Parallel Processing Letters
© World Scientific Publishing Company

DISTRIBUTED LEARNING OF EQUILIBRIA IN A ROUTING GAME

Dominique Barth and Johanne Cohen

Laboratoire PRiSM. Université de Versailles, 45, avenue des Etats-Unis, 78000 Versailles, FRANCE - dominique.barth — johanne.cohen@prism.uvsq.fr

Olivier Bournez

Ecole Polytechnique, laboratoire d'informatique. 91128 Palaiseau cedex. France - bournez@lix.polytechnique.fr

Octave Boussaton

LORIA/INRIA-CNRS-UHP. 615 Rue du Jardin Botanique, 54602 Villers-Lès-Nancy, FRANCE - octave.boussaton@loria.fr

Received (received date)
Revised (revised date)
Communicated by (Name of Editor)

ABSTRACT

We focus on the problem of learning equilibria in a particular routing game similar to the Wardrop traffic model. We describe a routing game played by a large number of players and present a distributed learning algorithm that we prove to (weakly) converge to equilibria for the system. The proof of convergence is based on a differential equation governing the global evolution of the system that is inferred from all the local evolutions of the agents in play. We prove that the differential equation converges with the help of Lyapunov techniques.

Keywords: learning algorithm, Nash equilibria, weak convergence, Lyapunov stability, routing game

1. Introduction

We consider here a problem that could very well arise in different kinds of networks, phone, computer, etc... Where the issue of optimizing the flows (of information or physical) in the network can become crucial. In the early fifties, Wardrop introduced, in a study on road transportation networks (see [32]), a model for helping to apprehend road traffic variations, observing patterns and other things. The goal of his study was to understand how to *improve* the road network in general, in terms of traveling time for drivers, queuing time at critical points, road congestion, danger for pedestrians and so forth. An alternative presentation of the model can be found in [29]. This model, that has proved to be solid, has been conceived to represent various road transportation problems with infinitely many users, each one of them being responsible for an infinitesimal amount of traffic. A user was seen as belonging to, or *assigned to*, what is called a *commodity* and for each of several commodities, a certain amount of traffic, or flow demand, has to be routed from a given source to a given destination via a collection of paths. A flow in which, for all commodities, the latencies of all used paths are minimal, with respect to the commodity they are linked to, is called a Wardrop equilibrium of the network. Whereas it is well-known that such equilibria can be solved by centralized algorithms in polynomial time, as in [30], we are interested in *distributed* algorithms to compute Wardrop equilibria.

Actually, our model is slightly different from the original Wardrop model [32] (and similar to the one considered in [30]) in the sense that we consider the flow to be incurred by a **finite** number N of users. Also, while each one of the players remains responsible for a fraction of the entire flow of their respective commodity, we will not be approaching the eventual convergence of the system from a commodity angle. Any player has a set of admissible paths among which he aims at balancing his own strategy, which consists in a choice policy over his set of paths, such that, after some time, he finds a strategy that would assure him the highest utility. In some cases, the jointly computed allocation will be both a Nash equilibrium and a Wardrop equilibrium for the system.

Our motivation is to understand if, how and when equilibria can be learned in games. The dynamic considered here has both the advantage of being decentralized and of requiring partial and very limited informations. We have indeed a discrete stochastic dynamic with N players playing a repeated game, at each time step they all choose between a finite number of strategies (paths) they can use. After each play, players get to know the *result* of the path they chose, or its latency. Players want to learn the optimal strategy which will be the one that keeps their latency as low as possible. After each play, each player updates his strategy based solely on his current action and the latency he experienced. We want to design a distributed algorithm that learns equilibria in games while requiring minimal informations for players to use.

In [26], Thathachar et al. presented a dynamic for learning Nash equilibria in multiperson games. This dynamic is such that, for general games and under certain strong conditions, all stable stationary points are Nash equilibria. Whereas the dynamic is not necessarily convergent for general games (see [26]), we show here that it is convergent for linear Wardrop networks. We call linear Wardrop networks the case where latency functions are affine.

Our approach is based on what can be seen as a macroscopic abstraction of the microscopic evolution rules of the involved agents, in terms of differential equations governing the global state of the system. We prove that this dynamic converges to some stable state for linear Wardrop networks with the help of Lyapunov techniques. We give a short hint of non converging settings in the last part of the paper.

2. Related Work

In the history of game theory, various algorithms for learning equilibrium states have been proposed: centralized and decentralized (or distributed) algorithms, games with perfect, complete or incomplete information, with a restricted number of players, etc... See e.g. [20] for an introduction to the learning automata model, and the general references in [26] for specific studies for zero-sum games, N-person games with common payoff, non-cooperative games, etc...

