}(\rho)$, as well. By using the Markov chain tree theorem (Theorem 7), the order q_x of π_x w.r.t. ε is

$$q_x \stackrel{\text{def}}{=} \min_{T \in \mathcal{T}_x} \sum_{(y,z) \in T} q_{y,z}$$

Therefore, the only components in π that do not go to 0 when ε goes to 0 are those with the smallest order:

$$(\lim_{\theta \to \infty} \pi_x > 0) \Leftrightarrow (q_x = \min_{y \in \mathcal{A}} q_y). \tag{3.7}$$

□

Equation (3.6) provides an explicit formula of the order of the transition probability from x to y. In particular, one may observe that the order is always non-negative, and equal to zero if and only if there is a revision set that only involves best responses of the players in that set for going from x to y.

The next theorem states that the total probability mass of the NEs goes to one when θ goes to ∞. This implies that Algorithm SmoothBRA(ρ, Q) will only visit NE with a high probability when θ is large.

Theorem 9 (Convergence to NE). *Let $\mathfrak{G}$ be a BR-potential game. If ρ is $\mathfrak{G}$-separable, then for all action profiles x that are not NE, the stationary probability π_x goes to 0 as θ goes to ∞.*

Proof. The proof is based on Lemma 8 that allows to characterize the states with the smallest order. From the remark following the lemma, $q_{x,y}$ is non-negative. It is equal to 0 if and only if there is a set V in $\text{Diff}(x, y) \cap \mathcal{S}(\rho)$ such that $y = \text{BR}_V(x)$.

From Corollary 6, if ρ is $\mathfrak{G}$-separable, and starting from any action profile x that is not a NE, there is a finite sequence $(V_n)_{0 \leqslant n < H}$ of sets of players in $\mathcal{S}(\rho)$ and actions profiles $(X_n)_{0 \leqslant n \leqslant H}$ such that

$$\begin{aligned} X_0 &= x \\ X_{n+1} &= \text{BR}_{V_n}(X_n), \quad \forall\, 0 \leqslant n < H, \end{aligned}$$

and X_H is a NE. Using these sequences for all x constructs a path whose order is 0, leading to the NE X_H.

Let T_x^* be the tree with minimal order, rooted in x. From T_x^*, it is possible to construct a tree rooted in X_H by adding the path from x to X_H with order 0 and removing the arc in T_x^* starting in X_H. This arc has a strictly positive order because X_H is a NE. The new tree has an order strictly smaller that T_x^*, so x cannot achieve the minimum in (3.7). Therefore, only NE may have positive stationary probabilities. □

Using the same construction, one can show more generally that any stochastically stable state is included in the recurrent set of algorithm SmoothBRA(ρ, Q).

This result is to be compared with [2] where a similar result is proved. It seems however that the absence of separability and of Assumption 1 in [2] jeopardizes their result.

A stronger result can be proved for revisions that do not allow for simultaneous revisions in the case of exact potential games.

Theorem 10 (**Convergence to optimal NE for asynchronous revisions).** *Let $\mathfrak{G}$ be a game with an exact potential F. If the revision law does not contain any simultaneous play (in other words, $\mathcal{S}(\rho) = \big\{\{k\}, k \in \mathcal{N}\big\}$) then the only stochastically stable profiles are the optimal NE.*

Proof. The proof of this result is a direct consequence of the results in Chapter 12 of [19]. We provide a short proof in the sake of completeness. First note that the stochastically stable profiles do not depend on the actual values of $\rho(\{k\})$, as mentioned before. Let us consider the case where all of them are equal: $\rho(\{k\}) = 1/N$ for all $k \in \mathcal{N}$. In that uniform case, the Markov chain (X_n) is reversible and the stationary probability is explicitly known: for all profiles x, $\pi_x \propto \exp(\theta F(x))$. Therefore, when θ goes to ∞, the total stationary probability of the profiles with optimal potential will go to one. □

3.3 Examples

The following examples show that there is little hope to prove more precise results, at least in the general case.

Example 4 shows that Theorem 9 is not true when the revision protocol is not $\mathfrak{G}$-separable. This example shows a game where SmoothBRA has stable non-NE points under a non-separable protocol.

Example 5 shows that Theorem 14 is not true if all players can play alone but simultanenous plays are also allowed. This example has 3 players and it is easy to find examples with any number of players larger than 3. However with two players, Theorem 11 says that if both players can play alone then simultaneous play will not jeopardize convergence to an optimal NE. Example 6 shows that Theorem 11 is not true for all separable protocols: if only one player among the two can play alone, then convergence to the optimal NE is not guaranteed.

Finally, Example 7 shows that Assumption 1 cannot be relaxed in Theorem 9 by exhibiting a game with two players (where both can play alone) that admits stable states that are not NE.

Example 4 (**No convergence to NE for a non-separable revision law**). Let us consider a 2-player game with 2 actions each, $\mathcal{A} := \{a, b\}$. The support of the revision law is made of a single set $\{1, 2\}$ (both players always play together). This revision law is not separable.

The payoffs of both players coincide with the potential, given by the following matrix:

$$F = \begin{array}{|c||c|c|}\hline 1\backslash 2 & a & b \\ \hline\hline a & 1 & 0.5 \\ \hline b & 0 & 1 \\ \hline\end{array}$$

This game has two NE (a, a) and (b, b). The order of all the transition probabilities are given in Figure 7.3, computed using (3.6).

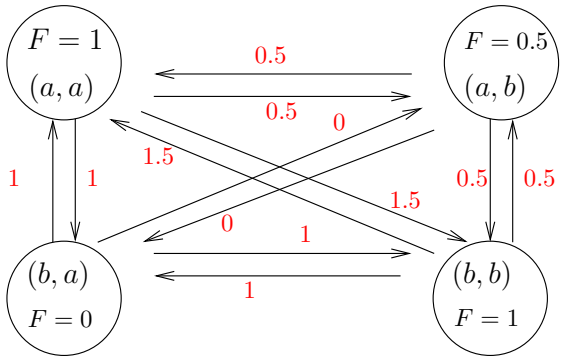


Figure 3.1: The orders of all the arcs in the transition graph of a 2-player potential game, when the revision process always makes both players play simultaneously

The minimal tree with root in (a, a), $T^*_{(a,a)}$ can be computed from Figure 7.3. Its order is $q_{(a,a)} = 1$. The minimal tree with root in (b, b), $T^*_{(b,b)}$, can also be found from Figure 7.3, with order $q_{(b,b)} = 1$. The minimal trees for the profiles (a, b) and (b, a) are both of order 1 as well: $q_{(a,b)} = 1$ and $q_{(b,b)} = 1$. Therefore, the stochastically stable state of SmoothBRA(ρ, Q) are all the states, (NE as well as non-NE). The algorithm will visit non-NE states with probabilities that do not vanish when the parameter θ becomes large. Actually, one can also compute the exact stationary distribution for all θ: $\pi((a, a), (a, b), (b, a), (b, b))$ is proportional to:

$$(s^4 + s^3 + es^2 + es, s^4 + 2s^3 + s^2, s^4 + 2es^2 + e^2, s^4 + s^3 + es^2 + es),$$

where $s \stackrel{\text{def}}{=} e^{\theta/2}$. Its limit when the parameter θ goes to infinity is $(1/4, 1/4, 1/4, 1/4)$ so all states are uniformly selected regardless of their potential. Even more surprisingly, notice that when θ is larger than 2, the state with the largest probability is (a, b), which is not a NE.

Example 5 (**Convergence to non-optimal NE when players can play alone**). Let us consider a 3-player game $\mathcal{N} = \{x, y, z\}$ with 2 actions each, $\mathcal{A} := \{0, 1\}$. The support of the revision law is the separable family $\{x\}, \{y\}, \{z\}, \{x, y, z\}$.

The payoffs of the players coincide with the potentials, given by: $F(0, 0, 0) = 3$, $F(1, 0, 0) = 0$, $F(0, 1, 0) = 1$, $F(0, 0, 1) = 2$, $F(1, 1, 0) = -10$, $F(0, 1, 1) = -11$, $F(1, 0, 1) = -12$, $F(1, 1, 1) = -1$.

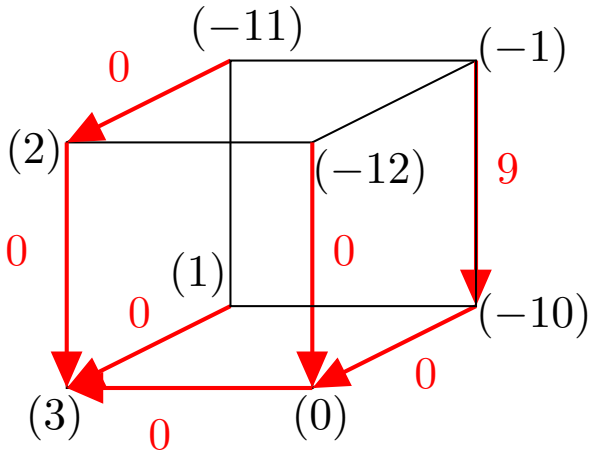


Figure 3.2: The minimal in-tree rooted in $(0,0,0)$ has order $q_{(0,0,0)} = 9$ (the potentials are given in parenthesis and the orders of the transitions are in red).

This game has two NE $(0,0,0)$ and $(1,1,1)$. The optimal one, with maximal potential is $(0,0,0)$. The minimal tree with root in $(0,0,0)$, $T^*_{(0,0,0)}$, is of order $q_{(0,0,0)} = 9$ (see Figure 3.2) and does not use the revision set $\{x, y, z\}$.

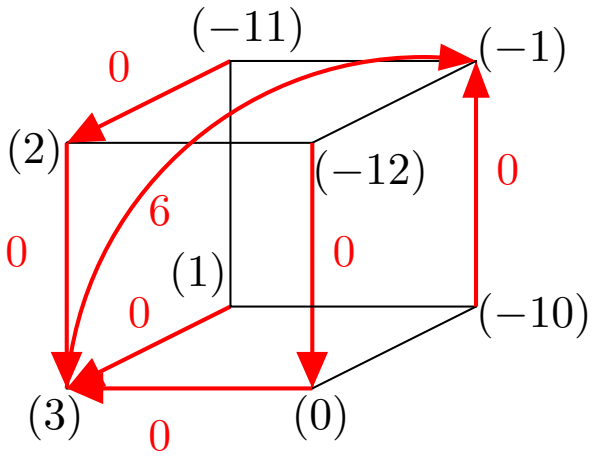


Figure 3.3: The minimal in-tree rooted in $(1,1,1)$ has order $q_{(1,1,1)} = 6$

The minimal tree with root in $(1,1,1)$, $T^*_{(1,1,1)}$, is of order $q_{(1,1,1)} = 6$ (see Figure 3.3) and uses the revision set $\{x, y, z\}$ to jump from $(0,0,0)$ to $(1,1,1)$ with order $1+2+3 = 6$, so it is minimal.

Therefore, the only stochastically stable profile of SmoothBRA(ρ, Q) will be $(1,1,1)$, whose potential is not optimal.

As mentioned before, convergence to optimal NE can be proved in the case of two

players who can play alone:

Theorem 11 (Convergence to optimal NE with two players). *Let $\mathfrak{G}$ be a two-players game with exact potential F. If the support of the revision process is $\{\{1\},\{2\},\{1,2\}\}$ then the only stochastically stable states are optimal NE.*

Proof. We will first prove that there exists a tree with minimal order ending in each NE whose arcs only use singletons as revision sets.

Let us notice that, from any state that is not a NE, there is an outgoing arc of minimal order (0), that uses a unilateral deviation. In the case where there are only two NE x^* and y^* (the general case being treated similarly by induction), the non-NE states are all covered by two sets S_1 and S_2: those connected to x^* (resp. y^*) with paths of order 0 that involves single players (some states may belong to both sets).

Now, to get the minimal tree rooted in x^*, only one path is missing, from y^* to one state in S_1. Let us assume that this path contains one diagonal arc (that involves two players), from state u to state v. With no loss of generality, the next vertex after v in this path, denoted w, shares the same action for player 1: $v_1 = w_1$.

The order of the arc from u to v is $F(u_1, \mathrm{BR}_2(u_1)) - F(u_1, v_2) + F(\mathrm{BR}_1(u_2), u_2) - F(v_1, u_2)$, and the order of the arc from v to w is $F(v_1, \mathrm{BR}_2(v_1)) - F(w)$.

Let us replace the path $u \to v \to w$ by the path from $u \to (v_1, u_2) \to w$ that does not contain any diagonal arc. The order of the arc from u to (v_1, u_2) is $F(\mathrm{BR}_1(u_2), u_2) - F(v_1, u_2)$, and that of the arc from (v_1, u_2) to w is $F(v_1, \mathrm{BR}_2(v_1)) - F(w)$. Summing both values gives a path with smallest order.

As for node v, which is no longer on the path, it is in set S_2, so that it belongs to a path with order 0 to y^*. As for the order for all other states, it is unchanged.

In total, the new tree has a smaller order as the previous one, so that all diagonal arcs can be removed on the minimal tree.

Therefore, there exists a minimal tree rooted in x^* that only uses single revision sets. Now, Theorem 14 says that a minimal tree with single revision sets rooted in an optimal NE has the smallest order. □

The condition on the revision process that all players can play alone, used in Theorem 11, is stronger than separability. Example 6 shows that separability is not enough to guarantee convergence to optimal NE, even with two players.

Example 6 (**Convergence to non-optimal NE for two separable players**). Let us consider the following 2-player game with respective actions $\mathcal{A}_1 = \{a, b\}$ and $\mathcal{A}_2 = \{a, b, c\}$. The support of the revision law is $\{\{2\},\{1,2\}\}$, hence it is separable.

The payoffs of both players coincide with the potential, given by the following matrix:

$F =$

1\2	a	b	c
a	11	0	5
b	5	10	8

This game has two NE (a,a) and (b,b). The minimal tree with root in (a,a) has order 7. Indeed, the path with smallest order to join (b,b) to (a,a) is $(b,b) \to (b,c) \to (a,c) \to (a,a)$ whose order is $2+5+0$. All other states can be added to this path with the order 0 thanks to unilateral best response of y. The minimal tree with root in (b,b) has order 6. Indeed, the path with the minimal order to join (a,a) to (b,b) is $(a,a) \to (b,a) \to (b,b)$ whose order is $6+0$. All other states can be added to this path with order 0 thanks to unilateral best response of y.

Finally, in the same scenario as in Theorem 11 (two players and a revision process that contains both singletons), even worse things can occur when NE are not strict. Non strict NE can only exist when the best response is not unique. This possibility has been discarded up to now in this chapter (Assumption 1). In the next example, we consider a case where the NE are not strict so that Assumption 1 is violated.

Example 7 (**No convergence to NE with two separable players if NE are not strict**). Consider a separable revision process, where both players can play alone, with support $\{1\},\{2\}$ and $\{1,2\}$ and a game with two actions per player given by the potentials (payoffs are equal to potentials):

$$F = \begin{array}{|c||c|c|}\hline 1\backslash 2 & a & b \\ \hline\hline a & 1 & 1 \\ \hline b & 1 & 0 \\ \hline\end{array}$$

States $(a,a),(a,b),(b,a)$ are non-strict NE and all states (including (b,b)) have a minimal in-tree of order 0, so even when the temperature $1/\theta$ goes to 0, the non-NE state (b,b) has a non-vanishing probability of being chosen. More precisely, by computing the exact stationary distribution π for all temperatures, one can check that when θ goes to infinity, $\pi((a,a),(a,b),(b,a),(b,b)) \to (36/79, 20/79, 20/79, 3/79)$ if the revision sets $\{1\},\{2\}$ and $\{1,2\}$ are chosen uniformly.

Chapter 4

Average Convergence Speed

4.1 Motivation

To get an idea of how long a given algorithm will take on a given class of problem or what memory requirement it will need, the most common solutions deal with the cost of the worst instances . These method are sensibles in that they offer a bound that will be applicable without needing other hypothesis, but in some cases the information they give is not helpful for practical usage.

Finding a Nash equilibria in a potential game is PPAD-complete (as shown in [4]) , this means that the best the worst case approach in analyzing any algorithm can do bring an exponential result. This would imply that best response is merely the tool of proof it was introduced at, as the trivial naive algorithm which simply test every state is also exponential.

In the current chapter, I propose an analysis in which i suppose that the game is chosen at random instead of averaging on the execution of the algorithm. This make the assumption that the problem on which the algorithm is used has no hidden structure, or that we already used every known information to optimize the dynamics, which would bring us back to close to an uniform distribution.

If the true but unknown distribution of the problem is not uniform, the following analysis still apply unless the distribution put a huge weight on a small subset of the set of possible games.

One could an approach and analysis similar to this chapter to obtain the probability for the algorithm to finish before a given time of any other metric needed for a particular use.

We recall algorithm 1 finding a Nash equilibrium under any revision sequence (R).

Algorithm 5: BRD under revision sequence R

Input: Game utilities $(u_k(\cdot))$; Initial state $(\mathbf{x} := \mathbf{x}(0))$; revision sequence R;
Initialize $t := 0$; List of satisfied customers $L := \emptyset$;
while *size(L)*$\neq n$ **do**
    Pick next player $k := R_t$; $t := t + 1$;
    if $x_k \notin BR_k(x)$ **then**
        Update strategy for player k to $x_k \in \mathrm{BR}_k(\mathbf{x})$;
        $L := \emptyset$;
    $L := L \cup \{k\}$;

4.2 Worst case analysis

Before starting the average time analysis, lets build an example of a game that need the maximal number of steps, A^{n-1}:

Let us construct a game for which there exists a starting point that makes the best response algorithm take nA^{n-1} steps before reaching a Nash equilibrium (NE). The game is built by induction on the number of players n.

With 1 player, the game takes one step (when the starting point is not a NE). With two players, it is possible to construct a game where the sequence of visited states is $(0,0),(1,0),(1,1)\cdots(i,i),(i+1,i),(i+1,i+1)\cdots(A-1,A-1),(0,A-1)$, by choosing the respective potentials of these states to be $0,1,\cdots(i,i)\to 2i,(i+1,i)\to 2i+1$ while the potential of all the other states is -1.

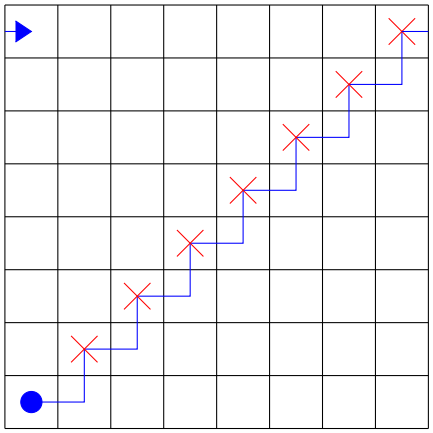


Figure 4.1: States visited by a round-robin sequence with two players.

This takes $2A$ steps before reaching a NE. For 3 players, let us assume that during the first $3A$ steps, the behavior is the same as for two players (with one additional dimension): on players 2 and 1 turns, they add 1 (mod A) to their strategy. The visited states are $(0,0,0),(1,0,0),(1,1,0),\cdots,(i,i,0),(i+1,i,0),(i+1,i+1,0),\ldots,(A-1,A-1,0),(0,A-1,0)$. In this sequence, at player 3's turns, namely in states of the form $(i,i,0)$, its best response is to stay in the same state (states marked with a red cross in Figure 4.1. Once state $(0,A-1,0)$ is reached, player 2's best response is to stay still and player 3 prefers to move to its next state: $(0,A-1,1)$. From this point on, players 1, 2 and 3 repeat the same stair pattern as before. Note that the staircase is shifted by one unit to the right (mod (A,A)). The states where player 3 does not move are now of type $(i+1,i,1)\mathrm{mod}(A,A)$. This construction is repeated with staircases of type $(i+j,i,j)$ for all $0\leqslant j\leqslant A-1$. The total number of steps is $3A^2$. Again, this can be obtained by choosing as potential -1 for all non visited states, and the order of visit for the visited states.

Let us generalize this construction for n players. During the first nA^{n-2} steps, the sequence of visited states is the same as with $n-1$ players. At each turn of player n, its strategies is to stay still. Once all these steps have been taken, The construction is shifted by one and repeated with sequences of type $(i_2 + i_3 + \cdots + i_{n-1} + j, i_2, \ldots, i_{n-1}, j) \bmod (A, \ldots, A)$. In total the number of steps is nA^{n-1}. Therefore the worst case complexity is bounded from below by nA^{n-1}.

Let us now show that this lower bound is tight. If the algorithm has not reached a NE then one player (say ℓ) has not played more than once in each "line" (set of states of type $(\cdot, x_{-\ell})$. There are A^{n-1} such lines and under round-robin revision sequences, player ℓ turns come every n steps, so that the number of steps before convergence is bounded from above by nA^{n-1}.

4.3 Randomization

4.3.1 Randomizing the Potential

The goal of this part is to define a random distribution over the games of given dimension that represents well the set of possible games and thus could be applied directly for problem with no or few priori.

Since most variant of best response dynamics will only use the values of the potential function when comparing two of them, the actual values can be summarized by their order. If we assume that there are no equality of potential, then choosing identically and independently the potential value of each configuration using any continuous distribution is equivalent in regard to the behavior of the dynamics.

This method gives a uniform choice among every possibles orders (thus every possible behavior of best response dynamics on potential games), allow for local and partial description and numerical analysis. In the different analysis, proofs and simulations we choose every potential independently and uniformly in $[0, 1]$

4.3.2 Randomizing Utilities

One could remark that the previous choice makes the false hypothesis that games from real problems will be evenly distributed among the possibles potential orders. This is true for game directly defined by their potential (example, when every player knows and want to maximize a common goal, or when the utilities are imposed by a designer exterior to the game for a precise goal), but most games are defined by the utilities of the players which do not have to be homogeneous.

To represent the differences between players we could use the following model : Assuming there is an arbitrary set of real continuous distributions, Every players choose one distribution in the set and all his utilities according to this distribution.
If the game is of potential, we keep it, else we try again.

This model only makes the assumption that the actuals utilities of each player are independent of the values of utility of the other (no player is altruistic or explicitly want to decrease other utility).

We can notice that the probability that the chosen game is of potential is near zero. One criterion for a game to be of potential is that all sum of variations of utilities along cycles of size four is null.

For each such cycle, this test only use four positives values and four negatives values chosen from the same two utilities law. This selection affect each distribution in a similar way, increasing further the weight on probable values and decreasing it on rare ones . This in turn means that the reject does not affect the order of the potentials.

Now, if we use the weakest definition of potential game : (see 2.3) the existence of a function $F : \mathcal{A} \to \mathbb{R}$ such that for any player k and action profile x

$$\mathrm{BR}_k(x) = \left\{ \underset{\alpha \in \mathcal{A}_k}{\operatorname{argmax}} F(\alpha, x_{-k}) \right\}. \tag{4.1}$$

then what matters is the coordinate of the maximum in each direction, which stay the same if we normalize the distributions.

After normalization, we can see that the difference of potential between two point can be written as the sum of a differences between one pair of independent random variables per directions of different coordinates. This is only dependent on the number of changed coordinate, and the expectancy of the absolute value of the difference is increasing with the distance.

Compared to the previous model, there is a weak bias that closer state for the hamming distance (pondered by the variance of the distribution if we skip normalization) will be closer in the global order.
This form of smoothness has two opposing effects : reducing slightly the number of Nash equilibria and accelerating the increase of current potential when following a trajectory as the new states are more likely to be neighbor to other high potential states.

In practice, those two effect are small and would be dwarfed by any property of the problem modeled by a game.

I choose to use the simpler independent potential model.