Wardrop traffic model was introduced in [32] to apprehend road traffic. More recently, it has often been considered as a model of computer network traffic. The price of anarchy, introduced by [19] in order to compare costs of Nash equilibria to costs of optimal (social) states has been intensively studied on these games: see e.g. [29, 28, 4, 13, 5].

There are a few works considering dynamical versions of these games, where agents try to learn equilibria, in the spirit of this paper.

In [10], extending [11] and [12], Fischer and al. consider a game in the original Wardrop settings, i.e. a case where each user carries an infinitesimal amount of traffic. At each round, each agent samples an alternative routing path and compares the latency on its current path with the sampled one. If an agent observes that it can improve its latency, then it switches with some probability that depends on the improvement offered by the better paths, otherwise, it sticks to its current path. Upper bounds on the time of convergence were established for asymmetric and symmetric games. A symmetric game is made of one single commodity, asymmetric games are with more than one commodity.

In [30] Fischer and al. consider a more tractable version of this learning algorithm, considering a model with a finite number of players, similar to ours. The considered algorithm, based on a randomized path decomposition in every communication round, is also very different from ours.

Nash equilibria learning algorithms for other problems have also been considered recently, in particular for load balancing problems.

Note that the proof of existence of a pure Nash equilibria for the load balancing problem of [19] can be turned into a dynamics: players play in turn, and move to machines with a lower load. Such a strategy can be proved to lead to a pure Nash equilibrium. Bounds on the convergence time have been investigated in [7, 8]. Since players play in turns, this is often called the *Elementary Step System.* Other results of convergence in this model have been investigated in [14, 22, 25].

Concerning models that allow concurrent redecisions, we can mention the followings works. In [9], tasks are allowed in parallel to migrate from overloaded to underloaded resources. The process is proved to terminate in expected $O(\log\log n + \log m)$ rounds.

In [3] is considered a distributed process that avoids that latter problem: only local knowledge is required. The process is proved to terminate in expected $O(\log\log n + m^4)$ rounds. The analysis is done only for unitary weights and identical machines. Techniques involved in the proof, relying on martingale techniques, are somehow related to techniques for studying the classical problem of allocating balls into bins games as evenly as possible.

The dynamics considered in our present paper has been studied in [26] for general stochastic games where Thathachar & al. proved that the dynamics is weakly convergent to some function, solution of an ordinary differential equation. This ordinary differential equation turns out to be a replicator equation. While a sufficient condition for convergence is given, no error bounds are provided and no potential function is established.

Replicator equations have been deeply studied in evolutionary game theory [17, 33]. Evolutionary game theory isn't restricted to these dynamics but considers a whole family of dynamics that satisfy a so called folk theorem in the spirit of Theorem 2.

Bounds on the rate of convergence of fictitious play dynamics have been established in [15], and in [18] for the best response dynamics. Fictitious play has been proved to be convergent for zero-sum games using numerical analysis methods or, more generally, stochastic approximation theory: fictitious play can be proved to be a Euler discretization of a certain continuous time process [17].

A replicator equation for allocation games has been considered in [1], where authors establish a potential function for it. Their dynamics is not the same as ours: we have a replicator dynamics where fitnesses are given by true costs, whereas for some reason, marginal costs are considered in [1].

3. Wardrop's Traffic Model

We consider a routing game as given by a directed acyclic graph $G = (V, E)$ with V the set of nodes and E the set of edges (directed connexion between 2 nodes). To each edge $e \in E$ is associated a continuous and non decreasing latency function $\ell_e : \mathbb{R}^+ \to \mathbb{R}^+$. We are given $[C] = \{1, 2, ..., C\}$ a set of commodities, each of which is specified by a triplet consisting, for commodity c, in: a source-destination pair of nodes denoted by (s_c, t_c), a subgraph $G_c = (V_c, E_c)$ of G connecting s_c and t_c with all possible paths between these 2 nodes, and a flow demand $R_c \geq 0$. The total flow demand in the network is $R = \sum_{c \in [C]} R_c$. We may assume without loss of generality that $R = 1$. Let $\mathcal{P}_c$ denote the admissible paths of commodity c, i.e. all paths connecting s_c and t_c. We may assume that the sets $\mathcal{P}_c$, for all $c \in [C]$ are disjoint and define c_P to be the unique commodity to which path P belongs.