4.4 Intersections

The main feature preventing the algorithm to behave in a Markovian way are the visited states. Calling intersection the event when the algorithm re-reads a known state, we introduce in this section the intersection free approximation (IFA) that will allow us to use a Markov chain in the complexity analysis.

4.4.1 Intersection Free Approximation

The direct analysis of the behavior of BRD over a random potential is difficult because, over time, more and more states have been visited by the algorithm. Thus, its behavior is non-homogeneous in time. To avoid this difficulty, we introduce the intersection free algorithm.

Under the Intersection Free Approximation, every time a new player who is not satisfied (in the algorithm, that correspond to a player not in L) has to compute its best response in a state (say $\mathbf{x}$), it compares $\Phi(\mathbf{x})$ with the potential of its a other strategies randomly re-generated, whether or not those a states have been visited during the previous steps of the algorithm.

The IFA is given in Algorithm 6. The difference with Algorithm 9 is that every time a player plays, it generates the potential of its a alternative actions before computing its best response.

Algorithm 6: IFA Algorithm (for player k)

Input: Initial state ($\mathbf{x} := \mathbf{x}(0)$);
Local clock, ticking w.r.t. a Poisson process with rate λ/n;
List $L_{IFA} := \emptyset$;
while *size(L)*$\neq n$ **do**
  On *each tick of the local clock*
    if $k \notin L_{IFA}$ **then**
      forall $u \neq x_k$, **Generate** $\Phi(u, \mathbf{x}_{-k}) \sim$ Unif$[0, 1]$;
      if $x_k \neq BR_k(\mathbf{x})$ **then**
        $x_k := \mathrm{BR}_k(\mathbf{x})$; $L_{IFA} := \{k\}$
      $L_{IFA} := L_{IFA} \cup \{k\}$

To illustrate the difference between BRD and IFA, let us consider a simple two-player game with four actions per player, with a random potential. Such a potential is given in Figures 4.2 and 4.3. They respectively show the behavior of BRD and IFA under the revision sequence 1,2,1,2.

Under BRD, the potential of all states can be generated at the beginning, and the game is played under this fixed value of the potential (here uniform in $\{0, 1, \ldots, 10\}$).

Under IFA, potentials are generated each time a new player acts.

The first difference between the two systems occurs during the second play of player 2: one of the states that it can opt for has already been visited in the past in BRD. The potential of that state is known to be smaller than the current potential. Under IFA, the potential of this state is redrawn a second time, independently of the rest and can lead to a higher potential, as in Figure 4.3. Let V be the number of visited states whose utility (or potential in the case of IFA) has been evaluated during the execution.

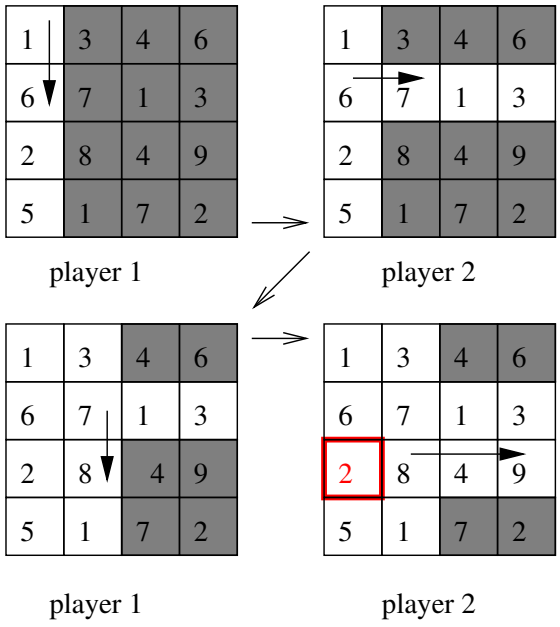


Figure 4.2: Evolution of the potential during BRD, under the revision sequence 1,2,1,2. The grey states have not been visited yet. When the second player plays for the second time, one state (in red) has already been visited.

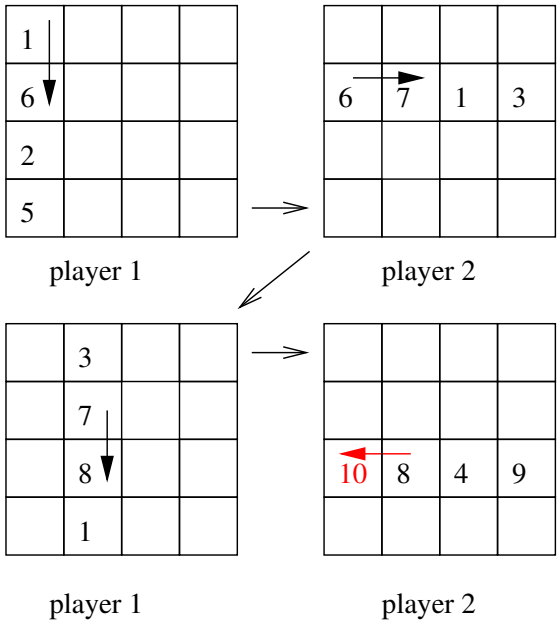


Figure 4.3: Evolution of the potential under IFA with the sequence 1,2,1,2. When the second player plays for the second time, the potential of state in red is redrawn and can become larger then the current potential.

In order to use this approximation to bound the complexity of the best response dynamics, we use this lemma :

Lemma 12 (Comparison with IFA)**.**

(i) The number V_{BRD} of states visited by BRD is smaller than the number V_{IFA} of states visited by the IFA approximation, for the stochastic order.

(ii) The expected number of steps to reach convergence for best response ($\mathbb{E}S_{BRD}$) and IFA ($\mathbb{E}S_{IFA}$) verify: $\mathbb{E}S_{BRD} \leq \mathbb{E}S_{IFA} + \frac{I}{a}$
where I is the expected number of states whose potentials are forgotten and chosen again by IFA and a is the number of alternative choices of a player (ie $a = A - 1$).

As an important note, this lemma does not use the fact that the algorithm use round robin, nor the fact that players act one after the others. This will allow us to reuse this approximation in every variants.

Proof of Lemma 12 i

The goal of this section is to prove the bound

$$V_{BRD} \leq V_{IFA}$$

We start by the construction of a coupling of BRD and IFA through the potentials of states newly visited: For both algorithms, we use the same infinite sequence, chosen in advance with values chosen independently according to the distribution $U([0, 1])$.

In both executions, each new potential used (or visited) by the algorithm takes the next value in the sequence U.

The coupling create a bijection preserving the measure between the space of potential games for a given number of players and choices per player as a bounded part of a vectorial space, and the set of infinite IID sequences with the probability measure.

The stopping condition in IFA is the presence of an consecutive values lower than the previous maximum. In the execution using best response dynamics , this would mean that in at least n directions, every values of potential is lower than the current value : this prove that the current point is a Nash equilibrium. If the algorithm had not yet converged, then it does now. This means that on every pair of the coupling, the number of new states visited by BRA is at most equal to the number visited by IFA. Since we used a bijection preserving the measure, this inequality is true stochastically on all games.

Proof of Lemma 12 ii

The goal of this section is to prove that $\mathbb{E}S_{BRD} \leq \mathbb{E}S_{IFA} + \frac{I}{a}$.

The number of new potentials visited per step under IFA is exactly a. The number of potential visited per step by BRA is also a, but some can be visited twice. The

number of new potentials visited by a step is a minus the number of intersections between this step and a previous one. The total number of new potentials visited by BRD is $V_{BRD} = aS_{BRD} - I.$ intersections.

Using the first part of the lemma, we obtain, for each coupled pair:

$$S_{BRD} = \frac{V_{BRD} + I}{a} \leq \frac{V_{IFA} + I}{a} = S_{IFA} + \frac{I}{a}$$

This being true for each pair in a coupling preserving the measure, it is true in expectation, hence

$$\mathbb{E}S_{BRD} \leq \mathbb{E}S_{IFA} + \frac{\mathbb{E}I}{a}$$

4.4.2 Number of Intersections

This section is devoted to studying the number of intersections.

We will prove the formula

$$\mathbb{E}I \leq \frac{3}{a(n-3)}\mathbb{E}(M^2) + \left(4 + (4c_1 + 1)\frac{n}{n - 4c_2}\right)\mathbb{E}(M). \tag{4.2}$$

where I is the expected number of intersections and M is the expected number of moves of the algorithm.

We first notice that the behavior of the algorithm can be cut into two parts : how long does the state stay the same, and what happen when the state change.

It is easy to see when looking at IFA, with one part affecting the number of player satisfied, and the other concerning the value of the potential, but it also apply to the dynamics.

Two steps can only cause an intersection if their starting points have exactly two coordinates different (we will say that their Hamming distance is 2), corresponding to their respective directions of play. In addition, two state in those positions can only create two intersections.

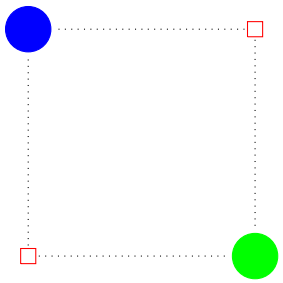


Figure 4.4: Two players with exactly two differing coordinates can have two intersections.

If we bound that by assuming every time the algorithm get into a state, all direction are tested, we only need to count the number of time the algorithm move to a state at a hamming distance two from another already visited.

Thus rewrite the intersection as a sum over the number M of moves (number of state changes during the execution of the algorithm) of the amount of following step at distance 2:

$$I = \sum_{i=0}^{M-1} I(i).$$

$I(i)$ is the number of intersections of the current state $\mathbf{x}(i)$ (reached after i moves) with all future states, visited at moves $i+1, i+2, \ldots, M$.

Let $d(\mathbf{x}, \mathbf{y})$ be the Hamming distance between states $\mathbf{x}$ and $\mathbf{y}$, namely the number of different coordinates.

By conditioning on the value of $M = m$ (the conditional expectation is denoted $\mathbb{E}^{(m)}$),

$$\mathbb{E}^{(m)} I = \mathbb{E}^{(m)} \sum_{i=0}^{m-1} I(i) = \sum_{i=0}^{m-1} \mathbb{E}^{(m)} I(i) \leqslant \sum_{i=0}^{m-1} 2\,\mathbb{E}^{(m)} \sum_{j=i+1}^{m-1} \mathbf{1}_{(d(\mathbf{x}_i, \mathbf{x}_j)=2)}. \tag{4.3}$$

We study $H(k,i) := \sum_{j=i+1}^{m-1} \mathbf{1}_{(d(\mathbf{x}_i, \mathbf{x}_j)=2)}$, namely the number of times the Hamming distance is 2. The evolution of the Hamming distance depends the moves. Let us denote by k_t the player that was active at the t-th move. The Hamming distance evolution over the moves depends on the current Hamming distance and on whether the k_t-th coordinate of $\mathbf{x}_i$ is equal or not to the k_t-th coordinate of the current state.

4.4.3 Markovian Model for the Intersections

Lets first make a remark on the form of the Markov chain modeling IFA : We can see it as two chains, one makes the number of satisfied players evolve and only depends on the potential without changing it. It determine the time between two moves which are characterized by a return to a state of the form $(1, x)$. These returns are stopping times.

The second chain acts once each time the number of satisfied players fall back to one. It determine which player changed its choice and the evolution of the value of the potential.

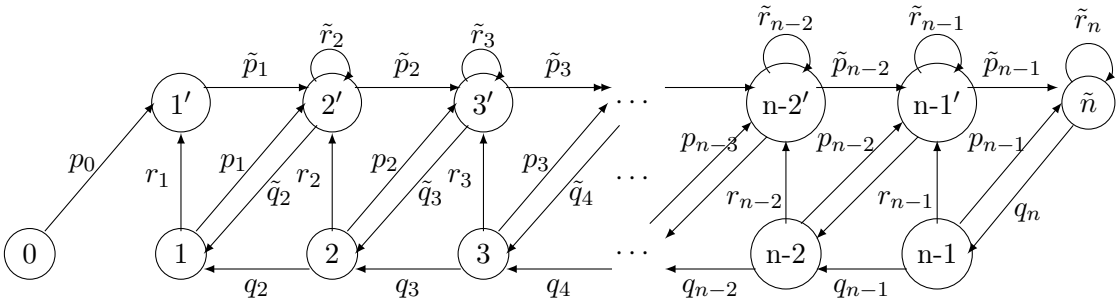
If we fix the potential in the first as an arbitrary parameter and choose the delays in the second arbitrarily, we can see that the two processes are independents and can be studied separately.

In order to compute the expected number of intersections, we follow a trajectory in the chain of moves and for each state, we count the expectation of intersections between this state and the rest of the trajectory.

There is an intersection between state x when arriving in state y when and only when :

- The number of coordinates with different values between x and y is exactly 2
- One of these two coordinates is the direction of the line
- During the stay in x, the other coordinate was accessed.

We can then build a first Markov chain M_1 where the state is defined by the Hamming distance between the current state and x, and whether the last move decreased said distance or not.



with

- The direction of the next efficient move is uniform among the directions other than the last played. For the states with $'$, this player has a different coordinate than the equivalent from x. For the ones without this player has the same coordinate.
- p_i is the probability of increasing distance from a distance of i knowing that the coordinate of the last player played is the same to the corresponding coordinate of x. It is exactly the probability that the next player correspond to one of those equal coordinates, hence $p_i = \frac{n-i-1}{n-1}$
- similarly, $\tilde{p}_i = \frac{n-i}{n-1}$
- q_i is the probability of decreasing distance from a distance of i knowing that the coordinate of the last player played is equal to the same coordinate of x. It is the probability of choosing a player with a different coordinate and then that the new value is the same as x's, hence $q_i = \frac{i}{n-1}\frac{1}{a}$, except in the case of q_1 where the only candidate is an already visited state, so $q_1 = 0$
- similarly, $\tilde{q}_i = \frac{i-1}{n-1}\frac{1}{a}$ and $\tilde{q}_1 = 0$
- r_i is the probability of maintaining distance from a distance of i knowing that the coordinate of the last player played is different from that of x. It is the probability of choosing a player with a different coordinate, and the probability that the new value is still different to that of x, hence $r_i = \frac{i}{n-1}\frac{a-1}{a}$. (except for r_1, in which case

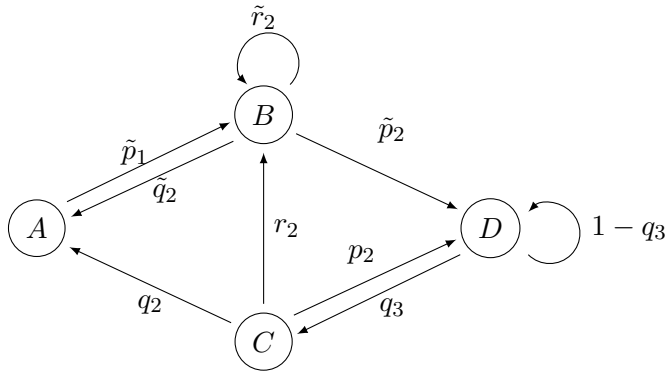


Figure 4.5: Simplified Markov chain M_2

we also have the information that the state which would decrease the distance is already visited, then $r_1 = \frac{i-1}{n-1}$)

- similarly, $\tilde{r}_i = \frac{i-1}{n-1}\frac{a-1}{a}$

In this Markov chain, we want to obtain an upper bound on the number of time the states 2 and $2'$ are accessed in a given number of remaining moves τ, starting from 1. The first remark we can make is that it is decreasing in τ. We can obtain an upper bound of this value if we build a coupling that coincide in both chain when in 2 or $2'$, and only skip states in M_1.

Let us now considers the Markov Chain M_2.

The transitions and probabilities from states 2 and $2'$ are the same as the one from C and B, and the probabilities to go in the direction of these states from any other are always greater in the second chain than in the first.

Thus we can build the following coupling of the two chains :
We associate the states A with $\{1, 1'\}$, B with $\{2'\}$, C with $\{2\}$ and D with $\{3, 3'\}$. We consider want the coupling to verify the following properties, for each pair of associated state, :

- The state in M_2 correspond to a set containing the state in M_1
- At least as many step happened in M_1 as in M_2
- Any passage in 2 or $2'$ in M_1 correspond to a passage in C or B, they cannot be skipped.

We associate the states that each chain start in, which are state 1 and state A. They verify the two properties.
If the current state is 2 or $2'$, associated with C or B, the possible transitions and their probabilities are the same in both Markov chain , so we choose the same transition for

the two trajectories and associate the next state in both chain . This preserve the two properties.
If the current associated states are A with 1 or $1'$, The chain M_2 can only go to $2'$ in one step. If the state in M_1 is $1'$, the only transition bring to B. Else only two trajectories exists from 1, both ending in $2'$ after one and two steps respectively. We then associate in both chain the next passage in $2'$.
If the state associated are D and 3, we can then build the coupling :

- If the next transition in M_1 is p_3 or r_3 (probability $\frac{2}{a(n-1)}$), then we chose in M_2 the self loop (same probability). In both case we associate the following state D in M_2 to the first return to 3 or $3'$ in M_1.

- If the next transition in M_1 is q_3 (probability $\frac{2}{a(n-1)}$), then we chose in M_2 the transition that return to 2 (of same probability) and associate in both chain the next state.

If the state associated are D and $3'$, we can then build the coupling :

- If the next transition in M_1 is $\tilde{p}_3$ or $\tilde{r}_3$ (probabilities $\frac{n-3}{n-1} = 1 - \frac{2}{n-1}$ and $\frac{2}{a(n-1)}$ for a total of $1 - \frac{2}{a(n-1)}$), we choose the transition to C with a global probability of $\frac{1}{a(n-1)}$, we then associate the next state in M_2 with the first return of $M-1$ in 2. Due to the structure of the chain, we know that there wont be any passage in $2'$ before this return. In the remaining $1 - \frac{3}{a(n-1)}$ we choose the self loop and associate the next state in both chain.

- If the next transition in M_1 is $\tilde{q}_3$ (of probability $\frac{2}{a(n-1)}$), then we chose in M_2 the transition that return to C and associate the next state in both chains.

We have now a new Markov chain which contains at least as many visits in B and C as the previous one had in 2 and $2'$, and in less steps for coupled trajectories. This will give a upper bound of the expectations of these visits for fixed length trajectories. We will write the transition matrix of this new chain R

This discrete time chain denoted X_t has the following transition matrix:

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{(n-1)a} & \frac{1-\frac{1}{a}}{n-1} & 0 & 1 - \frac{1}{n-1} \\ \frac{2}{a(n-1)} & 2\frac{a-1}{a(n-1)} & 0 & 1 - \frac{2}{n-1} \\ 0 & 0 & \frac{3}{(n-1)a} & 1 - \frac{3}{(n-1)a} \end{pmatrix}.$$

An upper bound for $H(2,i)$ is obtained by counting the number of visits in states B and C in $m-i$ time steps. As a further upper bound, we have

$$\mathbb{E}^{(m)}H(2,i) \leq \mathbb{E}_{\mathrm{B}}(\#\text{ visits to B before } S_{\mathrm{B}}) + \mathbb{E}_{\mathrm{C}}(\#\text{ visits to C before } S_{\mathrm{C}}), \qquad (4.4)$$

where $\mathbb{E}_x(\cdot)$ denotes the conditional probability $\mathbb{E}(\cdot\,|X_0 = x)$ and S_x is a stopping time defined by $S_x := \min\{t \geq m - i \text{ s.t. } X_t = x\}$. Notice that X_t is an irreducible Markov chain, and S_x is a stopping time such that $\mathbb{E}_x(S_x) < \infty$ and $X_{S_x} = x$. Hence, by Prop. 3 in [1, Chapter 2],

$$\mathbb{E}_x(\#\text{ visits to } x \text{ before } S_x) \leq \pi_x \mathbb{E}_x(S_x), \tag{4.5}$$

where π is the stationary distribution of the Markov chain. Then notice that

$$\mathbb{E}_x(S_x) = m - i + \sum_y [R^{m-i}]_{xy} \mathbb{E}_y(T_x), \tag{4.6}$$

where T_x is the first hitting time of state x. By Lemma 12 in [1, Chapter 2],

$$\pi_x \mathbb{E}_y(T_x) = Z_{xx} - Z_{yx},$$

where $Z := \sum_{j=0}^{\infty}(R^j - \mathbf{1}\pi^T)$ is the fundamental matrix of the Markov chain (here $\mathbf{1}$ denotes a column vector with four ones). Hence, we obtain

$$\pi_x \sum_y [R^{m-i}]_{xy} \mathbb{E}_y(T_x) = \sum_x [R^{m-i}]_{xy}(Z_{xx} - Z_{yx}) = [(I - R^{m-i})Z]_{xx}. \tag{4.7}$$

Using the definition of Z and the fact that $\lim_{j\to\infty} R^j = \mathbf{1}\pi^T$ (since the Markov chain is irreducible and aperiodic), we obtain

$$(I - R^{m-i})Z = \sum_{j=0}^{\infty}(R^j - R^{j+m-i}) = \sum_{j=0}^{m-i-1}(R^j - \mathbf{1}\pi^T) = I - \mathbf{1}\pi^T + \sum_{j=1}^{m-i-1}(R - \mathbf{1}\pi^T)^j$$

and hence

$$[(I - R^{m-i})Z]_{xx} = 1 - \pi_x + \sum_{j=1}^{m-i-1}[R - \mathbf{1}\pi^T)^j]_{xx} \leq -\pi_x + \sum_{j=0}^{\infty}\|(R - \mathbf{1}\pi^T)^j\|_{\max}, \tag{4.8}$$

where $\|\cdot\|_{\max}$ denotes the maximum modulus of all entries in a matrix.

Using equations (4.5)-(4.8) with $x = \mathrm{B}$ and $x = \mathrm{C}$, from the bound (4.4) we obtain

$$\mathbb{E}^{(m)}H(2,i) \leq (\pi_{\mathrm{B}} + \pi_{\mathrm{C}})(m - i - 1) + 2\sum_{j=0}^{\infty}\|(R - \mathbf{1}\pi^T)^j\|_{\max}. \tag{4.9}$$

The matrix of the transitions of this chain is

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{(n-1)a} & \frac{1-\frac{1}{a}}{n-1} & 0 & 1 - \frac{1}{n-1} \\ \frac{2}{a(n-1)} & 2\frac{a-1}{a(n-1)} & 0 & 1 - \frac{2}{n-1} \\ 0 & 0 & \frac{3}{(n-1)a} & 1 - \frac{3}{(n-1)a} \end{pmatrix}.$$

We can compute (the $(n-1)$ factor is to improve readability):

$$(n-1)R^2 = \begin{pmatrix} \frac{1}{a} & \frac{a-1}{a} & 0 & n-2 \\ \frac{a-1}{(n-1)a^2} & \frac{a-1}{a(n-1)} + \frac{1}{(n-1)a^2} & \frac{3}{a}\left(1-\frac{1}{n-1}\right) & (n-1)\left(1-\frac{1}{n-1}-\frac{4}{(n-1)a}\right) \\ 2\frac{a-1}{a^2(n-1)} & \frac{2(a-1)}{a(n-1)} + \frac{2}{(n-1)a^2} & \frac{3}{a}\left(1-\frac{2}{n-1}\right) & \left(n+1-\frac{5}{a}\right) - 2\frac{a-2}{an-1} \\ \frac{6}{(n-1)a^2} & \frac{6(a-1)}{(n-1)a^2} & \frac{3}{a}\left(1-\frac{3}{a(n-1)}\right) & \frac{3}{a}\left(1-\frac{2}{n-1}\right) + (n-1)\left(1-\frac{3}{a(n-1)}\right)^2 \end{pmatrix}.$$

The distribution at equilibrium is

$$\begin{cases} \pi_A &= \frac{6}{a^2n^2-3a(a-1)n+2(a^2+3)} \\ \pi_B &= \frac{n-2}{a}\frac{6}{a^2n^2-3a(a-1)n+2(a^2+3)} \\ \pi_C &= a\frac{6}{a^2n^2-3a(a-1)n+2(a^2+3)} \\ \pi_D &= \frac{(n^2-3n+2)a^2}{6}\frac{6}{a^2n^2-3a(a-1)n+2(a^2+3)} \end{cases}$$

We also know that

$$(R - 1\pi^T)^2 = R^2 - R1\pi - 1\pi R + (1\pi)^2 = R^2 - 1\pi.$$

Hence
$(n-1)(R - \mathbb{1}\pi^T)^2 =$

$$\begin{pmatrix} \frac{1}{a} + O\left(\frac{1}{na^2}\right) & \frac{a-1}{a} + O\left(\frac{1}{na}\right) & -\frac{3}{a} + O(\frac{1}{n}) & -1 + 18a + O(\frac{a}{n}) \\ \frac{a-7}{(n-1)a^2} + O\left(\frac{1}{n^2a^2}\right) & \frac{a-7}{a(n-1)} + \frac{1}{(n-1)a^2} + O\left(\frac{1}{n^2}\right) & -\frac{3}{a} + O(\frac{1}{n}) & -1 + 14a \\ 2\frac{a-4}{a^2(n-1)} + O\left(\frac{1}{n^2a^2}\right) & \frac{2(a-4)}{a(n-1)} + \frac{2}{(n-1)a^2} + O(\frac{1}{n^2}) & -\frac{3}{a} + O(\frac{1}{n}) & \left(18a + 2 - \frac{5}{a}\right) + O(\frac{a}{n}) \\ \frac{18a-12}{(n-1)(n-2)a^2} + O\left(\frac{1}{n^3a^2}\right) & \frac{-6}{(n-1)a^2} + O(\frac{1}{n^2}) & -\frac{3}{a} + O\left(\frac{1}{n}\right) & 18a - \frac{3}{a} + O(\frac{a}{n}) \end{pmatrix}$$

When n tend to infinity and a is a bounded integer superior to 1, the biggest coefficient of $(R - \mathbb{1}\pi^T)$ are constants times $\frac{a}{n}$. This means that the infinite sum converge and is bounded by $O(\frac{a}{n})$.

$\pi_{\mathrm{B}} + \pi_{\mathrm{C}} = \frac{3}{a(n-3)}$, and that, for some positive constants $c_1 \leq 2$ and c_2, $\|(R - \mathbf{1}\pi^T)\|_{\max} \leq c_1$ and $\|(R - \mathbf{1}\pi^T)^2\|_{\max} \leq c_2/n$. This implies that, for all $j \geq 1$, $\|(R - \mathbf{1}\pi^T)^j\|_{\max} \leq c_1(4c_2/n)^{(j-1)/2}$ if j is odd, and $\|(R - \mathbf{1}\pi^T)^j\|_{\max} \leq 4^{\frac{j}{2}-1}(c_2/n)^{j/2}$ if j is even. Hence,

$$\sum_{j=0}^{\infty} \|(R - \mathbf{1}\pi^T)^j\|_{\max} \leq 1 + \sum_{h=0}^{\infty} \left(c_1 + \frac{1}{4}\right)\left(\frac{4c_2}{n}\right)^h = 1 + \left(c_1 + \frac{1}{4}\right)\frac{1}{1 - 4c_2/n},$$

where the last equality is true for $n > 4c_2$. Plugging these bounds into (4.9), we obtain

$$\mathbb{E}^{(m)}H(2,i) \leq \frac{3}{a(n-3)}(m-i-1) + 2 + \left(2c_1 + \frac{1}{2}\right)\frac{n}{n-4c_2}.$$

Recalling that (4.3) gives $\mathbb{E}^{(m)}I \leq 2\sum_{i=0}^{m-1}\mathbb{E}^{(m)}H(2,i)$, we have

$$\mathbb{E}^{(m)}I \leq \frac{6}{a(n-3)}\sum_{i=0}^{m-1}(m-i-1) + \left(4 + (4c_1+1)\frac{n}{n-4c_2}\right)m.$$

Using $\sum_{i=0}^{m-1}(m-i-1) = (m-1)m/2 \leq m^2/2$ and deconditioning, we finally obtain

$$\mathbb{E}I \leq \frac{3}{a(n-3)}\mathbb{E}(M^2) + \left(4 + (4c_1+1)\frac{n}{n-4c_2}\right)\mathbb{E}(M). \tag{4.10}$$

4.5 Round robin

Theorem 13 (**Average complexity of BRA).** *Under the round-robin revision sequence, the average complexity of BRA over a potential game satisfies:*
(i) Average number of moves: $\mathbb{E}[M_{BRA}] = \log(n) + c + O(1/n)$*., where* $c \leqslant e^{\gamma}$
(ii) Average number of comparisons : $\mathbb{E}[C_{BRA}] = e^{\gamma}An + o(An)$.
(iii) Average number of steps: $\mathbb{E}[T_{BRA}] = e^{\gamma}n + o(n)$.

In order to prove this result, we will introduce a model that will serve again in the rest of the thesis.

4.5.1 Proof of Theorem 13

Let y be the potential of the current state x: ($y \stackrel{\text{def}}{=} \Phi(x)$). Let k be the number of players that have already played best response without changing the profile. This is also by definition the number of satisfied players and can replace the explicit set L used in Algorithm **??** when the revision sequence is round-robin.

We use the Intersection Free Approximation.

With the approximation, the evolution at the next step of Best Response Algorithm (BRA) under IFA is as follows:

The kth player computes its best response.
The player has a new strategies whose potential must be compared with the current potential y.
Under the approximation, we assume that the potentials of those strategies are independent and uniformly distributed in $[0,1]$.

With probability y^a none of the new strategies beat the current choice. The state remains at y and one more player is satisfied.

With probability $1-y^a$, one of the new strategies replace the current one as the best response. The state moves to a new state with a larger potential and the number of satisfied players is set back to 1 . The potential increases to a value larger than u with probability $1-u^a$.

Let Y_t be the potential at step t ($Y_t \in [0,1]$) and K_t be the current number of consecutive players whose best response did not change the current potential ($K_t \in \{1,2,\ldots,n\}$) (number of satisfied players). The previous discussion says that the couple (Y_t, K_t) is a discrete-time, continuous-space Markov chain whose kernel is:

$$\mathbb{P}\Big((Y_{t+1},K_{t+1})=(y,k+1)\Big|(Y_t,K_t)=(y,k)\Big)=y^a,$$

and, if $u>y$,

$$\mathbb{P}\Big((Y_{t+1},K_{t+1})\in([u,1],1)\Big|(Y_t,K_t)=(y,k)\Big)=1-u^a.$$

All the other transitions have a null probability.

Let $m(y,k)$ be the number of moves of IFA before convergence when the current state of the Markov chain is equal to (y,k).

With probability y^a, the next player does not change its choice so that $m(y,k) = m(y,k+1)$.
With probability density au^{a-1} the next player finds a new best response with potential u so that one move is taken and $m(y,k) = 1 + m(u,1)$.
Let $M(y,k) = \mathbb{E}[m(y,k)]$. The previous one step analysis of $m(y,k)$ makes $M(y,k)$ satisfy a forward Poisson equation:

$$M(y,k)=y^aM(y,k+1)+\int_y^1 au^{a-1}(M(u,1)+1)du.$$

By definition, the boundary conditions are: $\forall y, M(y,n)=0$ (the current state is NE when all players agree on this) and $\forall k, M(1,k)=0$ (the potentials are all bounded by 1, so a state with potential 1 is guaranteed to be a NE).

By setting $B(y) \stackrel{\text{def}}{=} \int_y^1 au^{a-1}(M(u,1)+1)du$, we get the following system of integral equations

$$\begin{cases} M(y,1) & = y^aM(y,2)+B(y), \\ M(y,2) & = y^aM(y,3)+B(y), \\ \vdots & = \vdots \\ M(y,n-2) & = y^aM(y,n-1)+B(y), \\ M(y,n-1) & = B(y). \end{cases}$$

Successive substitution of $M(y,2),\ldots,M(y,n-1)$ in the first equality yields $M(y,1) = B(y)H(y)$ where $H(y) \stackrel{\text{def}}{=} 1+y^a+\cdots+y^{a(n-2)}$. Differentiating w.r.t. y, one gets an ordinary differential equation in $M(y,1)$:

$$\frac{dM(y,1)}{dy}+\left(ay^{a-1}H-\frac{1}{H}\frac{dH}{dy}\right)M(y,1)=-ay^{a-1}H.$$

The equation is of the form $\dot{f} + gf = h$. Using the boundary condition $M(1,1) = 0$, its generic solution is

$$M(y,1) = e^{-Q(y)} \int_y^1 a u^{a-1} H(u) e^{Q(u)} du. \tag{4.11}$$

where

$$Q(y) \stackrel{\text{def}}{=} \int_0^y \left(a u^{a-1} H(u) - \frac{1}{H(u)} \frac{dH(u)}{du} \right) du = -\log(H(y)) + \int_0^y a u^{a-1} H(u) du. \tag{4.12}$$

The average number of profile changes in the execution of the algorithm starting from an arbitrary profile is $\mathbb{E}[M_{IFA}] = \int_0^1 M(y,1)dy$. Since $M(y,1)$ is decreasing in y, $\mathbb{E}[M_{IFA}]$ is upper-bounded by $M(0,1)$. Using $Q(0) = 0$, $H(0) = 1$ and replacing Q and H by their values,

$$\begin{aligned}
M(0,1) &= \int_0^1 a u^{a-1} \exp(\sum_{i=0}^{n-2} \frac{u^{a(i+1)}}{i+1}) du \quad = \int_0^1 \exp(\sum_{i=0}^{n-2} \frac{v^{i+1}}{i+1}) dv \quad \text{(with } v = u^a\text{)} \\
&= \int_0^{1-\frac{1}{n}} \exp(\sum_{i=1}^{n-1} \frac{v^i}{i}) dv + \int_{1-\frac{1}{n}}^1 \exp(\sum_{i=1}^{n-1} \frac{v^i}{i}) dv \\
&\leqslant \int_0^{1-\frac{1}{n}} \exp(\sum_{i=1}^{\infty} \frac{v^i}{i}) dv + \frac{1}{n} \exp(\sum_{i=1}^{n-1} \frac{1}{i}) \qquad (4.13) \\
&= \int_0^{1-\frac{1}{n}} \frac{dv}{1-v} + e^\gamma + O(1/n) \quad = \log(n) + e^\gamma + O(1/n). \qquad (4.14)
\end{aligned}$$

Furthermore, this bound is tight, up to an additive constant In the derivation of (4.14), the only inequality is (4.13). Let us show this inequality can be bounded by a constant.

$$\left[\int_0^{1-\frac{1}{n}} \exp(\sum_{i=1}^{\infty} \frac{u^i}{i})du + \int_{1-\frac{1}{n}}^{1} \exp(\sum_{i=1}^{n-1} \frac{u^i}{i})du\right] - M(0,1), \tag{4.15}$$

$$= \left[\int_0^{1-\frac{1}{n}} \exp(\sum_{i=1}^{\infty} \frac{u^i}{i})du + \int_{1-\frac{1}{n}}^{1} \exp(\sum_{i=1}^{n-1} \frac{u^i}{i})du\right] - \int_0^{1} \exp(\sum_{i=0}^{n-1} \frac{u^i}{i})du, \tag{4.16}$$

$$= \int_0^{1-\frac{1}{n}} \left(\exp(\sum_{i=1}^{\infty} \frac{u^i}{i}) - \exp(\sum_{i=1}^{n-1} \frac{u^i}{i})\right) du, \tag{4.17}$$

$$\leqslant \int_0^{1-\frac{1}{n}} \exp\left(\sum_{i=1}^{\infty} \frac{u^i}{i}\right)\left(\sum_{i=n}^{\infty} \frac{u^i}{i}\right) du, \quad \text{[by convexity of exp]} \tag{4.18}$$

$$= \int_0^{1-\frac{1}{n}} \frac{1}{1-u} \int_0^{u} \frac{v^{n-1}}{1-v} dvdu, \tag{4.19}$$

$$= \int_0^{1-\frac{1}{n}} \frac{v^{n-1}}{1-v} \int_v^{1-\frac{1}{n}} \frac{1}{1-u} dudv, \tag{4.20}$$

$$= \int_0^{1-\frac{1}{n}} \frac{v^{n-1}}{1-v} (\log(n) + \log(1-v))dv, \tag{4.21}$$

$$= \sum_{k=0}^{n-2} \int_{k/n}^{(k+1)/n} \frac{v^{n-1}}{1-v} (\log(n) + \log(1-v))dv, \tag{4.22}$$

$$\leqslant \sum_{\ell=2}^{n} \frac{\log(\ell)}{\ell+1} \left(1 - \frac{\ell-1}{n}\right)^{n-1}, \quad [\ell = n-k] \tag{4.23}$$

$$\leqslant \sum_{\ell=2}^{n} \frac{\log(\ell)}{\ell+1} \exp\left(-\ell + 1 + \frac{\ell-1}{n}\right) \tag{4.24}$$

$$\leqslant \sum_{\ell=2}^{\infty} \frac{\log(\ell)}{\ell+1} \exp\left(\frac{-\ell+1}{2}\right). \tag{4.25}$$

This series converges to a finite constant smaller than 0.39.

The inequality (4.23) comes from the fact that the positive function $f_1(v) := \frac{v^{n-1}}{1-v}$ is increasing in v while $f_2(v) := \log(n) + \log(1-v)$ is decreasing in v, and positive as long as $v < 1 - 1/n$. Therefore, an upper bound of the product $f_1(v)f_2(v)$, is obtained under the form of a staircase function with stairs of size $1/n$ by replacing the value of v by $\frac{k+1}{n}$ in f_1 and by $\frac{k}{n}$ in f_2. Namely, for all $\frac{k}{n} \leqslant v \leqslant \frac{k+1}{n}$, $f_1(v)f_2(v) \leqslant f_1(\frac{k+1}{n})f_2(\frac{k}{n})$.

Finally, the definition of M_{IFA} implies that $M(0,1) - 1 \leqslant \mathbb{E}M_{IFA} \leqslant M(0,1)$.

Let us now consider the average number of comparisons made by BRA under the intersection-free assumption. Let $C(y,k)$ be the average number of comparisons starting in a state with potential y and k players have played without changing their strategy.

The Poisson equation for $C(y,k)$ is :

$$C(y,k) = y^a(C(y,k+1)+a) + \int_y^1 au^{a-1}(C(u,1)+a)du,$$

with the boundary conditions $C(1,1) = a(n-1)$ and $C(y,n) = 0$.

The solution of this differential system can be obtained in closed form, using a similar approach as for $M(y,1)$.

$$C(y,1) = a\left(\sum_{i=0}^{n-2} y^{ai}\right)\exp\left(-\sum_{i=1}^{n-1}\frac{y^{ai}-1}{i}\right).$$

The average number of comparisons is $\mathbb{E}[C_{IFA}] = \int_0^1 C(y,1)dy$.

For all $y < 1$,

$$\begin{aligned} C(y,1) &= a\left(\sum_{i=0}^{\infty} y^{ai}\right)\exp\left(\sum_{i=1}^{n-1} 1/i\right)\exp\left(-\sum_{i=1}^{\infty} y^{ia}/i\right) + o(an) && (4.26)\\ &= a\frac{1}{1-y^a}(n-1)e^{\gamma}(1-y^a) + o(an) + O(1) && (4.27)\\ &= a(n-1)e^{\gamma} + o(an), && (4.28)\end{aligned}$$

where γ is the Euler constant ($\gamma \approx 0.5772...$). Therefore, the same equality holds for the integral, equal to $\mathbb{E}[C_{IFA}]$.

A natural question is whether this remarkable performance is simply because there are so many NE on average that any algorithm will be efficient. This question is addressed by comparing the performance of the dynamics with a fully random algorithm in Section 4.8, along with analyzes of other variants. A numerical study of the performance of BRA, especially under Bernoulli revisions is given in Section 4.9 and shows that it performs almost as well as round-robin.

4.6 Optimality of BRD

In this section, we prove that BRA finds a Nash equilibrium faster than any local search algorithm (defined in Definition 15), in the strong stochastic order sense.

Theorem 14 (Optimality of BRA). *Let A be any local search algorithm that computes a Nash Equilibria in potential games. Under the foregoing randomization, and choosing the starting point x^0 uniformly among all states, $\forall t \geqslant 0$, $\mathbb{P}(T_{BRA} \geqslant t | R = R_A) \leqslant \mathbb{P}(T_A \geqslant t)$, where R_A is the revision sequence constructed in A.*

Before starting on the proof, I need to precise and define a few things.

4.6.1 Local Search Algorithm

Let us consider the class of *Local Search Algorithms* defined by the following condition.

Definition 15 (Local Search Algorithm). An algorithm is a Local Search Algorithm if the only operations that it can perform are among the following list:

- Choose any strategy profile (*i.e.* an strategy for each player) $(x_1, \ldots, x_n)$.
- Read (and store in memory) the payoff vector of one player (k), under a given profile (*i.e.* all the values of $u_k(\cdot, x_{-k})$ for a fixed x_{-k}).
- Any operations over stored data (comparisons, arithmetics, ...).

Basically, the only restriction on local search algorithms is the fact that they can only access the game information, one player at a time. Any local search algorithm can be written in the following form, based on the history of the execution, $\mathcal{H}_t$, that corresponds to the amount of information gathered by the algorithm up to step t.

Algorithm 7: A general local search algorithm

Initial storage reduced to the initial profile: $\mathcal{H}_0 := \{(x(0)\}$.
repeat
  Select next player: $k := R(\mathcal{H}_t)$;
  Read payoff vector of k under current state: $u_k(\cdot, x_{-k}(t))$;
  Store the new visited states and their payoffs in memory:
  $\mathcal{H}_{t+1} := \mathcal{H}_t \cup \{((\alpha, x_{-k}(t)), u_k(\alpha, x_{-k}(t)))_{\alpha \in \mathcal{A}_k}\}$;
  jump to next state $x(t+1) := J(\mathcal{H}_{t+1})$;
  Set $stop := 1$ if the current state is a NE;
  $t := t + 1$;
until *stop*;

The functions J and R used in the inner loop are arbitrary functions that choose the next state as well as the next player to play, according to the whole history of the process. These functions can be deterministic or random. Testing if $x(t+1)$ is NE is not detailed. It can only be done when all the payoff vectors for all the players in state $x(t+1)$ have been stored in memory.

As examples, the Best Response Dynamics are local search algorithm, similarly, an algorithm that we could call full-BRA, which would search all $(na-1)$ neighbors of the current state then jump to the one with maximal potential or the algorithm choosing a state randomly then testing if it is a Nash equilibrium are local search algorithms.

In fact, one could say that all algorithms that would be implementable as distributed programs, executed independently by the players, will fall in this class.

4.6.2 Measure of Complexity

I choose to define the complexity of a local search algorithm A is defined as the total number of payoff vector reads (denoted T_A).

On a real instance, the effective complexity will sometime be the number of comparisons, when the limiting factor is the access to the utility values, other times it will be the number of line read, especially in distributed game where the limiting factor is to prevent player from acting too near each other, or when the reading of the different possibles payoff can be done concurrently. We can also imagine case where the pertinent data is the number of moves instead, if the algorithm can access and keep in memory big bloc of informations or when the problem have a measure of smoothness.

When the measure of complexity is not the number of lines accessed, we can use a reasoning similar showing that the variant of best response dynamics that obtain exactly one unit of the chosen complexity at each step become the optimum.

4.6.3 Proof of the Statement

The proof holds in several steps.

We first introduce a subclass of local search algorithms called *no-jump algorithms*. We will first prove that BRA is optimal within the class of no-jump algorithms before proving it for the entire class of local search algorithms.

Definition 16 (**No-Jump algorithm).** A no-jump algorithm is a local search algorithm that only reads a potential vector starting from a state visited during the previous read.

The profiles visited by a no-jump algorithm form a walk over the states, with only transition with neighbor states whose potential can be read and compared to the previous one.

BRA is a no-jump algorithm. (as other examples, full BRA is a no jump, while randomly testing for Nash equilibria is not).

Any no-jump algorithm can be written in the following form, similar to any local search up to the function J, replaced by a more constraint function W, that imposes the next state to belong to the newly explored states.

Algorithm 8: A general no-jump algorithm

Initial storage reduced to the initial profile: $\mathcal{H}_0 := \{(x(0), \Phi(x(0))\}$
repeat
  Select next player: $k := R(\mathcal{H}_t)$;
  Read payoff vector of k under current state $u_k(\cdot, x_{-k}(t))$;
  Store the new payoffs in memory:
  $\mathcal{H}_{t+1} := \mathcal{H}_t \cup \{((\alpha, x_{-k}(t)), u_k(\alpha, x_{-k}(t)))_{\alpha \in \mathcal{A}_k}\}$;
  walk to next state $x(t+1) := W(\mathcal{H}_{t+1})$; (in $\{(\alpha, x_{-k}(t))_{\alpha \in \mathcal{A}_k}\}$)
  Set *stop* $:= 1$ if the current state is a NE
  $t := t + 1$;
until *stop*;

To compare two algorithms other potential games, we use the measure over games defined by chosing potential identically and independently defined earlier. Any continuous

distribution being equivalent, we use the uniform distribution between 0 and 1.

Let us consider one run of algorithm 8 starting with state $x(0) = (x_1(0), \ldots, x_n(0))$ and using the sequence R of players. After the first step (involving player, say ℓ), the new state becomes $x(1)$ where only the ℓ-th coordinate may have changed. At this point, we add a flag to all the visited states, of the form $(\alpha, x_{-\ell}(1))$, whose potential is smaller (or equal to) the potential of $x(1)$.

The flagging goes on in the same manner with BRA. This brings the definition of *traces* that capture the whole history of the algorithm in a compact form.

Definition 17 (**Trace).**

- A *trace* is the sequence S_i of the flagged states at each step of the algorithm, together with a distribution F of the potential of the current state x. Let $T(R, i, \mathfrak{G}, x^0) = (S_i, x, F)$ denote the trace on game $\mathfrak{G}$, starting in x^0, after the strategies of the first i plays in R. It should be clear that the past behavior of the algorithm up to step i is entirely characterized by its trace up to step i.
- Two traces (S, x, F) and (S', x', F') are *comparable*, denoted $(S, x, F) \prec (S', x', F')$ if they have the same final state, $(x = x')$ and the sequences of flagged states are included in one another ($\forall i, S_i \subset S'_i$) and the distribution of potential of the final states are comparable: $\forall u \in [0, 1], F(u) \leqslant F'(u)$. Two traces are also comparable when the final state of the second trace is a Nash equilibrium.

The goal of this comparison is to determine which trace is nearer to reaching a NE.

Definition 18 (**Transform).** A measure-preserving transform θ is a bijection on games that preserves the probability measure on games: If G is any measurable set of games (or equivalently for us a measurable set of potentials), then $\mathbb{P}(G) = \mathbb{P}(\theta(G))$.

Definition 19 (m**-class).** A set of games forms an m-class if they are equivalent up to step m: $\mathfrak{G}_1 \equiv_m \mathfrak{G}_2$ if there exists $x^0, x^{0'}$, and a transform θ such that $\forall i \leqslant m, T(R, i, \mathfrak{G}_1, x^0) = \theta T(R, i, \mathfrak{G}_2, x^{0'})$.

Lemma 20. *Let us consider a revision sequence that used player ℓ at step m while the current step is $m' > m$, and two arbitrary distinct strategies of ℓ, namely α_1 and α_2.*

There exists a measure-preserving transform θ such that for any game $\mathfrak{G}$ and for any state z,

- *If $z_\ell \neq \alpha_1$ and $z_\ell \neq \alpha_2$, z is flagged in $\mathfrak{G}$ iff z is flagged in $\theta(\mathfrak{G})$.*
- *If $z_\ell = \alpha_1$, z is flagged before m in $\mathfrak{G}$ iff $(z_{-\ell}, \alpha_2)$ is flagged before m in $\theta(\mathfrak{G})$*
- *If $z_\ell = \alpha_2$, z is flagged before m in $\mathfrak{G}$ iff $(z_{-\ell}, \alpha_1)$ is flagged before m in $\theta(\mathfrak{G})$*
- *After m' steps, the current state in $\mathfrak{G}$ and $\theta(\mathfrak{G})$ is the same.*

In other words, the set of all the flagged states of the hyperplane of coordinates α_1 and α_2 at time m can be switched for the purpose of trace comparisons, after the next jump along ℓ.

Proof. Let us consider a game $\mathfrak{G}$ and let x^m be the current state at step m.
If $x_\ell^{m'} \neq \alpha_1$ and $x_\ell^{m'} \neq \alpha_2$, then we can construct a new game by permuting the coordinates α_1 and α_2 in ℓ. This transform is a valid candidate for θ. Since every change is a permutation of independent random variable, the measure is preserved.

If $x_\ell^{m'} = \alpha_1$. We build a new game by temporary adding a new coordinate along ℓ, assume it has never been visited before m and carry x^m. Every flagged state in this plane has been flagged by a line of the form $(z_{-\ell}, \cdot)$ and the corresponding point of any other plane, including $(\cdot, \alpha_1)$ is flagged as well, so the part of the trace between the jumps is valid. We exchange the plane $(\cdot, \alpha_1)$ and $(\cdot, \alpha_2)$, then bring back the new plane on α_1, flagging any point that was flagged in either $(\cdot, \alpha_1)$ or the new plane. This transformation preserves the potential and their comparisons, so it preserves the measure.

In all cases, we have built a new game with permutations of the potential, preserving the final state and exchanging two hyperplanes. □

We are now ready for the proof of the main lemma

Lemma 21. *Let us consider two traces T and T' that are comparable at step m: $T_m = (S_m, f, y) \prec (S'_m, f', y) = T'_m$. Then there exists a coupling such that they remain comparable steps by steps when both are completed using BRA, until convergence, of BRA on the two games.*

Proof. The proof holds by induction on the number of players.

With a single player a trace for BRA is either empty (already an equilibrium) or one step, with the case equilibrium being shorter than any other. If two trace are comparable, then after a step under BRA in both the final step only have flagged states as neighbor and they are still comparable.

Let us now assume that the property holds for $\ell - 1$ players. In the first m steps of sequence R, player ℓ appears k times. We use now an induction on k. If $k = 0$, we are back to the case with $\ell - 1$ players, where the property holds by assumption. Let us now assume the property holds for all sequence when player ℓ appears $k - 1$ times and consider its k-th appearance. We can use the recurrence hypothesis to build a transform θ up to this step. For any class of θ, we consider the coordinate chosen at this step.
We further constraint the transform θ by associating, whenever possible, the traces where the new coordinate along ℓ is the same. We know that the potential and flagged state are both higher in the second trace, and that any choice not stay immobile nor flagged is equivalent, hence we can focus on the cases where the trace do not coincide, are of the form: new coordinate for the first trace, immobile for the second.

If both traces keep still, we can ignore this step, and use the second induction hypothesis (on k) on the sequence without this instance of ℓ. The result will be preserved when reinserting these two empty steps.

If both trace lead to new states, we can use the lemma 10 and a permutation to ensure that it is the same new states, regarding flagged state, hence we still have comparability. We can then use the first induction hypothesis on the sequence from this step to the step m to obtain the result.

If only the first trace stays still, we can use Lemma 20 between the planes of the two current states of the second trace in order to have comparable flagged states : We obtain a new pair of games in dimension $\ell - 1$, such that the set of flagged state of the first is included in the second, and the law of the potential of the first is below that of the second. We can apply the first induction hypothesis on the reduced game.

We can remark that at any point, the inclusion of the sets of flagged states impose that if the second trace starting point is a detected equilibrium, then so is the corresponding point of the first one.

In the second and third cases, we can apply the lemma on the induced game and obtain coupling where either the first trace has converged globally before the second, the first converged and the second continue, or both continued and the induction assumptions are preserved.

□

Proof of Theorem 14 for no-jump algorithms. Let us consider two algorithms. The first one uses the choice function W up to step $m+1$ and the best response function **br** for the remaining steps. The second one uses the choice function W up to step m and the best response function **br** for the remaining steps.

From step $m + 1$ on, the comparison between the algorithms can be seen as a comparison between two ordered traces, under BRA. After step m, the traces will remain comparable by Lemma 20. The first one being smaller than the second one (sequence of flagged states as well as potential of the end point). The previous lemma 21 says both traces will remain comparable, using the appropriate transform, under BRA. This implies that the second trace will reach a local maximum before the first one.

Proof of Theorem 14 for all local search algorithms. Consider any local search algorithm. By modeling the payoff vectors as vertices of a graph with arcs when the vertices directed by the order of visit, we obtain a graph that we can divide and cover by no-jump algorithms. The algorithm can then be seen as a group of no-jump algorithm that may sometimes merge or fork. In case of fork the new trace can always improved (in terms of trace comparison) by starting at the last point of the first one instead of in the middle. This means that all the no-jump algorithms composing the local search

algorithm can be compared trace-wise with a single one.

4.7 Second Moments

In this part, we compute $\mathbb{E}[m^2]$ and $\mathbb{E}[\tau^2]$ where m and τ are the random variables denoting respectively the number of moves and the number of steps of IFA.

Using the same notations as in Section **??**, the number of moves squared satisfies
$m^2(y,k) = m^2(y,k)$ w.p. $P((y,k),(y,k+1))$ and
$m^2(y,k) = (1+m(u,1))^2$ w.p. $P((y,k),(u,1))$.

Let $S(y,k) := \mathbb{E}m^2(y,k)$. The previous equalities imply that, using the first moment M,

$$S(y,k) = y^a S(y,k+1) + \int_y^1 au^{a-1}(S(u,1) + 2M(u,1) + 1)du.$$

Using the same approach as in Section **??**, and the function $H(y) = 1 + \cdots + y^{a(n-2)}$ defined in Section **??**, one gets

$$S(y,1) = G(y)H(y) \tag{4.29}$$

where $G(y) \stackrel{\text{def}}{=} \int_y^1 au^{a-1}(S(u,1) + 2M(u,1) + 1)du$. Its solution is

$$S(y,1) = \exp(-Q(y)) \int_y^1 au^{a-1}(2M(u,1)+1)H(u)\exp(Q(u))du.$$

where Q is defined in (4.12). Since $M(0,1) = \log(n) + c + o(1/n)$, this implies that

$$\begin{aligned} S(0,1) &= (2\log(n) + 2c + o(n^{-1})) \left[\exp(-Q(y)) \int_y^1 au^{a-1}H(u)\exp(Q(u))du\right], \\ &= (2\log(n) + 2c + o(1/n))(\log(n)) + c + o(1/n)) \end{aligned}$$

because the term between brackets is exactly $M(0,1)$ (see Equation (4.11)). Therefore, the standard deviation of the number of moves is $\sqrt{S(0,1) - M(0,1)^2} = \log(n) + O(\sqrt{\log(n)})$.

The computation of the standard deviation of the number of steps follows the same path but is more cumbersome. It is easier to get a bound on the second moment. Let $U(y,k) \stackrel{\text{def}}{=} \mathbb{E}\tau^2(y,k)$. The one step analysis of the behavior of $\tau(y,k)$ yields the following formula for U, using the first moment T:

$$U(y,k) = y^a(U(y,k+1) + 2T(y,k+1) + 1) + \int_y^1 au^{a-1}(U(u,1) + 2T(u,1) + 1)du.$$

Using this recurrence equation and the recurrence equation satisfied by T, namely

$$T(y,k) = y^a(T(y,k+1) + 1) + \int_y^1 au^{a-1}(T(u,1) + 1)du,$$

as well as the inequality $T(y,k) \leqslant T(0,1) = e^{\gamma}na + o(n)$, one gets

$$\begin{aligned} U(y,1) &\leqslant \left(y^a(2e^{\gamma}na+1) + o(n) + \int_y^1 au^{a-1}(U(u,1)+1)du \right) H(y), \\ &\leqslant 2e^{\gamma}n^2a + o(n^2). \end{aligned}$$

This gives a standard deviation $\sqrt{U(0,1) - T^2(0,1)} = O(n)$.

4.8 Variants

This analysis of BRA leads naturally to consider variants of BRA that may improve on the convergence time. Let us consider the following variant of the BRA, called Full-BRA in the following.

- At each step *all* the players are active and they all compute their best response.
- The player whose best response has the largest potential wins and only this player actually plays its best response.

This algorithm should use less moves than BRA before convergence because, at each step, the increase of the potential is larger under Full-BRA.

Its average time complexity can be computed in the same way as before, and the computation is even simpler.

Theorem 22. *The number T_F of steps and the number of moves M_F, needed for the Full-BRA Algorithm, under IFA, to reach a NE are equal and constant on average: $\mathbb{E}[M_F] = \mathbb{E}[T_F] = e - 1 + o(1)$. The number of comparisons C_F before convergence is on average $\mathbb{E}[C_F] = ea(n-1) + o(An)$.*

Before giving the proof of the theorem, let us comment on this result. It should be rather striking that the number of steps T_F does not depend on the number of players nor on the number of strategies per player. However, notice that its number of comparisons is larger than for BRA.

Proof. First, it should be clear by definition of Full-BRA that the the number of moves is equal to the number of steps. Indeed, a step that does not provoke a move implies that a NE has been found.

Let us consider now a new discrete time Markov chains $Y(t)$ whose state at step t is the potential of the profile reached by Full-BRA at step t, under the intersection-free approximation.

The transitions probabilities are the following (using $b \stackrel{\text{def}}{=} a(n-1)$):

$$\mathbb{P}(Y(t+1) \in [z,1] | Y(t) = y) = 1 - z^b, \quad z > y \tag{4.30}$$
$$\mathbb{P}(Y(t+1) = y | Y(t) = y) = y^b. \tag{4.31}$$

The average number of steps before convergence starting with a potential y, $S_F(y)$ satisfies the Poisson equation:

$$S_F(y) = \int_y^1 bu^{b-1} S_F(u) + 1) du,$$

with the boundary condition $S_F(0) = 1$.

By differentiating w.r.t. y, one gets $S'_F(y) = -by^{b-1}(S_F(y) - 1)$, whose solution is

$$S_F(y) = e^{y^b} - 1.$$

The expected number of steps is $\mathbb{E}T_F = \int_0^1 S_F(y)dy$. By replacing y by one in the integral, we get $\mathbb{E}[T_F] \leqslant e - 1$. More over, this bound is tight because for all y, $S_F(y)$ goes to $e - 1$ when n or A go to infinity. Therefore,

$$\mathbb{E}[T_F] = e - 1 + o(1).$$

As for the average number of comparisons performed by the algorithm. It satisfies

$$\mathbb{E}[C_F] = a(n-1)(\mathbb{E}[T_F] + 1) = ea(n-1) + o(An).$$

□

At this point, we have shown that BRA converges after $\log(n)$ moves while Full-BRA converges after a constant number of steps. However, Full-BRA makes more comparisons than BRA on average (eAn against $e^\gamma An$, so approximately An more comparisons for Full-BRA). This gives the impression that both algorithms are very efficient to compute NE in large games. However a natural question is whether this is simply because there are so many NE on average that any algorithm will be efficient. For that let us consider the following naive random algorithm (denoted rand) to find a NE:

- Pick a profile at random and check if this is a NE.
- Else, pick again.

Using the same approach, the average number of steps (or moves) T_{rand}, starting with potential y, of this random algorithm satisfies $T_{\text{rand}}(y) = (1 - y^c) \int_0^1 (M_{\text{rand}}(u) + 1) du$, with $c \stackrel{\text{def}}{=} (A-1)n$. The average number of steps is therefore

$$\mathbb{E}[T_{\text{rand}}] = \int_0^1 T_{\text{rand}}(y)dy = (1 + \mathbb{E}[T_{\text{rand}}]) \int_0^1 (1 - y^c)dy = c = an.$$

This is linear in A and n, while the average number of comparisons can also be computed. Here, as soon as a state is found with a larger potential than the picked profile, this profile is rejected as not being a NE. Therefore, the number of comparisons per step is a random variable, smaller than an. Its average value over all steps is

$$\mathbb{E}[C_{\text{rand}}] = \mathbb{E}[T_{\text{rand}}] \int_{y=0}^{1} \left(\sum_{i=0}^{c} y^i \right) dy = c \sum_{i=1}^{c} \frac{1}{i} = An \log(An) + O(An).$$

So this simple algorithm takes more steps than the BR approaches in terms of number of comparisons. Also the quality of the NE (the potential of the NE) obtained by the BR approaches is better than for random NE. Let us assume that the potential Φ of all states is uniformly distributed in $[0, 1]$ (unlike for the variables M, C and T, the potential of a NE depends on the distribution used to generate Φ).

In that case, the average potential of a random NE_{rand} is

$$\mathbb{E}[\Phi(\text{NE}_{\text{rand}})] = \int_0^1 (1 - t^c) dt = \frac{c}{c+1} \approx 1 - \frac{1}{An}.$$

Meanwhile, the average potential of a NE found by the BRA and Full-BRA is given by the following computation (only done for Full-BRA, but BRA is treated the same way).

Let us denote by $D_F(y, h)$ the probability that the potential of the NE found by Full-BR is larger than h, starting with a profile with potential y. $D_F(y, h)$ satisfies the Poisson equation,

$$D_F(y, h) = (1 - h^b) + \int_y^h (b - 1) u^b D_F(u, h) du.$$

Differentiating w.r.t. y yields

$$D_F(y, h) = (1 - h^b) \exp(h^b - y^b).$$

The average potential of the Nash equilibrium reach by Full-BRA, (NE_F) is therefore

$$\mathbb{E}[\Phi(\text{NE}_F)] = \int_0^1 \left(y + \int_y^1 (1 - h^b) \exp(h^b - y^b) dh \right) dy.$$

This is bounded from below by the average potential of a NE starting with potential 0. $\mathbb{E}[\Phi(\text{NE}_F)] \geqslant \int_0^1 (1 - h^b) \exp(h^b) dh$.

Note that this compares favorably with the average potential of a random NE (the gap to 1 is at least twice as small) since by bounding $\exp(y)$ by $(1 + y)$,

$$\mathbb{E}[\Phi(\text{NE}_F)] \geqslant 1 - \frac{1}{2b+1} \approx 1 - \frac{1}{2AN}.$$

4.9 Numerical Experiments

The expected values computed in the previous section might only be theoretical results with little practical interest if the distribution of the execution time has a large variance.

To test this we have run several simulations of BRA over random potential games and we computed the standard deviations and confidence intervals. To run these experiments with a large number of players, we do not generate the potentials of all states (there are A^N states). Instead, we generate the potentials on the fly, only when they are needed. Hopefully, as suggested by Theorem 13, BRA should converge fast, so without exploring too many states. Of course, the following simulations are exact runs of BRA and do not use the intersection-free approximation: the potentials of all visited states are kept in memory in case some states would be visited several times by the algorithm.

The following numerical experiments are also used to test our claim that while IFA is an upper-bound of BRA, this bound should be very tight. We compare the theoretical results obtained for IFA in Section 2 with the experimental performance of BRA obtained here.

Finally, the numerical experiments are also used to assess the performance of alternative revision sequences, such as Bernoulli sequences, instead of round-robin.

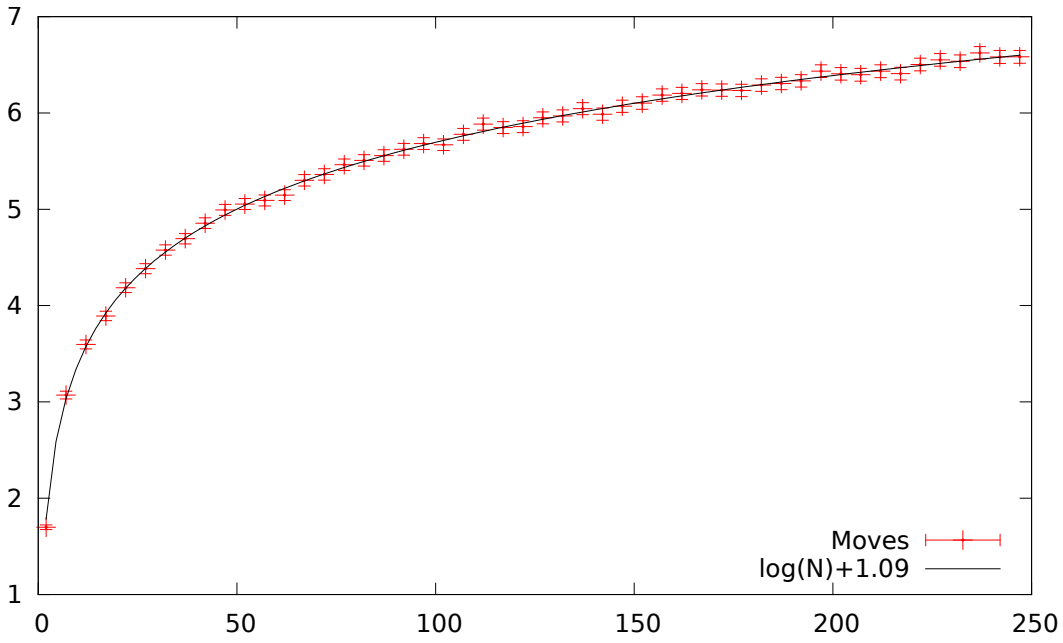
The following figures correspond to simulations on games, all with $A = 30$. The number of players N ranges from 0 to 250. All these games have potentials chosen uniformly and independently in $[0, 1]$. For each value of N, algorithm BRA is run 5000 times. The error-bars correspond to confidence interval at 95%.

4.9.1 Number of Moves, Number of Steps

As one can see in Figure **??** the average number of moves for BRA with round-robin revisions is approximately $\log N + 1.09$, while Theorem 13 predicts that it should less than $\log N + e^\gamma$, with $e^\gamma \approx 1.7$. The confidence intervals are very small. This is a numerical evidence of the fact that the distribution of M_{BRA} is tight around its mean. Also, the confidence intervals do not grow with N, meaning that the variance barely depends on N.

Figure **??** displays the average number of steps for BRA under round-robin to reach an equilibrium. The best fit function is $(e^\gamma - 1)N$ while exactly N additional steps are needed to check that the current state is a NE, so that the total number of steps is $e^\gamma N$. This is exactly what Theorem 13 says. Here also, the very tight confidence intervals do not grow with N and provide numerical evidence that the variance is small (and the distribution is not spread).

Average number of moves.
The best fit function is $\log(N) + 1.09$



Average number of steps before reaching a NE.
The predicted value is $(e^{\gamma} - 1)N$.

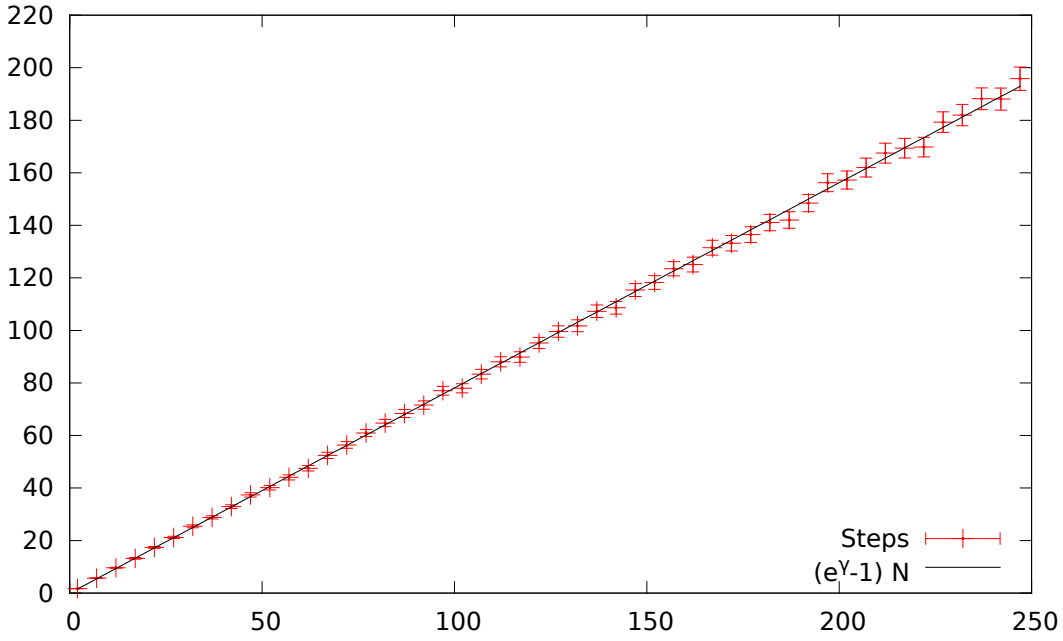
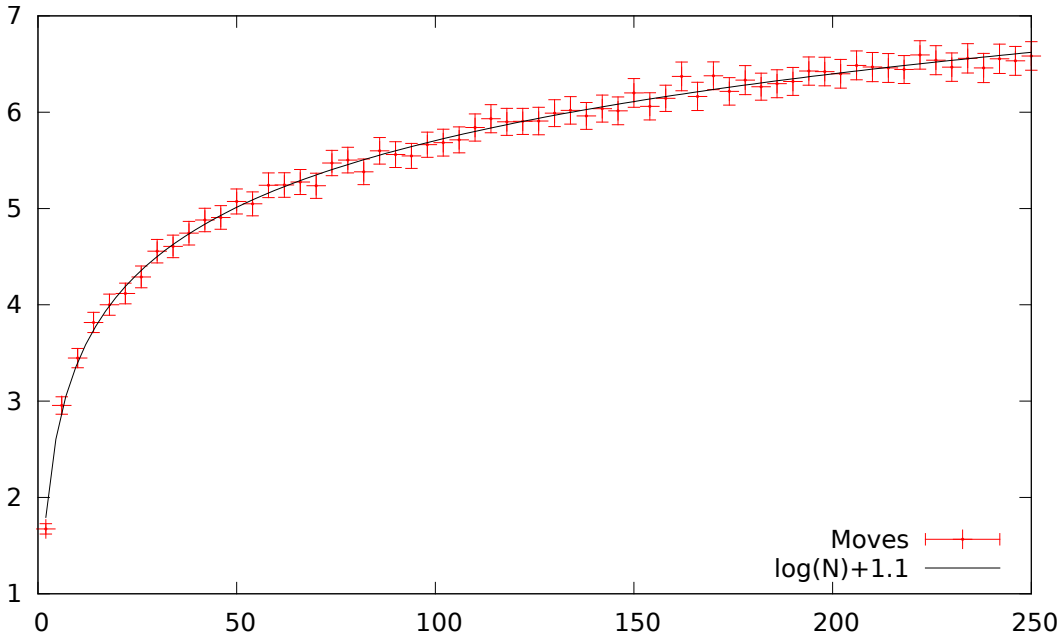


Figure 4.6: Simulation results for Round-Robin BRA

Average number of moves.
The best fit function is $\log(N) + 1.1$.



Average number of steps before reaching a NE.
The best fit function is $1.22N$.

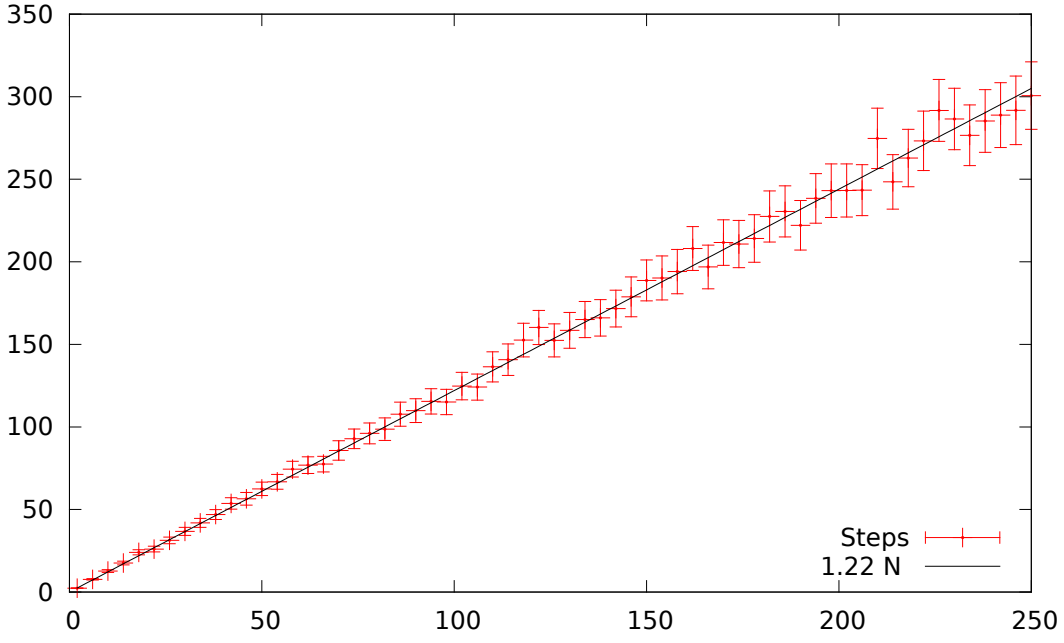


Figure 4.7: Simulation results for BRA, with a Bernoulli random revision sequence.

4.9.2 Bernoulli Revision Sequences

We have run BRA with Bernoulli random revision sequences: at each step, the next player is chosen randomly, each with probability $1/N$.

Its average performance has not been analyzed theoretically. It is rather direct to show that the average number of steps under a Bernoulli revision sequence can be bounded by the average number of steps under round-robin multiplied by $\log N$ by using a coupon collector argument: it takes on average $N \log N$ steps before all N players have played once. However this bound is obviously not tight.

The average number of steps for BRA with a random Bernoulli revision sequence before reaching an equilibrium is displayed in Figure **??**. The empirical best fit function is $1.22N$. The total number of steps is equal to this value plus the verification time for NE, the latter is given by the coupon collector, whose average duration is $N \log N + \gamma N + o(N)$. Therefore, the empirical number of steps is equal to $1.22N + (N \log N + \gamma N)$. This is to be compared with the linear behavior obtained for round-robin where the average number of steps can be decomposed into $(e^{\gamma} - 1)N + N \approx 0.78N + N$. Under round robin, a NE is reached on average when 78% of the players have played once. As for Bernoulli revisions, the simulation means that an equilibrium is reached on average before the size of the list L of satisfied players becomes bigger that $1 - e^{-1.22}N \approx 0.7N$.

In Figure **??**, the standard deviation increases with N. This growth comes from the fact that the variance also depends on the revision sequence. The variance of the last step (verification that the last state is indeed a NE) is not displayed in Figure **??**. This is simply the standard deviation of the coupon collector, known to be equal to $\frac{\pi N}{\sqrt{6}} + o(N)$. This brings an additional inefficiency in this case.

The average number of moves for BRA with a random revision sequence obtained by simulation is displayed in Figure **??**. With IFA, the number of moves is exactly the same under all revision sequences. As expected, without IFA, these simulations show that the expected number of moves is very close to the round-robin case, just slightly larger: best fit is $\log N + 1.1$ here instead of $\log N + 1.09$ for round-robin.

4.9.3 Simulations of Full-BRA

As seen in Figure 4.8, the average number of steps for full-BRA remains constant, equal to e. Again the upper-bound predicted by Theorem 22 happens to be tight, and our claim about IFA and BRA being close is further confirmed. The number of comparisons is $e(A-1)N$ in this case, as under IFA. This confirms the fact that on average Full-BRA uses slightly more comparisons that BRA. Note that confidence intervals are even smaller (this could have been predicted since each step uses more random variables and some averaging takes place), and again, they do not grow with N.

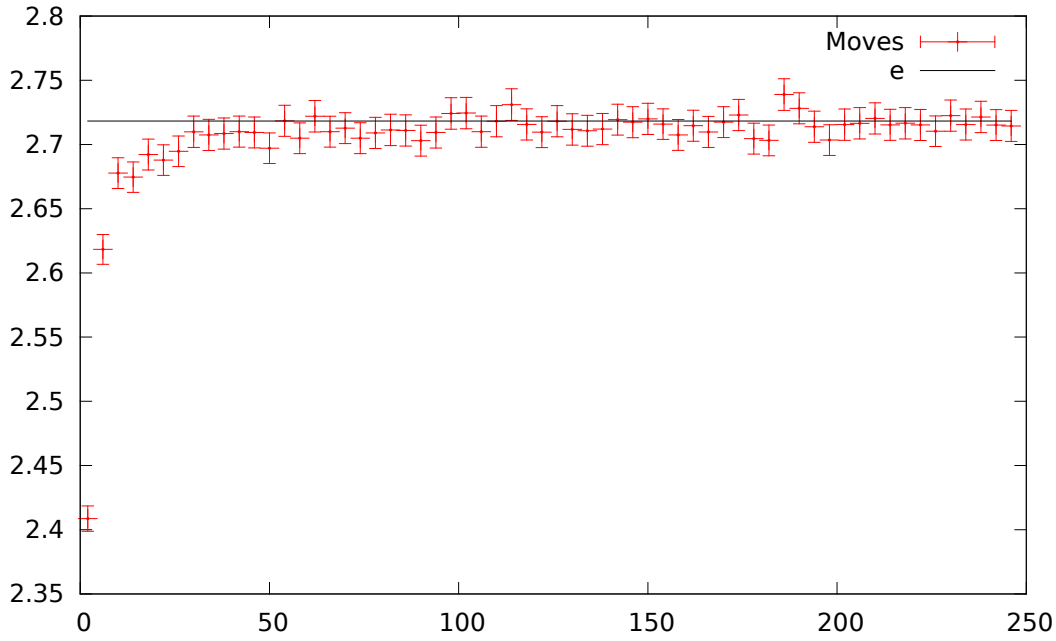


Figure 4.8: Average number of steps for full-BRA.

Chapter 5

Asynchronous Dynamics

This chapter is the longest of the thesis and contains the most advanced result. After a description of the distributed context for best response dynamics and the corresponding algorithm, we will then present the average complexity of the dynamics with low playing rates or with optimized playing rates. Following that will be the memoryless model used, then the details of both calculus.

5.1 Distributed Games

One of the original idea behind game theory is that the numerous players act independently for their own goals, even when we want to study the global result.

A natural consequence is that many cases of applications will feature numerous unit acting without much communication or synchronization.

In a congestion game with packets as players, the different players have no means of even knowing the existence of the other beyond the effects of said others and are also not synchronized by nature. In other network games, the players can communicate, but only limited informations and to direct neighbors

In these cases, the model the most natural is a distributed system in which that each players is active on the ping of an internal clock. In addition, the chosen model need do define what memory are specific to each player and what variables are instead globals, accessible in writing and reading by all. It also need to define the form and limits of the communication between players.

5.1.1 Poisson Clock

We chose to model the activation of player by a Poisson process. This model has a few advantages : Mainly the fact that the distribution law of the global sequence of players is also a Poisson process with a rate faster by a factor n and uniformly chosen players, instead of a more complex and non homogeneous distribution. This is one of the factors that allow us to use a Markovian approach.

Another point is that Poisson clocks are easy to implement and in many of the case we model through distributed game theory, a good representation of the different latencies.

As alternatives, we could have fixed delays for each players with or without an additional random noise latency. This would look more like an imperfect round robin when

treated globally. We could also considers clocks with very different rates depending on the players, or with a non homogeneous distribution.

5.1.2 Accessible data for the players

We assume that each player has a relatively big personal memory and access to a much lower global memory using concurrent read exclusive right and replacing communication.

We also assume that player can read the current state of the game, use it to compute internally the utilities they would get for each of their choices and edit the portion corresponding to their own choice. In chapter 6.1 we will see what happens if they can only read the current utility.

At first, in section 5.3 we will assume that the sequence of actions [reading, computing, editing] is atomic, i.e. that it happens fast enough to not be interrupted.

Later, in section **??** we will suppose it always takes the same deterministic time independently of the player and current state. One can notice that if it were instead a Poisson process or an arbitrary random variable, the fact that it will only be compared to Poisson values and sum of Poisson value make it behave the same.

5.2 Algorithm

Let us now consider a distributed version of Algorithm 1 where players act according to individual Poisson clocks with rate λ/n. Apart from this change of the sequence of plays, the algorithm is the same as Algorithm 1: L is the list of players that are satisfied with the current state (they played under the current state and did not change their action).

Algorithm 9: Distributed BRD (for player k)

Input: Game utilities $(u_k(\cdot))$; Initial state $(\mathbf{x} := \mathbf{x}(0))$;
Initialize $t := 0$; List of satisfied customers $L := \emptyset$;
Local clock, ticking w.r.t. a Poisson process with rate λ/n;
while *size(L)*$\neq n$ **do**
    On *each tick of the local clock*
        if $k \notin L$ **then**
            if $x_k = BR_k(\mathbf{x})$ **then**
                $L := L \cup \{k\}$
            else
                $x_k := \mathrm{BR}_k(\mathbf{x})$; $L := \{k\}$

In this distributed version of BRD, one can distinguish two phases. The first phase ends when the last change in the profile occurs (the last time when L is a singleton), at

this point the algorithm has reached a Nash equilibrium but does not know it yet. The second phase is the time needed for all players to play and check that their best response in the current state dos not change, thus certifying the Nash equilibrium (this is the time needed for L to grow up to its maximal size, n). We will denote by R (*reaching time*) the duration of the first phase, and by T (*execution time*) the total time taken by the two phases. We denote by δ the time taken by one player (say k) to compute its best response, $\mathrm{BR}_k(\mathbf{x})$, under state $\mathbf{x}$. We assume that this time does not depend on the player nor on the current state, but may depend on $a = A - 1$ the number of alternative actions that a player needs to check before deciding its best response (for example, it could be linear in a). More precisely, if for one player, say k, the clock ticks at time t, then it takes δ amount of time for the player to read the global state $\mathbf{x}$, compute its best response $\mathrm{BR}_k(\mathbf{x})$, and update its state to $x_k = \mathrm{BR}_k(\mathbf{x})$. We use the classical CREW-PRAM model for this distributed algorithm. The global variables are $\mathbf{x}$ and L, accessed by all players in a concurrent read and exclusive write mode. The PRAM model implies than the duration of play δ is constant over all times and all players, even if several plays overlap.

The main difference in the behavior of the distributed version of BRD vs the centralized one, is that the state may not be same for different players when their play overlap.

To make the point clear, let us consider a small example illustrated in Figure 5.1. The figure shows a time line of a game with three different players (say P_1, P_2, P_3), each with two possible actions and with clocks ticking respectively at times T_1, T_2, T_3, and their playing duration overlap as in Figure 5.1. Let us say that the initial state of the three players is $(0, 0, 0)$. Player P_1 plays first and changes its action for a better payoff to $x_1 = 1$. When player P_2 starts playing, P_1 has already finished its play, so P_2 compares the payoff of $(1, 0, 0)$ with $(1, 1, 0)$. Let us say that its best response is $x_2 = 1$. Meanwhile P_3 has started playing, because its play overlap with P_2. Its current state is still $(1, 0, 0)$, so P_3 compares the payoff of $(1, 0, 1)$ with $(1, 0, 0)$. Its best response is, say, $x_3 = 1$. This means that the state at time $T_3 + \delta$ will be $(1, 1, 1)$ whose potential has not been compared with the potential of $(1, 0, 0)$ (so it could be lower).

As shown in the previous example, overlaps can be seen as simultaneous plays and therefore, the potential may decrease in this case. However convergence to a Nash equilibrium still holds with probability one: Using the Borel Cantelli Lemma, a sequence of round robin plays of arbitrary length and with no overlap will eventually occur, ensuring convergence. Of course it may take a very long time before such an event occurs.

Figure 5.2 shows the evolution of the potential during the execution of Algorithm 9 over one potential game with 50 players. As one can see, potential does not always increase, unlike what happens in the classical centralized case of Algorithm 1. However, convergence occurs eventually (after 60 steps in that case).

In some cases, overlap can be guaranteed not to happen or non-limiting. We will show a Markovian model bounding the complexity and the computation of said complexity with

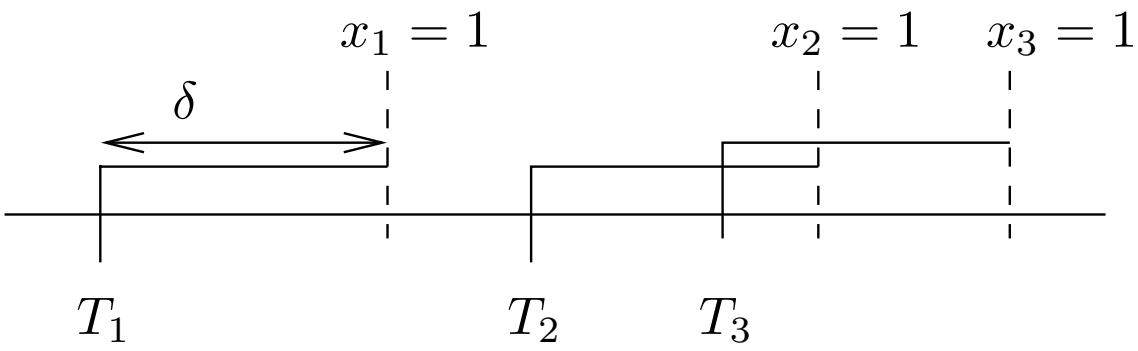


Figure 5.1: Time line of a distributed game when plays overlap

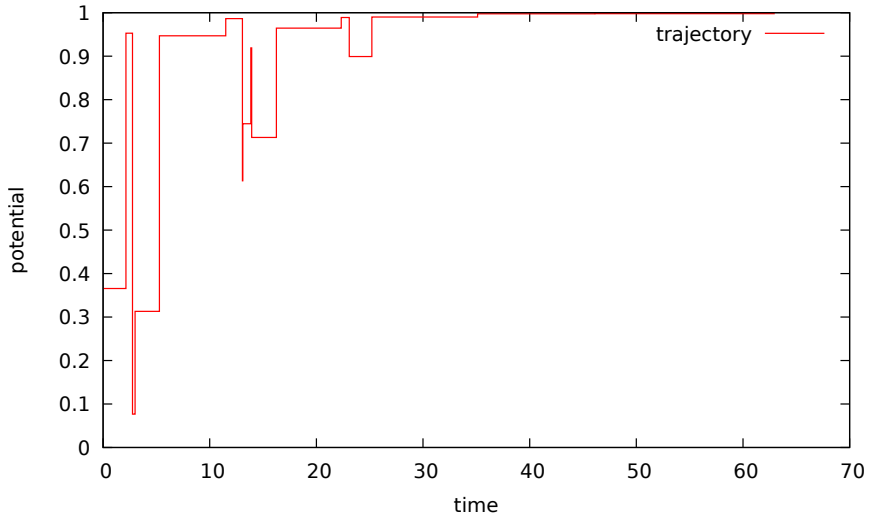


Figure 5.2: The evolution of the potential during the execution of Algorithm 9 up to the time when a Nash equilibrium is reached: Overlaps induce potential drops

a set Poisson rate and no overlap, then with overlap and choosing the rates to minimize execution time.

5.3 Low Playing Rates

The cases covered by this section can be interpreted in two ways : either the players can act instantaneously, or the rates of plays are slow enough so that the probability that a players acts too soon after another is negligible. It also cover the cases where a variant of the algorithm preventing overlap is used.

We measure the complexity of the algorithm as the number of steps before convergence. We can bound the expectancy of this number of step by

Theorem 23. *complexity without overlap* $T_{BRA} \leq 2e^{\gamma} n \log n + O(n)$

The proof will be given in Section 5.7.

We can remark that the bound is not tight. A way to see it is to consider the following: the algorithm is divided in two phases:

- The first phase potentials behave very similarly with the round robin case, with some steps lost as a satisfied player act, and some other less efficients due to intersections. One could expect the expected reaching time to be of the form $O(n)$.
- The second phase is a coupon collector problem (every players must play at least once after a given event) and takes $H_{n-1} = n \log n + n\gamma + o(n)$.

This would bring the complexity to $n \log n + O(n)$ with the constant in front of the $n \log n$ being 1.

A distributed algorithm using Poisson activations and without overlaps is equivalent to a centralized algorithm with a n times faster clock and uniformly choosing which player activate. We can recall the numerical analysis in figure **??**, which shoes that the reaching time is indeed linear, with a higher constant than Round Robin.

5.4 High Playing Rates

We know consider the frequency of the Poisson clock as an adjustable parameter, and write the global rate λ. We will write $p = 1 - e^{-\lambda\delta}$ the probability that the activation of a player is too soon after the last activation, causing an overlap.

Without the need to keep this probability negligible, the rates considered will be higher than in the previous study.

Intuitively, a faster rate means more overlaps and that the algorithm need more steps before convergence, whereas a slower rate increase the average time taken between steps and the convergence time.

The notation $O(f(p,n))$ means a term asymptotically smaller than $f(p,n)$ as n goes to infinity, uniformly in p. In other words, $h(p,n) \in O(f(p,n))$ if there exists a constant C such that $\lim_{n\to\infty} \sup_{p\in[0,1]} \frac{h(p,n)}{f(p,n)} \leqslant C$. The same convention applies to $o(f(p,n))$.

Theorem 24. *The expected execution time of the distributed best response dynamics with Poisson clocks of total rate λ as given in Algorithm 9 satisfies*

$$\frac{nH_n}{\lambda} \leqslant \mathbb{E}[T_{BRD}] \leqslant \frac{1}{\lambda}\left(n^{1-p}H_{n-1} + \frac{n^{1-p} + C_1 + \frac{C_2}{n} + C_2\frac{G(p,n)}{n(1-p)}}{1-p} + 1\right) G(p,n). \quad (5.1)$$

where $G(p,n) = \left(\frac{e^{pH_{n-1}}}{p} - \frac{1-p}{p}\right)$, H_n is the nth harmonic number, $p = 1 - e^{-\delta\lambda}$ is the overlap probability, and C_1 and C_2 are constants.

A bound uniform in p can be given for $G(p,n)$: $\forall p \in [0,1], \forall n \geqslant 2$, $G(p,n) \leqslant e(n-1)$. Indeed, with $n-1 = 1/e$, $G(p,1/e) \leqslant 1$, and $\frac{\partial(G(p,n)-e(n-1))}{\partial n} \leqslant 0$ for all $n \geqslant 2$. This yields a first simple bound for $\mathbb{E}[T_{BRD}]$:

$$\mathbb{E}[T_{BRD}] \leqslant \frac{e}{\lambda}\left(n^2\log(n) + \frac{n^2}{(1-p)^2}\right)(1+o(1)).$$

Let us denote by $B(n,p)$ the upper bound given in the theorem (up to the factor $1/\lambda$).

$$B(n,p) := \left(n^{1-p}H_{n-1} + \frac{n^{1-p} + C_1 + \frac{C_2}{n} + C_2\frac{G(p,n)}{n(1-p)}}{1-p} + 1\right) G(p,n).$$

- When p goes to 0, $B(n,p)$ goes to $e^\gamma nH_{n-1}^2 + O(nH_{n-1})$. This is a bound on the number of steps before convergence in the ideal case when $\delta = 0$. Actually, a tighter bound can be computed in this case: When $p = 0$, a direct proof (not reported here) shows that $\mathbb{E}[S_{BRD}] \leq e^\gamma nH_{n-1} + O(n)$.

- When p goes to 1, $B(n,p)$ goes to infinity, as expected: when all plays overlap, convergence in finite time cannot be guaranteed.

- When $p \in (0,1)$ is fixed and n goes to infinity, we have

$$B(n,p) = \frac{e^\gamma}{p}\left(n\log(n) + \frac{n}{1-p}\right)(1+o(1)).$$

  This is obtained recalling that $\log(n-1)+\gamma+\frac{1}{2n} \leq H_{n-1} \leq \log(n-1)+\gamma+\frac{1}{2(n-1)}$, and hence $e^{pH_{n-1}} \leq (n-1)^p e^{p\gamma+\frac{p}{2(n-1)}}$.

Finally, letting p depend on n yields the following result.

Corollary 25. *Under a playing rate that minimizes the upper bound asymptotically, namely* $\hat{\lambda} = \frac{\log(\log(n)) - \log\log\log(n))}{\delta}$, $\hat{p} = 1 - \frac{\log(\log(n))}{\log(n)}$, *the average execution time satisfies*

$$\frac{\delta n \log(n)}{\log(\log(n))}(1 + o(1)) \leqslant \mathbb{E}[T_{BRD}] \leqslant e^{\gamma} \frac{\delta n \log(n)}{\log(\log(n))}(1 + o(1)) \tag{5.2}$$

where γ *is the Euler constant.*

As a note, one can notice that $\hat{p}$ tend to 1 as n goes to infinity. This goes against the intuition that an algorithm would need very few exceptions to function.

5.5 Intersection-Free Approximation on Distributed system

The goal of this section is to link the execution of the Best Response Dynamics on randomly chosen game to the Intersection Free Approximation and prove the lemma :

This lemma will apply in the low playing rates case and the high playing rates case, with different computation for T_{IFA} and different values for I .

5.6 Proof for High Playing Rates

This section focuses on the high playing rate case and contains the proof of theorem 24.

5.6.1 Comparison with IFA

Lemma 26. *Comparison with IFA*

$$\mathbb{E}S_{BRD} \leqslant \mathbb{E}S_{IFA} + \frac{C_1}{1-p} G(p,n) + C_2 \frac{G(p,n)}{n(1-p)} + C_2 \frac{G(p,n)^2}{n(1-p)^2},$$

where $G(p,n) = \left(\frac{e^{pH_{n-1}}}{p} - \frac{1-p}{p}\right)$, H_n *is the nth harmonic number,* $p = 1 - e^{-\delta\lambda}$ *is the overlap probability, and* C_1 *and* C_2 *are constants.*

Proof Lemma 12, shows that

$$\mathbb{E}S_{BRD} \leqslant \mathbb{E}S_{IFA} + \frac{I}{A}$$

Equation 4.2 says that

$$\mathbb{E}I \leq C_1 \mathbb{E}(M) + \frac{C_2}{n} \mathbb{E}(M^2)$$

The rest of the proof consists in a bound on the first and second moments of the number of moves

The analysis of the expected number of moves, follows the same path as for the execution time.

First, one needs to show that the number of moves is larger under the IFA approximation. This is done by using the same coupling as for the number of steps. Then, one also needs to show that the number of moves is larger under the Restart approximation. Again, the proof is done by coupling IFA and Restart. By the same coupling argument as in the proof of $V_{BRD} \leqslant_{st} V_{IFA}$, we can show that $M_{BRD} \leqslant_{st} M_{IFA} \leqslant_{st} M_{RST}$. Finally, it is also easy to see that the case $A = 2$ is an upper bound on the case where $A \geqslant 2$.

The expected number of moves for Restart satisfies a Poisson equation similar to the equation for the number of steps:

Let us denote by $m(y, k)$ the average number of moves before reaching a Nash equilibrium, starting with potential y and k satisfied players. One-step analysis yields:

$$\begin{aligned}
m(y,n) &= 0 \\
m(y,i) &= \frac{i}{n}m(y,i) + \frac{n-i}{n}ym(y,i+1) \\
&\quad + \frac{n-i}{n}\int_y^1 (m(u,1)+1)\mathrm{d}u, \quad n > i > 1; \\
m(y,1) &= \frac{1}{n}m(y,1) + \frac{n-1}{n}(1-p)ym(y,2)) \\
&\quad + \frac{n-1}{n}(1-p)\int_y^1 (m(u,1)+1)\mathrm{d}u \\
&\quad + \frac{n-1}{n}p\int_0^1 m(u,1)\mathrm{d}u.
\end{aligned}$$

By simplifying, we get

$$\begin{aligned}
m(y,n) &= 0 \\
m(y,i) &= 1 - y + ym(y,i+1) \\
&\quad + \int_y^1 m(u,1)\mathrm{d}u, \quad n > i > 1; \\
m(y,1) &= 1 - y + yp - p + y(1-p)m(y,2) \\
&\quad + (1-p)\int_y^1 m(u,1)\mathrm{d}u \\
&\quad + p\int_0^1 (m(u,1)+1)\mathrm{d}u.
\end{aligned}$$

We define $M(z) := \int_z^1 m(u,1)\mathrm{d}u$ and use telescoping so that the previous equations become a single ordinary differential equation with implicit initial condition $M(0)$:

$$m(y,1) = (1-p)\sum_{j-0}^{n-2} y^i M(y) + (1-p)\sum_{j-0}^{n-2} y^i(1-y) + yp + pM(0).$$

This equation admits an explicit solution for $M(0)$, using the same function $q(u)$ as before.

$$\begin{aligned} M(0) &= \frac{\int_0^1((1-p)(1-u^{n-1})+pu)e^{q(u)}\mathrm{d}u}{1-p\int_0^1 e^{q(u)}\mathrm{d}u} \\ &\leqslant \frac{(1-p)Q_0+pQ_1}{1-pQ_0} \\ &\leqslant \frac{Q_0}{1-pQ_0}. \end{aligned}$$

Using the bound $Q_0 \leq \frac{1}{p} - \frac{1-p}{p}e^{-pH_{n-1}}$, given in (5.13),

$$M(0) \leqslant \frac{1}{1-p}\left(\frac{e^{pH_{n-1}}}{p} - \frac{1-p}{p}\right) = \frac{1}{1-p}\,G(p,n). \tag{5.3}$$

As for the second moment, let $m_2(y,1) = \mathbb{E}(M(y,k)^2)$ denote the second moment of the number of moves, starting with potential y and k satisfied players. One step analysis yields the following Poisson equation, involving both m_2 and m.

$$\begin{aligned} m_2(y,n) &= 0 \\ m_2(y,i) &= ym_2(y,i+1) \\ &\quad + \int_y^1 (m_2(u,1)+2m(u,1)+1)\mathrm{d}u, \quad n>i>1 \\ m_2(y,1) &= y(1-p)m_2(y,2) \\ &\quad +(1-p)\int_y^1 (m_2(u,1)+2m(u,1)+1)\mathrm{d}u \\ &\quad +p\int_0^1 (m_2(u,1)+2m(u,1)+1)\mathrm{d}u \end{aligned}$$

By telescoping all $m_2(y,i), i>1$ and using $M_2(z) := \int_z^1 m_2(u,1)\mathrm{d}u$,

$$\begin{aligned} m_2(y,1) &= (1-p)\sum_{j=0}^{n-2} z^j M_2(z) \\ &\quad +(1-p)\sum_{j=0}^{n-2} z^j((1-z)+2M(z)) \\ &\quad +p+2pM(0)+pM_2(0). \end{aligned}$$

This yields an integral form for $M_2(0)$

$$M_2(0 = \frac{\int_0^1 e^{q(z)}(p+2pM(0)+(1-p)(1-z^{n-1})+2(1-p)\sum_{j=0}^{n-2} z^j M(z))\mathrm{d}z}{1-p\int_0^1 e^{q(u)}\mathrm{d}u}$$

Using the fact that $M(z) = \int_z^1 m(u,1)\mathrm{d}u$, and $m(u,1)$ is decreasing, we get $M(z) \leqslant (1-z)\int_0^1 m(u,1)\mathrm{d}u = (1-z)M(0)$. This implies that

$$\begin{aligned} M_2(0) &\leqslant \frac{Q_0 + 2Q_0M(0)}{1-pQ_0} \\ &\leqslant \frac{Q_0}{1-pQ_0} + 2\left(\frac{Q_0}{1-pQ_0}\right)^2 \\ &\leqslant \frac{1}{1-p}G(p,n) + \frac{2}{(1-p)^2}G(p,n)^2 \end{aligned}$$

where the second inequality uses (5.3).

We then have :

$$\mathbb{E}(M) \leq \frac{1}{1-p}G(p,n), \tag{5.4}$$

and

$$\mathbb{E}(M^2) \leq \frac{1}{1-p}G(p,n) + \frac{2}{(1-p)^2}G(p,n)^2. \tag{5.5}$$

This end the proof of the lemma.

5.6.2 Restart Approximation (RST)

The future behavior of IFA only depends on the current potential $\Phi_{IFA}(t)$ and the current set $L_{IFA}(t)$, as long as there are no overlaps. In the actual game, when an overlap takes place, as seen in Figure 5.1, the current player will not consider the state computed by the previous player but the state obtained by the latest player with whom it is not overlapping.

The behavior of $X_{IFA}(t) \stackrel{\text{def}}{=} (\Phi_{IFA}(t), L_{IFA}(t))$ is not Markovian as such.

We construct yet another approximation, of the behavior of IFA this time, whose state transition $X_{RST}(t) \stackrel{\text{def}}{=} (\Phi_{RST}(t), L_{RST}(t))$ is different from $X_{IFA}(t) = (\Phi_{IFA}(t), L_{IFA}(t))$ only when overlaps occur. This approximation will be called a *restart* version of IFA (denoted RST in the following). At any Poisson point (called t), if the current play overlaps with the previous one, who has been active (*i.e.* has changed its action), then the state is reset to a uniform potential in $[0,1]$, disregarding the previously obtained potential and the set L is reset to a single player (the latest one).

More precisely, if $X_{RST}(t^-) = (\Phi, L)$ and the next player k' overlaps the previous play, then, $X_{RST}(t) = (U, \{k'\})$, where $U \sim \mathrm{Unif}[0,1]$. In all other cases, restart behaves exactly as IFA : if the number of satisfied player is not one, that means that the last player kept the same choice, and even if the new player activation overlap, the information used by her is accurate.

Note that by definition, $|L_{RST}(t)| = 1$ if $|L_{IFA}(t)| = 1$ for all t. Only the current potential differs when both sets are singletons.

The goal of the next Lemma is to compare the behavior of the two systems, and more precisely $\Phi_{IFA}(t)$ and $\Phi_{RST}(t)$.

Lemma 27. *For all t, $\Phi_{IFA}(t) \geqslant_{st} \Phi_{RST}(t)$.*

Proof. We compare IFA and RST by coupling the instants of plays of all players (denoted $(t_k)_{k\in\mathbb{N}}$), the order of play as well as the overlaps in both systems. One can refer to Figure 5.1 that displays the time line with one overlap.

We will keep the same notations for the original systems and the coupled systems. The proof holds by induction on t_k (the next playing instant). Let t_3 the first time that an overlap occurs while the set of satisfied players is a singleton, as in Figure 5.1. Up to time t_2, both systems coincide under the previous coupling. We have $\Phi_{IFA}(t_1) = \Phi_{RST}(t_1)$. To deal with the general case directly, let us only use instead the weaker property $\Phi_{IFA}(t_1) \geqslant \Phi_{RST}(t_1)$, that will propagate by induction.

Under restart, the value of the potential U after any play in overlap (here, at time $t_3 + \delta$) is uniformly distributed in $[0, 1]$. In IFA, the player acting at time t_2 has changed its action. Therefore the potential $\Phi_{IFA}(t_2)$ must be higher than the potential $\Phi_{IFA}(t_1)$. By definition of IFA, this potential is the maximum of a uniformly distributed variables in the interval $[0, 1]$, conditioned by the fact that it is larger than $\Phi_{IFA}(t_1)$. This implies that $\Phi_{IFA}(t_2) \geqslant_{st} U$, U being uniform in $[0, 1]$.

Now let us consider the player acting at time t_3. It uses the state of the system at time t_2 to compute its best response. If the potential of its best response is not larger than $\Phi_{IFA}(t_2)$, which happens with positive probability, then this player does not change its action and the state at time $t_3 + \delta$ remains the state computed by the previous player: the potential is $\Phi_{IFA}(t_3) = \Phi_{IFA}(t_2) \geqslant_{st} U$. On the other hand, if the potential of its best response is larger than $\Phi_{IFA}(t_2)$, then it changes its action and the system reaches a new state where two players act simultaneously, whose potential under IFA $\Phi_{IFA}(t_0)$, is uniformly distributed in $[0, 1]$. In total by considering both cases, $\Phi_{IFA}(t_0) \geqslant_{st} U$.

Now, if several overlaps occur in succession, the same type of reasoning concludes that with a positive probability (smaller and smaller as the number of overlaps increases) the potential remains the same as the potential of the first player, which is stochastically larger than a uniform one, and in the complementary case, the potential is newly generated with a uniform distribution.

In all cases, the potential remains stochastically larger than a uniform one, hence larger than the potential under RST. □

Corollary 28. *The time to reach a full set L is stochastically smaller under IFA than under the restart approximation.*

Proof. This result is a direct consequence of the previous lemma. Indeed, the order of play is the same for both systems as well as the occurrence of overlaps and since, at any time, the potentials are comparable, then if L_{IFA} is full at time t, L_{RST} must also be full at the same instant. Therefore, the first times of being full (implying termination in both systems) compare stochastically. □

The construction of the restart approximation ensures that the future behavior of the system under RST only depends on its current state, namely the current potential and the list of satisfied players. It should also be clear that all states with the same number of satisfied players can be aggregated and the behavior remains Markovian, because the next player to play is IID, uniformly distributed among all players and does not depend on the previous plays. Therefore, in the following we focus on the evolution of the couple $(\Phi(t), |L(t)|)$ (potential, number of satisfied players).

5.6.3 Complexity of Restart

The sections above show that the average time for the original distributed best response algorithm with Poisson clocks to compute a Nash equilibrium is bounded from above by the time taken under the restart approximation.

The rest of this section is devoted to the computation of the average duration of RST.

Since all players play according to independent Poisson processes with rates λ/n, the global point process of active players $(T_i)_{i\in\mathbb{N}}$, forms a Poisson process with rate λ. In the following we focus on the behavior of the system at those points (called "instants" in the following).

It should be clear by now that $(\Phi(T_i), |L(T_i)|)$ is a discrete time Markov chain with a hybrid state space (a discrete component and a continuous one: the state space is $\{1, \cdots, n\} \times [0, 1]$).

The average hitting time of the set of states $\{|L| = n)\}$ starting from a state with potential y and k satisfied players (including the current one) is given by a system of Poisson equations derived from a one-step analysis:

Let us denote by $s(y, k)$ the expected instant when a full set of satisfied players is reached (this corresponds to the termination of RST), starting with potential y and k satisfied players. One-step analysis yields the following equations, where $p = 1 - e^{-\lambda\delta}$ is the probability that the current play overlaps the previous one (since the playing times of all players form a Poisson process with rate λ).

$$
\begin{aligned}
s(y,n) &= 0 \\
s(y,i) &= 1 + \frac{i}{n}s(y,i) + \frac{n-i}{n}y^a s(y,i+1) + \frac{n-i}{n}\int_y^1 a u^{a-1} s(u,1)\mathrm{d}u, \quad n > i > 1 \\
s(y,1) &= 1 + \frac{n-1}{n}y^a(1-p)s(y,2) + \frac{n-1}{n}(1-p)\int_y^1 a u^{a-1} s(u,1)\mathrm{d}u \\
&\quad + \frac{n-1}{n}p\int_0^1 s(u,1)\mathrm{d}u + \frac{1}{n}s(y,1)
\end{aligned}
$$

These equations are derived using the behavior of restart. Starting in point $(y, i), i > 1$, the previous player did not modify its action so overlapping the current action with the previous one will not make any difference. This is why the equation for $s(y, i), i > 1$ does

not depend on overlaps and hence does not involve p. Now, upon a play of a new player (this happens with probability $\frac{n-i}{n}$), the current profile will not change if this player does not find an action whose potential is above the current potential, namely y. This happens with probability $\frac{n-i}{n}y^a$. If the new player finds an action with potential $u > y$, then the current player becomes the only satisfied player and the potential goes to u. Therefore, the Markov chain jumps to $(u, 1)$. This happens with probability $\frac{n-i}{n}au^{a-1}\mathrm{d}u$.

When the starting point is $(y, 1)$, the situation is slightly more complex because overlaps do play a role.

Let us first consider the case with no overlap. This happens with probability $1 - p$. In this case, the behavior is similar to the case $(y, i), i > 1$. Now, when an overlap occurs, the potential is always reset to a uniform potential in $[0, 1]$. This is precisely the effect of the restart approximation, and it happens with probability $\frac{n-1}{n}p$.

Defining $z := y^a$, $s^{(1)}(z, i) := s(z^{\frac{1}{a}}, i)$, one gets

$$\begin{aligned} s^{(1)}(z,n) &= 0; \\ s^{(1)}(z,i) &= \frac{n}{n-i} + \int_z^1 s^{(1)}(u,1)\,\mathrm{d}u + z s^{(1)}(z,i+1); \\ s^{(1)}(z,1) &= \frac{n}{n-1} + p\int_0^1 s(u,1)\,\mathrm{d}u \\ &\quad + (1-p)\int_z^1 s^{(1)}(u,1)\mathrm{d}u + (1-p)z\, s^{(1)}(z,2). \end{aligned}$$

Rewriting the last equation as

$$s^{(1)}(z,1) = p\frac{n}{n-1} + p\int_0^1 s(u,1)\,\mathrm{d}u + (1-p)\left[\frac{n}{n-1} + \int_z^1 s^{(1)}(u,1)\mathrm{d}u + z\, s^{(1)}(z,2)\right]$$

and telescoping all other equations into it, we obtain the following equation only involving $s^{(1)}(z, 1)$.

$$s^{(1)}(z,1) = \frac{pn}{n-1} + +p\int_0^1 s(u,1)\,\mathrm{d}u + (1-p)\sum_{j=0}^{n-2}\frac{n}{n-j-1}z^j + (1-p)\sum_{j=0}^{n-2} z^j \int_z^1 s^{(1)}(u,1)\,\mathrm{d}u. \quad (5.6)$$

We define

$$S(z) := \int_z^1 s^{(1)}(u,1)\mathrm{d}u.$$

$S(0)$ is the quantity we want to bound. Notice that $S'(z) = -s^{(1)}(z,1)$, hence (5.6) gives an ordinary differential equation for $S(z)$:

$$S'(z) + (1-p)\left[G(z)S(z) + V(z)\right] + pC_0 + D = 0,$$

where

$$G(z) := \sum_{i=0}^{n-2} z^i = \frac{1-z^{n-1}}{1-z} \tag{5.7}$$

$$V(z) := \sum_{j=0}^{n-2} \frac{nz^j}{n-j-1} \tag{5.8}$$

$$C_0 := \int_0^1 s(u,1) \tag{5.9}$$

$$D := \frac{p\,n}{n-1} \tag{5.10}$$

We solve this equation on the interval $[0,1]$ using the boundary condition $S(1) = 0$. We obtain:

$$S(z) = -e^{-q(z)} \int_1^z \left((1-p)V(u) + pC_0 + D\right) e^{q(u)} \mathrm{d}u.$$

where

$$q(z) := (1-p)\int_0^z G(u)\,\mathrm{d}u = (1-p)\sum_{j=1}^{n-1} \frac{z^j}{j}.$$

Since $q(0) = 0$, this gives

$$S(0) = (1-p)\int_0^1 V(u)e^{q(u)}\mathrm{d}u + (pC_0 + D)\int_0^1 e^{q(u)}\mathrm{d}u. \tag{5.11}$$

Notice that, for $z \in [0,1]$, $z^{\frac{1}{a}} \geq z$, and hence $s(z,1) \leq s^{(1)}(z,1)$.
This makes $C_0 = \int_0^1 s(z,1)\mathrm{d}u \leq \int_0^1 s^{(1)}(z,1)\mathrm{d}u = S(0)$.

Replacing this bound in (5.18), and posing

$$Q_j := \int_0^1 u^j e^{q(u)}\mathrm{d}u.$$

, we obtain

$$(1 - pQ_0)\,S(0) \leq (1-p)\int_0^1 V(u)e^{q(u)}\mathrm{d}u + \frac{n}{n-1}pQ_0, \tag{5.12}$$

To study Q_0, we separate the integral in two terms. The first term is

$$\int_0^{1-b} e^{q(u)}\mathrm{d}u \leq \int_0^{1-b} e^{(1-p)\sum_{j=1}^{\infty}\frac{u^j}{j}}\mathrm{d}u = \int_0^{1-b}(1-u)^{p-1}\mathrm{d}u = \frac{1-b^p}{p}.$$

The second term is

$$\int_{1-b}^{1} e^{q(u)}\mathrm{d}u \leq \int_{1-b}^{1} e^{(1-p)\sum_{j=1}^{n-1}\frac{1}{j}}\mathrm{d}u = be^{(1-p)H_{n-1}}.$$

Hence we have

$$Q_0 \leq \frac{1}{p} - \frac{b^p}{p} + be^{(1-p)H_{n-1}}.$$

This bound is true for all $b \in (0,1)$. By using the best value for b, namely $b = e^{-H_{n-1}}$, we get

$$Q_0 \leq \frac{1}{p} - \frac{1-p}{p}e^{-pH_{n-1}} \tag{5.13}$$

and hence $1 - p\,Q_0 \geqslant (1-p)e^{-pH_{n-1}}$.

To bound Q_j let us first show that $Q_j \leq \frac{j}{j+p}Q_{j-1}$.

First consider $jQ_{j-1} = \int_0^1 ju^{j-1}e^{q(u)}\mathrm{d}u$. Integrating by parts, we have

$$\begin{aligned} jQ_{j-1} &= \left[u^j e^{q(u)}\right]_0^1 - \int_0^1 q'(u)u^j e^{q(u)}\mathrm{d}u \\ &= e^{(1-p)H_{n-1}} - (1-p)\sum_{i=0}^{n-2}\int_0^1 u^{j+i}e^{q(u)}\mathrm{d}u. \end{aligned}$$

Then consider $(j+p)Q_j = (j+1)Q_j - (1-p)Q_j$. For the term $(j+1)Q_j$ we do the same integration by parts as above, and hence we obtain

$$\begin{aligned} &(j+p)Q_j = (j+1)Q_j - (1-p)Q_j \\ &= \left[u^{j+1}e^{q(u)}\right]_0^1 - \int_0^1 q'(u)u^{j+1}e^{q(u)}\mathrm{d}u - (1-p)\int_0^1 u^j e^{q(u)}\mathrm{d}u \\ &= e^{(1-p)H_{n-1}} - (1-p)\sum_{i=0}^{n-2}\int_0^1 u^{j+i+1}e^{q(u)}\mathrm{d}u - (1-p)\int_0^1 u^j e^{q(u)}\mathrm{d}u \\ &= e^{(1-p)H_{n-1}} - (1-p)\sum_{i=0}^{n-1}\int_0^1 u^{j+i}e^{q(u)}\mathrm{d}u \\ &= jQ_{j-1} - (1-p)\int_0^1 u^{j+(n-1)}e^{q(u)}\mathrm{d}u \end{aligned}$$

We have obtained $(j+p)Q_j \leq jQ_{j-1}$, which gives

$$Q_j \leq \frac{j}{j+p}Q_{j-1} \leq \left(\prod_{\ell=1}^{j}\frac{\ell}{\ell+p}\right)Q_0. \tag{5.14}$$

We introduce the following technical lemma that is useful to get an explicit bound for Q_j.

Lemma 29. *For all* $0 \leq p \leq 1$, $\prod_{\ell=1}^{j} \frac{\ell}{\ell+p} \leq (j+1)^{-p}$.

Proof. Proof. Let us define the function

$$F : p \mapsto \sum_{\ell=1}^{j} \log\left(\frac{\ell+p}{\ell}\right) - p\log(j+1).$$

Proving the lemma is equivalent to proving that $F(p) \geq 0$ for all $p \in [0,1]$. The second derivative of F is $F''(p) = \sum_{\ell=1}^{j} \frac{-1}{(\ell+p)^2}$, non-positive in $[0,1]$ implying that F is concave on $[0,1]$ and hence its minimum is either in 0 or in 1. $F(0) = 0$ and $F(1) = \sum_{\ell=1}^{j} \log\left(\frac{\ell+1}{\ell}\right) - \log(j+1) = -\log 1 + \log(j+1) - \log(j+1) = 0$, so F is non-negative on $[0,1]$.

□

Using Lemma (29), and $Q_j \leq \left(\prod_{\ell=1}^{j} \frac{\ell}{\ell+p}\right) Q_0$, we obtain

$$Q_j \leq (j+1)^{-p} Q_0$$

We are now ready to find a bound for the numerator in (**??**).

$$\begin{aligned}
(1-p)\sum_{j=0}^{n-2} \frac{nQ_j}{n-j-1} &\leq (1-p)Q_0 \sum_{j=0}^{n-2} \frac{n}{n-j-1}\frac{1}{(j+1)^p} \\
&= (1-p)Q_0 \sum_{j=1}^{n-1} \frac{(n-j)+j}{n-j}\frac{1}{j^p} \\
&= (1-p)Q_0 \left(\sum_{j=1}^{n-1} j^{-p} + \sum_{j=1}^{n-1} j^{1-p}\frac{1}{n-j}\right)
\end{aligned}$$

Finally, we notice that

$$\sum_{j=1}^{n-1} j^{-p} \leq \int_0^n x^{-p}\mathrm{d}x = \frac{n^{1-p}}{1-p}$$

and

$$\sum_{j=1}^{n-1} j^{1-p}\frac{1}{n-j} \leq (n-1)^{1-p}\sum_{j=1}^{n-1}\frac{1}{n-j} = (n-1)^{1-p}H_{n-1}.$$

5.6.4 Proof of Theorem 24

Equation (**??**) gives an upper bound for the average number of steps of a Nash equilibrium for the Restart approximation. Using Lemmas 12 and 27, we get an upper bound on the average number of steps taken by BRD to compute a Nash equilibrium.

Let S_{BRD} (resp. S_{IFA}, S_{RST}) be the number of steps in BRD (resp. IFA, RST) before termination.

$$\begin{aligned}
\mathbb{E}[S_{BRD}] &\leq \mathbb{E}[S_{IFA}] + C_1\frac{G(p,n)}{1-p} + C_2\frac{G(p,n)}{n(1-p)} + C_2\frac{G(p,n)^2}{n(1-p)^2} \\
&\leq \mathbb{E}[S_{RST}] + C_1\frac{G(p,n)}{1-p} + C_2\frac{G(p,n)}{n(1-p)} + C_2\frac{G(p,n)^2}{n(1-p)^2} \\
&\leq \left(n^{1-p}H_{n-1} + \frac{n^{1-p} + C_1 + \frac{C_2}{n} + C_2\frac{G(p,n)}{n(1-p)}}{1-p} + 1 \right) G(p,n).
\end{aligned}$$

The expected time before getting to a Nash equilibrium is given by the formula $\mathbb{E}[T_{BRD}] = \frac{1}{\lambda}\mathbb{E}[S_{BRD}]$, that follows from Wald's lemma.

5.6.5 Rate Optimization: Proof of Corollary 25

One can choose the playing rate λ (or the overlap probability p) to minimize the expected time needed to obtain a Nash equilibrium. The overlap probability is $p = 1 - \exp(-\lambda\delta)$, so that $\lambda\delta = -\ln(1-p)$.

Since we only know a bound on $\mathbb{E}[T_{BRD}]$, we can instead compute the overlap probability $p^*(n)$ that minimizes that bound.

The optimal value can be estimated by differentiating and numerically solving $\frac{\partial \frac{-1}{\ln(1-p)} B(n,p)}{\partial p} = 0$. This was done using Maple. The behavior of $p^*(n)$ as n grows, is displayed in Figure 5.3. It shows that $p^*(n)$ is non-decreasing with n and grows from $p^*(3) \approx 0.65$ to $p^*(250) \approx 0.78$.

Analyzing the main term in the bound $B(n,p)$, it is direct to show that $p^*(n)$ goes to 1 as n goes to infinity.

Since $p^*(n)$ does not have a closed form, one can instead minimize the asymptotic behavior of $B(n,p)$. When p is close to 1 and n is large, the dominating terms are $n^{2p-1}/(1-p)^2$ and $n/(1-p)$ on one side and $n\log n$ on the other. This suggests choosing $\hat{p} = 1 - \log\log(n)/\log(n)$, with a corresponding playing rate satisfying $\delta\hat{\lambda} = \log(\log(n)) - \log\log\log n$. The average execution time under $\hat{p}$ is now such that

$$\mathbb{E}[T_{BRD}] \leqslant e^{\gamma}\delta\frac{n\log(n)}{\log(\log(n))}(1+o(n)) \tag{5.15}$$

where $e^{\gamma} \approx 1.78$.

Finally, it should be clear that in BRD each player needs to play at least once to check if a state is a Nash equilibrium. The coupon collector theorem says that on average nH_n plays are necessary before all players have played once. Under rate $\hat{\lambda}$, each play takes $1/\hat{\lambda}$ units of time on average. This implies that

$$\mathbb{E}[T_{BRD}] \geqslant \delta\frac{n\log(n)}{\log(\log(n))}(1+o(n)).$$

This ends the proof of Theorem 24 and of its corollary.

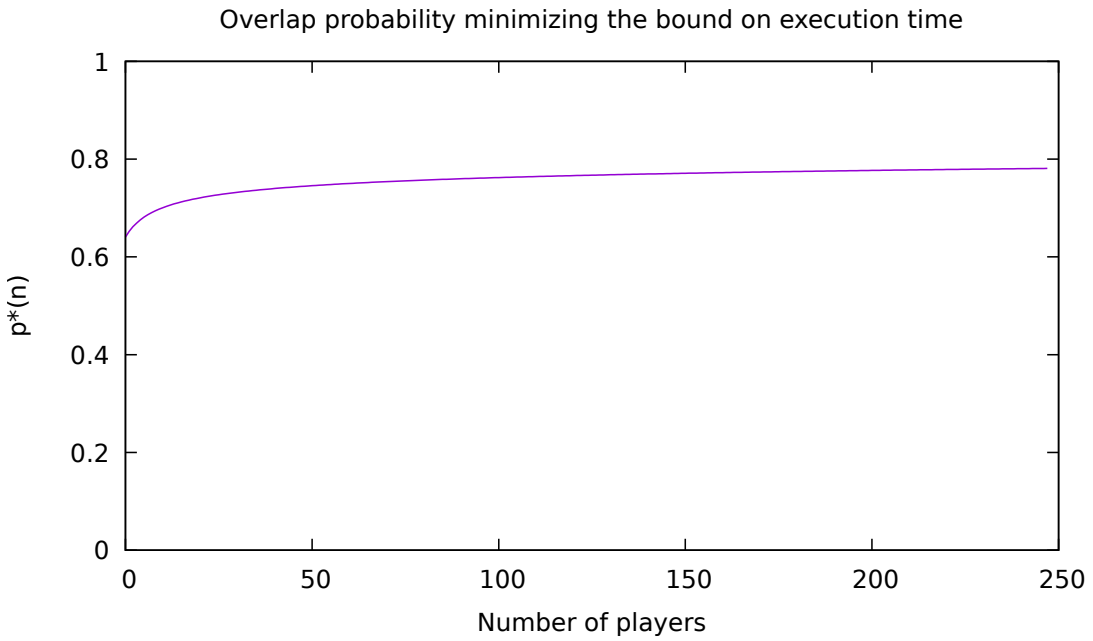


Figure 5.3: Overlap probability $p^*(n)$ minimizes the bound $\frac{1}{\lambda}B(n,p)$ on the expected convergence time $\mathbb{E}[T_{BRD}]$.

5.7 Proof for Low Playing Rates

As first comment, we can reuse the result from Theorem **??**(corolaire HPR) in the case $p = 0$, which gives us $ES_{BRD} \leq en \log^2 n + O(n)$.

We can also use the round robin result **??** and get

Proposition 30. $\mathbb{E}T_{IID} \leq \mathbb{E}(\lceil T_{RR} \rceil)n = (e^\gamma + \frac{1}{2})n^2 + o(n)$

Proof. A distributed execution with no overlap is equivalent to a sequential execution where each player is chosen uniformly among all possible and the time of the activations correspond to a Poisson clock of rate n times faster.

We can use this to obtain a bound on the expected reaching and convergence time.

Let us consider the sequence of chosen players. We can cut this sequence in the shortest pieces such that each piece including each player at least once in the same order. The average size of the first such piece is n times the n^{th} harmonic sum, equivalent to $n \log n$ when n goes to infinity. The average size of the following pieces is n^2

We then apply the intersection free approximation. On each piece, the algorithm behave similarly to a round of round robin with the players in the order of first appearance in the piece, with additional moves those effect can only increase the potential and the size of the satisfied list.

The second increase only matter if it force a Nash equilibrium earlier. It is also easy to see in the computation for the round robin that the expected reaching time decrease with the potential. Neglecting these additional moves allow us to have an upper bound on the expectation of the reaching and convergence time. We can also fully neglect the intersections caused by and happening during those steps as their normal effect would have to ignore part of the effect of the step while we ignore all.

We obtained a pseudo round-robin under IFA sequence, with each block of n round robin step costing an average of n^2 steps.

Using the bound on round robin IFA we can compute the expected number of block needed before convergence. Since we can ignore the intersection happening outside of the pseudo round-robin extracted sequence, we have the same number of intersection as round robin, allowing to switch back to BRA.

Reaching and confirming a Nash equilibrium (ie obtaining n steps without moves) is a stopping time. We can apply the Wald identity to obtain the expectation of the number of steps needed under uniform revision sequence, as the sum of the expectations of the number of steps per piece.

□

A tighter result can be obtained, as stated in Theorem 23 :

$$T_{BRA} \leq 2e^{\gamma} n \log n + O(n)$$

Proof of Theorem 23

The proof follows the same principles as those for high playing rates. In this case, there is no need for the restart approximation. The simpler calculation allows us to get a tighter bound

We have again a hybrid Markov chain. Denoting as before $s(y,k)$ the expected time before all players are satisfied, the one steps equations are :

$$\begin{aligned}
s(y,n) &= 0 \\
s(y,i) &= 1 + \frac{i}{n} s(y,i) + \frac{n-i}{n} y^a s(y,i+1) + \frac{n-i}{n} \int_y^1 a u^{a-1} s(u,1) \mathrm{d}u, \quad n > i > 1 \\
s(y,1) &= 1 + \frac{n-1}{n} y^a (1-p) s(y,2) + \frac{n-1}{n} \int_y^1 a u^{a-1} s(u,1) \mathrm{d}u + \frac{1}{n} s(y,1)
\end{aligned}$$

Unlike for high playing rates, the last equation is simply a specific case of the previous one. The justifications for the other transitions stay the same.

Defining $z := y^a$, $s^{(1)}(z,i) := s(z^{\frac{1}{a}}, i)$, one gets

$$\begin{aligned} s^{(1)}(z,n) &= 0; \\ s^{(1)}(z,i) &= \frac{n}{n-i} + \int_z^1 s^{(1)}(u,1)\,\mathrm{d}u + z s^{(1)}(z,i+1); \\ s^{(1)}(z,1) &= \frac{n}{n-1}\,\mathrm{d}u + \int_z^1 s^{(1)}(u,1)\mathrm{d}u + z\, s^{(1)}(z,2). \end{aligned}$$

Rewriting the last equation as

$$s^{(1)}(z,1) = \frac{n}{n-1} + \int_z^1 s^{(1)}(u,1)\mathrm{d}u + z\, s^{(1)}(z,2)$$

and telescoping all other equations into it, we obtain the following .

$$s^{(1)}(z,1) = \sum_{j=0}^{n-2} \frac{n}{n-j-1} z^j + \sum_{j=0}^{n-2} z^j \int_z^1 s^{(1)}(u,1)\,\mathrm{d}u.$$

We define $S(z) := \int_z^1 s^{(1)}(u,1)\mathrm{d}u$. $S(0)$ is the quantity we want to study. Since $S'(z) = -s^{(1)}(z,1)$, we obtain as previously an ordinary differential equation for $S(z)$:

$$S'(z) + G(z)S(z) + V(z) = 0$$

with

$$G(z) := \sum_{i=0}^{n-2} z^i = \frac{1-z^{n-1}}{1-z} \tag{5.16}$$

$$V(z) := \sum_{j=0}^{n-2} \frac{n z^j}{n-j-1} \tag{5.17}$$

We solve this equation on the interval $[0,1]$ using the boundary condition $S(1) = 0$.

We obtain:

$$S(z) = -e^{-q(z)} \int_1^z V(u) e^{q(u)} \mathrm{d}u.$$

where

$$q(z) := \int_0^z G(u)\,\mathrm{d}u = \sum_{j=1}^{n-1} \frac{z^j}{j}.$$

Since $q(0) = 0$, this gives

$$S(0) = \int_0^1 V(u) e^{q(u)} \mathrm{d}u = \int_0^1 \left(\sum_{j=0}^{n-2} \frac{n}{n-j-1} \right) e^{\sum_{i=1}^{n-2} \frac{u^i}{i}} du. \tag{5.18}$$

We can major the sum in the exponent by replacing the power of u by one: $\forall u \in [0,1] \sum_{i=1}^{n-2} \frac{u^i}{i} \leq \sum_{i=1}^{n-2} \frac{1}{i} = \log(n) + \gamma + o(1)$.

This allow to inverse the finite sum and the integral and integrate each separate part easily:

$$\int_0^1 \frac{n}{n-j-1} u^j n e^\gamma du = \frac{n}{(j+1)(n-j-1)} n e^\gamma$$

We can also notice that

$$\sum_{j=0}^{n-2} \frac{n}{(j+1)(n-j-1)}$$

simplifies itself into

$$\sum_{j=0}^{n-2} \frac{n-j-1}{(j+1)(n-j-1)} + \sum_{j=1}^{n-2} \frac{j+1}{(j+1)(n-j-1)} = 2 \sum_{j=1}^{n-1} \frac{1}{j} = 2 \log n + 2\gamma + o(1).$$

thus

$$S(0) \leq 2e^\gamma n \log n + e^{2\gamma} n + o(\log n)$$

Finally, we can extend the calculus of the number of moves from the high playing rate case. With $p = 0$, the bounds on the expectations from formula **??** of M and M^2 becomes $C_1 n$ and $C_2 n^2$ respectively.

The number of intersections is bounded by expression 4.2:

$$\mathbb{E}I \leq \frac{3}{a(n-3)} \mathbb{E}(M^2) + \left(4 + (4c_1 + 1) \frac{n}{n - 4c_2}\right) \mathbb{E}(M). \tag{5.19}$$

Thus the number of intersections is $O(n)$.

Thus

$$T_{BRA} \leq 2e^\gamma n \log n + O(n)$$

Remarks

We know from experiments that the actual formula is of the form $n \log n + \gamma n + Cn + o(n)$ with the first two terms corresponding to the coupon collector part and the Cn being the dominant term reaching time, but the bounding approach relies on the fact that the algorithm end on a stopping time and is forced to include the confirmation phase which then hides the shorter reaching phase.

In the approach based on the round robin computation, this is the most obvious. The confirmation time is exactly one bloc long, by definition and goes from n steps to h_n steps, as none of the skipped steps have any effect. On the other hand, in the reaching phase, almost always, less than half the player are satisfied, meaning the probability for a skipped steps to have had an effect (the same effect a non skipped one would have had) is more than a half. This means that if a tight approach was possible, it would find this phase linear.

5.8 Termination Test

We will now introduce a distributed algorithm using a convergence test that does not rely on the global variable L as before.

In this algorithm, each player can still access the payoff of its actions but cannot know who is currently satisfied nor the global state of the system. To make termination possible, we let each player, when activated, also initiate a game-wide communication to make a termination test. Let us consider the communication procedures named *Termination Test Sender* and *Termination Test Receiver* in Algorithm 10: At every tick of its clock, each player has a probability q of broadcasting a message to every other player. Upon reception of such a message, the receivers interrupt their clock and send an acknowledgment (ack). Once the initial sender gets all the acks, it sends a second message. Upon reception of this second message, each player tests if it needs to change its best response (a player is *stable* if no change is needed) and sends back its stable/unstable status before restarting its 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 most players during two broadcasts and the clock of one (random) player for four broadcasts. We denote the average interruption time by σ_n.

Theorem 31. *When using the best value for q, the execution time of Algorithm 10, is*

$$\mathbb{E}[T^{(4)}] \leqslant \mathbb{E}[T_{BRD}] + 2\sqrt{\mathbb{E}[T_{BRD}]\sigma_n} + \sigma_n.$$

Proof. To study the execution time of Algorithm 10, one can separate the complexity of reaching the equilibrium with the complexity of the termination test. The second part is therefore independent of the overlaps and can be inspired from the approach used in the simpler case were overlaps are neglected, as in [6]. This execution time, denoted $T^{(4)}$, satisfies $T^{(4)} = W_1 + \cdots + W_k + k\sigma_n$, where W_i's are the times elapsed between two consecutive termination tests, and the random number k is the number of termination tests sent before a Nash equilibrium is reached. By construction of the algorithm, the random variables W_i's are independent and identically distributed, according to an exponential law of parameter $p\frac{\lambda}{n}$. Since W_i is independent of the event $\{k > i\}$, the expectation of $T^{(4)}$ can be computed with Wald's lemma:

$$\mathbb{E}[T^{(4)}] = \mathbb{E}[k]\mathbb{E}[W_1] + \mathbb{E}[k]\sigma_n. \tag{5.20}$$

On the other hand, this time is also the end of the first test after the reaching time of a Nash equilibrium, denoted $R^{(4)}$. The waiting time W from $R^{(4)}$ has the same exponential distribution as W_i's. Hence we also have $T^{(4)} = R^{(4)} + W + \sigma_n$.

Algorithm 10: Distributed BRD with termination test

Function Main algorithm (for player k)
  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 local clock*
      if $x_k \notin BR_k(\mathbf{x})$ **then**
        Update strategy to $x_k \in \mathrm{BR}_k(\mathbf{x})$;
      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 Local 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 local clock;

Function Termination Test, Receiver (for player k)
  Stop Clock;
  Send(Ack) to p;
  wait until Test received;
  If $\mathbf{BR}(\mathbf{x}) = x_k$ **Send**(Stable) to sender;
  else **Send**(Unstable) to sender;
  Restart Clock;

Since the distributed algorithm uses Poisson clocks, the order of play is exactly as in Algorithm 9. Therefore, $\mathbb{E}[R^{(4)}] \leqslant (k-1)\sigma_n + \mathbb{E}[T_{BRD}]$. This yields

$$\mathbb{E}[T^{(4)}] = \mathbb{E}[T_{BRD}] + \mathbb{E}[W] + \mathbb{E}[k]\sigma_n. \tag{5.21}$$

By subtracting (5.21) from (5.20), and using $\mathbb{E}[W] = \mathbb{E}[W_1] = \frac{1}{q\lambda}$, one gets $\mathbb{E}[k] = q\lambda\mathbb{E}[T_{BRD}] + 1$.
Equation (5.21) becomes

$$\mathbb{E}[T^{(4)}] \leqslant \mathbb{E}[T_{BRD}] + \frac{1}{q\lambda} + q\lambda\mathbb{E}[T_{BRD}]\sigma_n + \sigma_n.$$

The best value for q that minimizes the formula above is $q^* = \frac{1}{\lambda\sqrt{\mathbb{E}[T_{BRD}]\sigma_n}}$. Finally,

$$\mathbb{E}[T^{(4)}] \leqslant \mathbb{E}[T_{BRD}] + 2\sqrt{\mathbb{E}[T_{BRD}]\sigma_n} + \sigma_n.$$

□

In particular, if one uses a classical model for global synchronization on a distributed algorithm (as in [5]), the duration of our two-steps broadcast is of the form $\sigma_n = O(\log(n))$. In this case, the execution time remains asymptotically bounded by the same value as for distributed BRD, namely

$$\mathbb{E}[T^{(4)}] \leqslant \delta C n \log(n)(1 + o(1)).$$

Chapter 6

Black Box Model

6.1 Model and Motivation

Until now, we were working under the hypothesis that players knew their utilities, or that they could obtain the utilities of their different options easily and without affecting the game. That is the case when said utility is computable from data the player has access to, which is not always the case.

If the details of the instance of the game are unknown or changing, if the utilities are measured instead of computed then the previous assumption fail.

We can introduce a new mode, in which a player always know her current utility and can obtain when active the value of one of her alternate choices.
In a routing game, we could assume that each player knows her current delay and can send a tracer to sample the delay along an alternative path. We can assume that the tracer are light enough not to affect the delay of the other paths.

In effect, that means that reading all alternative utilities no longer necessary.

We can notice the similarities between this new model and a game with an players having each 2 choices. The sections in this small chapter will be about applying the approach through IFA and a Markov chain to make this isomorphism in this case.

6.2 Round Robin

Lets first define what applying a round robin to this model means. One could ask if it is better to have the first player test all his choices before the second start, of if each player try their first choices, then each try their second, until they try their last and start over, or a combination of the two.

The answer is that it does not matter, as long as the only possibility of a choice being tested a second time since the last move is when we reached a Nash equilibrium.

Theorem 32. *Average complexity, round robin black box the average number of steps needed to reach a Nash equilibrium is* $e^{\gamma}na$.

Proof. Similarly to the case treated before, we build a coupling of the random variables used to choose potential between this game and a game where the actual values are forgotten.
The ending condition on the IFA model is that an consecutive potentials are lower than the current chosen.This is a sufficient condition for this algorithm to end as well.

The differences are the intersections, now only possible if the same unique state is tested twice from different states. It is easy to see that for such an event, there need to be at least a full cycle between them : at least one coordinate of the intersection is changed then accessed again between the two checks.

Assuming j players changed their choices, the probability of intersection by the new step with a given one is bounded by $\frac{x^{(n-j)a}}{a^j}$ (with x being the potential of the second state of the pair), which is negligible unless almost all players were satisfied and the potential is high, in which case, the probability that we are on a Nash equilibrium, preventing intersection, is almost one.

□

As a remark, we can notice that the number of utilities read (also the number of comparisons) is the same as the round robin case.

This is actually more general. Assume we change the condition of the steps, for example all player look at all choices (where only the player change, so na choices) and choose the one increasing utility the most, or a subset of players read a subset of their possible choices then move to the state that show the best increase. Then there exists an algorithm which will need this number of comparison, and if possible, it will verify the property that a player check the same choice twice from the same point only if no change is possible.

6.3 Distributed Case

We now consider this model on the distributed case.

6.3.1 Without Overlap

At first, lets assume there are no overlap. We can obtain the following theorem as a consequence of theorem 23:

Theorem 33. *The expectation of the number of comparison needed before convergence is bounded by* $2e^{\gamma}na\log n + O(na)$

Proof. Lets assume for now that the change, test and switch is fast enough that only one player is active at a time. As before, this is equivalent to a centralized process where the next player is chosen uniformly. We can use the same coupling of potential as previously, while the ending condition remains that na consecutive new values are lower than the current maximum (in any state there are n direction with each a potential that could possibly be better than the current one).

The probability of intersection is bounded by its value for the classical case:
Instead of needing two state at a hamming distance of two playing along the directions of the differences, an intersection now also need the new coordinates to match the precises values. If we bound by grouping all the sampling one each direction together and

assuming the full line is read, this lead us back to the exact classical case.
This means we can use the IFA approximation as before.

The model we obtain is identical to the one for a game of na players with two choices and get the same solution.

□

Unlike above, the number of comparison needed seems to increase. That is due to the bound on the reaching time and the time cost of the second phase, which cause a coupon collector problem over many more players.
If we use a hybrid system, where each player is chosen randomly, but cycles its choices using a round robin approach, the numerical experiments show a similar number of comparisons needed.

6.3.2 With Overlaps

Theorem 34. *Black box model with overlap The expected execution time of the distributed best response dynamics with Poisson clocks of total rate λ is bounded by*

$$\frac{anH_{an}}{\lambda} \leqslant \mathbb{E}[T_{BRD}] \leqslant \frac{1}{\lambda}\left((an)^{1-p}H_{an-1} + \frac{(an)^{1-p} + C_1 + \frac{C_2}{an} + C_2\frac{G(p,an)}{an(1-p)}}{1-p} + 1\right) G(p,an). \tag{6.1}$$

where $G(p,n) = \left(\frac{e^{pH_{n-1}}}{p} - \frac{1-p}{p}\right)$, H_n is the nth harmonic number, $p = 1 - e^{-\delta\lambda}$ is the overlap probability, and C_1 and C_2 are constants.

The proof is similar to the one presented in Section 5.6 applied to a game of na players with 2 choices each, with the following difference:

When two virtual players corresponding to two possible strategies of the same player overlap, it does not cause an overlap in the black box model, unlike the virtual game.

This means the probability of overlap is lower than considered in the proof. This does not invalidate the bound.

A reason to chose to allow players to test new choices without changing the state is that if we did not, an overlap would influence both players instead of just the second, complicating the model greatly.

6.3.3 Stopping Criterion

The algorithm stops when all players know that they are all satisfied, or when it knows from another source that all players are satisfied. This approach use the fact that a player could test trivially if they were on their best response, without affecting the game or slowing the algorithm.

This is not true in the black box model, as player do not know their utilities. In addition, if player can have several utilities with the same values, or if the sampling have noises, a player cannot detect when others act.

A solution to that issue could be to have a periodic centralized stop, individuals tests in parallels, and then conclude or resume the algorithms. This would need the assumption that the network does not lose message, that the players have a way to communicate that guaranteed all are accounted and a few other hypothesis that run counter to the idea of the model.

If we instead assume that all utilities of a player are distinct and different enough that a change of state cannot be hidden by the sampling noise, then they can individually keep a list of strategy tested since the last move, and use that to now whether they are satisfied or not.

Finally, in some cases, that issue will not arise. If the goal is to maximize the time spent on Nash equilibria in a changing game, or if the modeled problem includes a native way to test the current state.

Chapter 7

Network Games

7.1 Indifferent Players

In this section, we consider that players may not all interact with each other, and we want to take advantage of this to design new algorithms to compute NE.

Let us define $\Delta_k(x)$, the profile obtained after player k has played her best response under profile $\mathbf{x}$:

$$\Delta_k(\mathbf{x}) \stackrel{\text{def}}{=} (x_0, \dots, \mathrm{BR}_k(\mathbf{x}), \dots, x_{n-1}) = (\mathrm{BR}_k(\mathbf{x}), \mathbf{x}_{-k}).$$

Using this notation, we give a definition of indifferent players more general than what is usually adopted in the literature (see for example [12]).

Definition 35 (Indifferent Players, Interaction Graph)**.** Player i is *indifferent* to player j if, for any state $\mathbf{x}$,

$$\Delta_i(\Delta_j(\mathbf{x})) = \Delta_j(\Delta_i(\mathbf{x})). \tag{7.1}$$

Otherwise, we say that i and j are *neighbors*. The *interaction graph* G is the undirected graph linking neighbors.

In particular, Condition (7.1) is satisfied when the payoff function for player j, $u_j(\mathbf{x})$, does not depend on x_i. This stronger criterion is used in [12] to define independence of two players. Actually all the results stated in this section will remain valid if the interaction graph G is replaced by any graph that contains G as a sub-graph. In particular if indifference is replaced by the stronger notion used in [12], the resulting graph will contain G as a subgraph. When several players (say i and j) play simultaneously, the corresponding *simultaneous* best response operator is

$$\mathrm{BR}_{\{i,j\}}(\mathbf{x}) \stackrel{\text{def}}{=} \underset{(\alpha_i,\beta_j)\in\mathcal{A}^2}{\arg\max}\ F(\alpha_i, \beta_j; \mathbf{x}_{-i-j}).$$

The corresponding state is

$$\Delta_{\{i,j\}}(\mathbf{x}) \stackrel{\text{def}}{=} (x_0, \dots, \alpha_i, \dots, \beta_j, \dots, x_{n-1}),$$

where α_i and β_j are the argmax in the previous equation.

Lemma 36. *If two players i and j are indifferent, then $\Delta_i(\Delta_j(\mathbf{x})) = \Delta_j(\Delta_i(\mathbf{x})) = \Delta_{\{i,j\}}(\mathbf{x})$.*

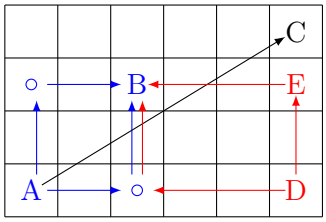


Figure 7.1: Illustration of the proof of Lemma 36

Proof. Let us consider the potential matrix restricted to two independent players as illustrated by Figure 7.1. Player 1 acts on the first coordinate (lines), and player 2 chooses the second coordinate (columns). By definition of indifference, starting from any state A and letting Player 1 play before Player 2 or Player 2 play before Player 1 leads to the same state B in the figure.

Let us suppose that state B (best-response state after 1 and 2 have played) does not have the global optimum potential over the whole matrix, the state with optimal potential being state C. Now, if C and B have distinct second coordinates, let us start the game from another state, D with first coordinate and second coordinate in common with A and C respectively. If we let Player 2 act first, followed by Player 1, we should end again in B. By indifference, the same state B is reached when Player 1 plays before the Player 2. The intermediate state (E) is thus the best response of Player 1 in D. It has a larger potential than any state on the same column, including C. This implies that C cannot be the global optimum. □

7.2 Concurrent Case

One can exploit the commutativity of indifferent players by making them play simultaneously. For that, one can pre-compute a covering of the interaction graph G by independent sets[1] and in the algorithm BRA one can activate players in groups (one group being a set of the covering). This version of BRA will converge at least as fast as a game without indifference but with only as many players as there are sets in the covering. One can use a coloring of the interaction graph with a minimal number of colors to get the smallest number of groups of players.

Theorem 37. *Under minimal coloring of the interaction graph G, the reaching and execution times of Algorithm 11 satisfy*

$$\begin{aligned} \mathbb{E}[R^{(3)}_{\mathrm{RR}}] &= \delta_{BR}(e^{\gamma}-1)\chi(G) + o(\chi(G)) \\ \mathbb{E}[T^{(3)}_{\mathrm{RR}}] &= \delta_{BR}e^{\gamma}\chi(G) + o(\chi(G)) \end{aligned}$$

where $\chi(G)$ is the chromatic number of the interaction graph G and δ_{BR} is, as before, the complexity of one call to the function BR.

[1] in a graph, an independent set is a set of nodes not inter-connected by any edge

Algorithm 11: Parallel Best Response Algorithm with simultaneous plays

Input: Game utilities $(u_k(\cdot))$; Initial state $(\mathbf{x} := \mathbf{x}(0))$;
Construct a coloring of G and fix a round robin sequence of colors;
repeat
  Pick next color c
  foreach *player in c in parallel* **do**
    choose the action that maximizes payoff
until *convergence*;

Proof. A direct consequence of the definition of indifference and of Lemma 36 is that when several indifferent players act simultaneously, they reach the state of maximal potential among them, as if they had played one after the other in any order.

This shows that if the interaction graph is colored and if BR is used by all players with the same color simultaneously, the behavior is the same as in a new game where players are the colored sets (called super-players in the following). The call of best response for these super-players is merely the parallel call to best response for each player of the set, and hence it still costs only δ_{BR}. This allows us to use Equations (**??**) and (**??**) on the new game to assess its complexity. □

If the minimal coloring is replaced by a smallest cover with maximal indifferent sets, the performance is further improved (though only slightly).

The interaction graph of our example (Figure 7.3) is 2-colorable. In this case, Algorithm 11 alternates between the central player alone, and all other players together. The average convergence time is constant, equal to $2e^{\gamma}\delta_{\mathrm{BR}}$.

7.3 Distributed Algorithms for Network Games

We write $d(i)$ the number of players not indifferent to player i and d_{max} the maximum value of d. It can be read as the degree of a player in the interaction graph, and the degree maximum of the graph.

An overlap can only happen between neighbors, hence the probability of overlap p is now dependent on the player involved and is $p_i = 1 - e^{-\frac{\lambda d_i}{n}\delta}$ bounded by $p_{max} = 1 - e^{-\frac{\lambda d_{max}}{n}\delta}$.

If the degree of the graph is small compared to the number of players then p_{max} is much smaller than the previous value of p, the two become equal if we replace λ by $\frac{n}{d_{max}}\lambda$. the theorem 24 and the corollary 25 become

Theorem 38. *The expected execution time of the distributed best response dynamics with*

Poisson clocks of total rate $\frac{n}{d_{max}}\lambda$ *in a network of degree bounded by* d_{max} *satisfies*

$$\frac{d_{max}H_n}{\lambda} \leqslant \mathbb{E}[T_{BRD}] \leqslant \frac{d_{max}}{n\lambda}\left(n^{1-p}H_{n-1} + \frac{n^{1-p} + C_1 + \frac{C_2}{n} + C_2\frac{G(p,n)}{n(1-p)}}{1-p} + 1\right)G(p,n). \tag{7.2}$$

where $G(p,n) = \left(\frac{e^{pH_{n-1}}}{p} - \frac{1-p}{p}\right)$, H_n *is the nth harmonic number,* $p = 1 - e^{-\delta\lambda}$ *is now a bound on the overlap probabilities of the players, and* C_1 *and* C_2 *are constants.*

and

Corollary 39. *Choosing the rate* $\hat{\lambda} = \frac{n}{d_{max}}\frac{\log(\log(n)) - \log\log\log(n))}{\delta}$, $\hat{p} \leq 1 - \frac{\log(\log(n))}{\log(n)}$, *the average execution time satisfies*

$$\frac{\delta d_{max}\log(n)}{\log(\log(n))}(1+o(1)) \leqslant \mathbb{E}[T_{BRD}] \leqslant e^{\gamma}\frac{\delta d_{max}\log(n)}{\log(\log(n))}(1+o(1)) \tag{7.3}$$

where γ *is the Euler constant.*

We can make a few remarks :

- This bring a guaranteed acceleration by a factor $\frac{n}{d_{max}}$. If we reuse the example from the previous section, in which the graph is plan, we now have a time complexity of $d_{max}e^{\gamma}\frac{\log(n)}{\log\log n} \simeq 7\frac{\log n}{\log\log n} = O(\log n)$
- Intuitively, it would make sens to use the degree of each player to set its rate. We know that the steps before reaching a Nash equilibrium have mostly the same weight and that after reaching the equilibrium, it is a coupon collector problem. Modeling this completely would be difficult.
- A possible intuition would be to use the average degree instead : it would fail. Imagine a graph with a clique of size $\sqrt{n}$ and $n - \sqrt{n}$ players connected to the member of the clique by one edge each. The probability of overlap using the average degree (2) in the clique would go to one too fast and slow convergence.

7.4 Structured Graphs

Until now, we only consider homogeneous playing rates for all players. When the interaction graph G is highly asymmetrical (as in our running example), one might think that it could be better to adapt the playing rate to the structure of the game graph (e.g., to the number of neighbors). In addition, given that the hypothesis of little knowledge on the game is false, one could think that the whole analysis lose in accuracy.

To test the validity of this intuition, we have run the distributed algorithm on a few simple but asymmetrical networks while varying the relative values of the playing rate of the classes of player.

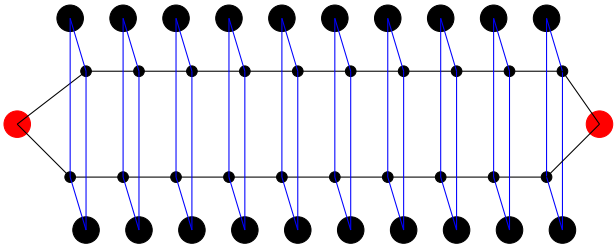


Figure 7.2: A routing game made of one central (horizontal) flow with two possible routes (up or down) against $n-1$ transversal (vertical) flows, each with two routes (left or right). The routes of the central flow share two hops with each transversal flow, but the routes of transversal flows do not intersect each other.

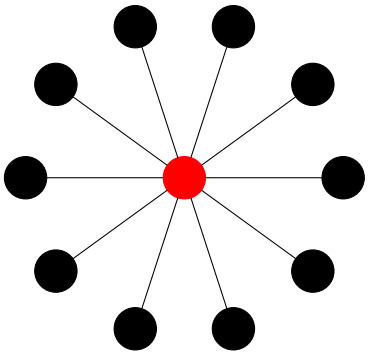


Figure 7.3: Interaction graph of the routing game displayed in Figure 7.2, with a minimal coloring (red for the central player and black for all other players).

Here is an example. We consider the following multi-commodity flow problem. A set of *flows* of packets must be routed over the network. Each flow (considered as a player) is characterized by a source-node, a destination-node and a nominal arrival rate of packets. Also, each flow is assigned a set of paths in the network from its source to its destination. *Configuration* are choices of one path per flow. For each flow (player), the payoff (or cost here) is the delay on its chosen route.

This corresponding game is an *atomic non-splittable routing game*. It is a potential game if the delay on each link only depends on the number of players using it. In such a game, two players whose paths do not intersect are independent, according to our definition. Furthermore, two players are also independent if any path with common nodes is less efficient in all configurations than another path for both players.

The interaction graph of this game feature two classes of players : external and central.

In Figure 7.4 we report the empirical mean execution/reaching times (95% confidence

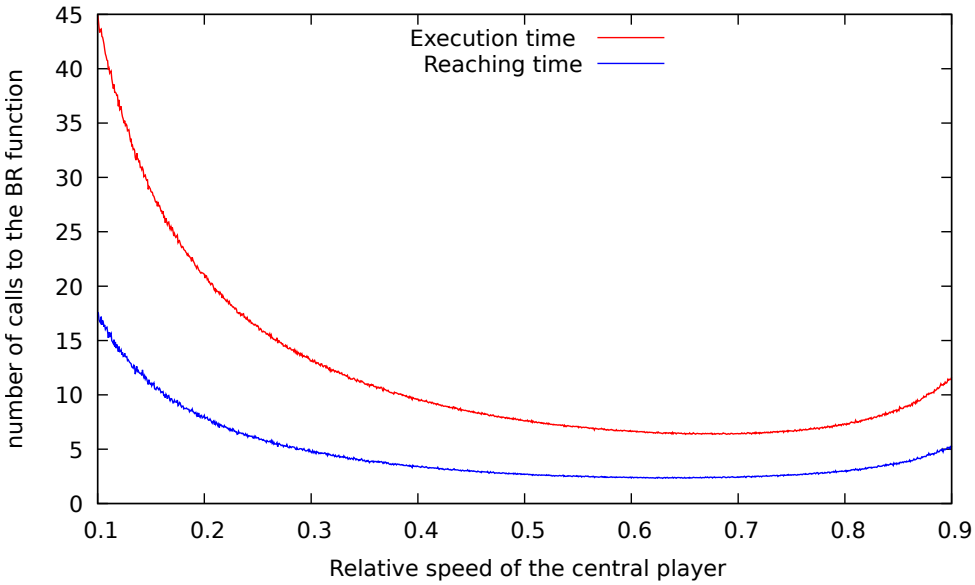


Figure 7.4: Mean execution time and reaching time of Algorithm **??** over 10,000 random samples of the routing game displayed in Figure 7.2, when the relative speed of the central player ($\frac{\lambda_0}{\lambda_0+\lambda_1}$) ranges from 0.1 to 0.9.

intervals are too small to be seen). One can see that the performance of the algorithm is rather insensitive to the relative speed of the players when the extreme cases are avoided. One can notice that the homogeneous case ($\lambda_0 = \lambda_1 = 1$), corresponding to a relative speed ($\frac{\lambda_0}{\lambda_0+\lambda_1}$) of the first player equal to 0.5, is almost optimal in spite of the heterogeneity of the interaction graph.

The games tested to get this figure were randomly generated through the utility of the players, chosen uniformly in ranges expectable for the source problem.

The relative flatness of the figure, which was also the case for the other tested networks tells us that the time complexity is not much dependent on the precise rate making an optimization of it player by player not necessary.

In addition, the variation of both average times as well as variances of reaching time with N still corresponds mostly to the model.

Chapter 8

Perspectives

The best response dynamics was first introduced as a way to prove the existence of an equilibrium in potential games. Despite this, the analysis of the average time it takes to find a Nash equilibrium show that it is:

- Fast, with a linear time in its original condition of use.
- Adaptable, it can be used in a coordinated system, in a coherent parallel system or in a completely distributed system. It can tolerate the player being grouped together or having to sample each strategy separately. It can even be altered to find a global extremum instead of local.
- Robust, the condition for it to always gives a correct state are quite permissive and when they are not respected, the only risk is to not get any answer. In fact, in most cases, having players interfere with each other only slow the dynamics a bit.

This shows that with the correct criterion and approach an algorithm considered slow can be shown practical, without having to modify it.
This is not a new concept: the simplex is also well known for being exponential in worst case but much faster in most criteria than its polynomial counterpart in practice.

In order to be more precise in the analysis, one could change the choice the activation rate of the player. Using a distribution of event other than Poisson. As an example one could use fixed delay with noise : each player activate at a different starting time, then wait a given time with little variations before activating again. It looks like it would effectively gives a round robin sequence, but this model is not that simple as the overlap, which can now be quasi cyclic, could lock the game in a cycle until the noise change the activation time enough to switch or remove the overlap.

Another way to extend this work would be to take into account games with known properties on the distribution which would translate to a bias on the values of potential or a restriction on the set of possible games, affecting the computation of the average.

In the domain of game theory, there are other algorithms on which a similar approach could apply. For example, the Howard policy iteration algorithm is analysed in the worst case in [20]. It is linear in worst case but near constant time in simulations.

Another possible approach would be to do a similar Markovian analysis using smooth complexity instead of the simple average.

Chapter 9

Perfect Sampling of Markov Chains

The following contains the summary of the work done at the beginning of my thesis on models of random walk in specific Markov chains. To preserve the thesis coherence, it will be just an overview as this part is less important and not related to the main work.

9.1 Perfect Sampling

9.1.1 Context

The object of the study are markov chain modeling open Jackson Networks (as defined in [9]). The states of those chain are defined by the filling level of several queue. As a consequence, their state space is exponential in the number of queues (product of the different capacity) and follow a grid structure, only containing edge between neighbor or state at a bounded distance.

As those chain are too big to analyze explicitly to compute the equilibrium distribution, the approach is instead to sample this distribution, to choose states according to the equilibrium distribution of the chain, in order to study or test properties.

A classical sampling method would be to choose a point randomly then follow the chain transitions for a time. This only gives an approximation of the stable distribution, unless one let the walk last an infinite time.

Perfect sampling gives a method to obtain in finite time a state chosen according to the exact equilibrium distribution.

9.1.2 Perfect Sampling on Open Jackson Networks

Perfect sampling is the idea to replace the random starting point by the set of all point and only use the sample after all trajectories have converged into one. Doing so normally introduce a bias in favor of the point allowing such fusion, so the progression need to be backward.
start at a time $-t$ and follow all trajectories until the time 0.
If all trajectories have converged on the same point, this is the sample.
Else, start again at time $-2t$ using the same random values when in the interval $[-t, 0]$ and keep doubling until the trajectories have fused.

Since the stopping time is fixed and independent from the trajectories and the random variables used, the final sample is chosen without bias. Since any trajectory end in the

same point, so would one starting with a state chosen with the equilibrium distribution. The sample is chosen according the correct distribution.

In 1, we consider open Jackson networks with losses with mixed finite and infinite queues and analyze the efficiency of sampling from their exact stationary distribution. We show that perfect sampling is possible, although the underlying Markov chain may have an infinite state space. The main idea is to use a Jackson network with infinite buffers (that has a product form stationary distribution) to bound the number of initial conditions to be considered in the coupling from the past scheme. this report also provide bounds on the sampling time of this new perfect sampling algorithm for acyclic or hyper-stable networks. These bounds show that the new algorithm is considerably more efficient than existing perfect samplers even in the case where all queues are finite.

In addition to the content of this report, there were some minor work around the implementation that was made in the software psi3. Notable among them were the work on skipping : the criterion to detect when a step does not have any effect and will not have in the future as well as implementing an algorithm using this property. The no regret proof to learn predict the size of the needed trajectory, using the median from a training period.

This work is presented in the paper 1.

9.2 Random Walk with Forbidden Edges

We suppose that we have a grid lacking a bounded number of edges and we want to sample the altered equilibrium distribution.

We construct an algorithm to sample the stationary distribution of a random walk over $(1, \cdots, N)^d$ with forbidden arcs, using both the rejection method the coupling from the past from perfect sampling. It also contain a complexity analysis of the approach in a few case.

The conference paper in which these results were published is the 6 item in the publication list below.

Publications

This thesis has given rise to the following publications.

Journal Papers

1. **Perfect sampling of Jackson queuing networks**
   With Ana Busic, Bruno Gaujal and Florence Perronnin.
   In Queuing Systems (Springer) 2015, 80 (3),
   This paper details the results mentioned in Section 9.1

2. **Distributed of Best Response Dynamics with High Playing Rates in Potential Games**
   with Federica Garin and Bruno Gaujal.
   Performance Evaluation (to appear).
   The content of this article corresponds to Chapter 5

Conference Papers

3. **Distributed Best Response Algorithms for Potential Games**.
   With Federica Garin and Bruno Gaujal
   In the 16th European Control Conference (ECC 2018), Jun 2018, Limassol, Cyprus. 2018.
   This paper contains a short version of Chapter 4 and introduces the concept of indifferent players along with the results that are shown in Chapter **??**.

4. **Efficiency of Best Response Dynamics with High Playing Rates in Potential Games**
   With Federica Garin and Bruno Gaujal.
   In the 36th Int. Symposium on Computer Performance, Modeling, Measurements and Evaluation (IFIP Performance 2018) at Toulouse, Dec. 2018,
   (extended Abstract to appear in Performance Evaluation Review).
   The content of this paper is included to Chapter 5.

5. **Complexity and Optimality of the Best Response Algorithm in Random Potential Games**
   With Bruno Gaujal
   In Symposium on Algorithmic Game Theory (SAGT) on (september) 2016, at Liverpool, United Kingdom.

pp.40-51
This paper is the conference version of Chapter 4.

6. **A perfect sampling algorithm of random walks with forbidden arcs** with Bruno Gaujal, Florence Perronnin and Jean-Marc Vincent.
published in QEST 2014 – 11th International Conference on Quantitative Evaluation of Systems, on Sep 2014 in Florence, Italy, Springer, 8657, pp.178-193, 2014, LNCS.
This paper details the results mentioned in Section 9.2 .

7. **General Revision Protocols in Best Response Algorithms for Potential Games**
With Pierre Coucheney, Bruno Gaujal and Corinne Touati.
IEEE Explore.
In Network Games, Control and Optimization (NetGCoop) on Oct 2014, Trento, Italy.
This paper is the conference version of Chapter 3.

Other

Workshop **Average complexity of the Best Response Algorithm in Potential Games**
With Bruno Gaujal
In Atelier Evalution de Performance 2016, on Mar 2016 in Toulouse, France.
The content of the paper is a short version of Chapter 4.

National Conference **Average complexity of the Best Response Algorithm in Potential Games**
with Bruno Gaujal.
in the 17ème conférence dela Société française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2016), on Feb 2016, at Compiègne, France.
The work on this conference correspond to the content of Chapter 4 with a focus on the justification on the approach and an explanation of the context to a public less used to Game theory.

Bibliography

[1] David Aldous and James Allen Fill. Reversible markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014, available at `http://www.stat.berkeley.edu/$\sim$aldous/RWG/book.html`.

[2] C. Alos-Ferrer and N. Netzer. The logit-response dynamics. *Games and Economic Behavior*, 68(2):413–427, 2010.

[3] M. Beckman, C. B. McGuire, and C. B.Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.

[4] P.W. Goldberg C. Daskalakis and C.H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(3):195–259, 2009.

[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. MIT Press, 2009.

[6] Stéphane Durand, Federica Garin, and Bruno Gaujal. Distributed best response algorithms for potential games. In *European Control Conference*, Limassol, Cyprus, 2018.

[7] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 604–612. ACM, 2004.

[8] R. G. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, 25(1):73–85, 1977.

[9] J. R. Jackson. Job shop-like queueing systems. *Management Sci.*, 10,131, 1963.

[10] Frank Thomson Leighton and Ronald L Rivest. *The Markov Chain Tree Theorem*. Massachusetts Institute of Technology, 1983.

[11] Jun S. Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):pp. 958–966, 1994.

[12] Jason R. Marden and Jeff S. Shamma. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. *Games and Economic Behavior*, 75(2):788 – 808, 2012.

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

[14] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124 – 143, 1996.

[15] John Nash. Equilibrium points in n-person games. *Proc. of the Nat. Acad. of Sciences*, 38:48–49, 1950.

[16] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multi-user communication networks. *IEEE/ACM Trans. on Networking*, 1(5):510–521, 1993.

[17] Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *Int. J. of Game Theory, Springer*, 2(1):65–67, 1973.

[18] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.

[19] William H. Sandholm. *Population Games and Evolutionary Dynamics*. MIT Press, 2010.

[20] Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. *Mathematics of Operations Research*, 41(3):758–774, 2016.

[21] Mark Voorneveld. Best-response potential games. *Economics letters*, 66(3):289–295, 2000.

[22] J.G. Wardrop. Some theoretical aspects of road traffic research. Part ii. *Proc. of the Institute of Civil Engineers*, 1:325–378, 1954.