A non-negative path flow vector $(f_P)_{P \in \mathcal{P}}$ is *feasible* if it satisfies the flow demands $\sum_{P \in \mathcal{P}_c} f_P = R_c$, for all $c \in [C]$. A path flow vector $(f_P)_{P \in \mathcal{P}}$ induces an edge flow vector $f = (f_{e,c})_{e \in E, c \in [C]}$ with $f_{e,c} = \sum_{P \in \mathcal{P}_c : e \in P} f_P$. The total flow on edge e is $f_e = \sum_{c \in [C]} f_{e,c}$. The latency of an edge e is given by $\ell_e(f_e)$ and the latency of a path P is given by the sum of the latencies of its edges.

$$\ell_P(f) = \sum_{e \in P} \ell_e(f_e). \tag{1}$$

A flow vector in this model is considered stable when no fraction of the flow (which is a player's flow in our case) can improve his latency by moving unilaterally to another path. It is easy to see that this implies that all used paths in the same commodity must have the same (minimal) latency.

Definition 3.1. A feasible flow vector f is at a **Wardrop Equilibrium** if for every commodity $c \in [C]$ and paths $P_1, P_2 \in \mathcal{P}_c$ with $f_{P_1} > 0$, $\ell_{P_1}(f) \leq \ell_{P_2}(f)$ holds.

We now *extend* the original Wardrop model [32] to an N player game (similar settings have been considered in [30]). We assume that we have a finite set $[N]$ of players, each of which is associated to one commodity. Players represent a small fraction of the total flow of their fixed commodity and they want to find the strategy with the lowest latency in the sense of Nash which is formally defined in Definition 3.2.

In this present work, we narrowed down our investigations to the case of linear cost functions. We assume here that for every edge e, there are some constants α_e, and $\beta_e > 0$ such that the associated latency function is of the form $\ell_e(f_e) = \alpha_e f_e + \beta_e$. As we said previously, we study the evolution of the system from a player perspective. In order to ease the notation we denote $com(i)$ the commodity player i plays in.

3.1. *Game Theoretic Settings*

Players are selfish and they play without any centralized control, they also have a very local - restricted - view of the system.

Every player $i \in [N]$ has the finite set of actions, or paths, $\mathcal{P}_{com(i)}$. For every player i, choosing an element of $\mathcal{P}_{com(i)}$ can be considered as playing a *pure strategy.* For player i we note $m_i = |\mathcal{P}_{com(i)}|$ the number of paths available to him.

As already said, we suppose that the game is played repeatedly. At each elementary time step t, players know their latency and the path they choose. Each one of them selects a path at each game according to his mixed strategy that we note $q_i(t)$ for player i. $\forall i \in [N], \forall s \in [1, m_i]$, $q_{i,s}$ denotes the probability for player i to select path s at step t.

More precisely, at the end of each game, players are *rewarded* with random payoffs which are function of all the latencies of all the paths they chose for the game and what it *cost* them (here, the latency they experience). This notion of payoff is often referred to as utility or also fitness in evolutionary game theory. This indicates how satisfied - happy - players are about the game they just played and what the result is for them. Here we note $r_i(t)$ the utility for player i at game t with

$$r_i(t) = 1 - \ell_{P_i(t)}, \quad \forall i, \forall t, \;\; r_i(t) \in [0,1] \tag{2}$$

where $P_i(t)$ is the path player i chose at time t.

Remark 3.1. We assume here that latencies are always greater than (or equal to) 0 and always smaller than (or equal to) 1. We do so without loss of generality in what follows, in order to simplify the reasoning. If latencies don't fit these requirements and under certain conditions on the latency functions, they can be normalized by players while attributing the utility value to the latency they get. A normalized utility for player i would be of the form $r_i = 1 - \frac{\ell_{P_i(t)}}{X}$ with X big enough.

The result of a game for any player i depends on all the choices made by the other players. We define the payoff function $d_i : \prod_{j=1}^{N} \mathcal{P}_{com(j)} \rightarrow [0,1], 1 \leq i \leq N$, by:

$$d_i(a_1, a_2, ..., a_N) = r_i \mid \text{player } j \text{ chose action } a_j \in \mathcal{P}, 1 \leq j \leq N \tag{3}$$

where $(a_1, ..., a_N)$ is the set of pure strategies played by all the players.

Recall that we defined ℓ_P to be the latency of path P and that $\forall i$, $a_i(t)$ corresponds to one specific path in $\mathcal{P}_{com(i)}$ at each time t. From now on, we will write $\ell_{a_i(t)}(f)$ as the latency of the path implied by $a_i(t)$.

$$d_i(a_1, a_2, ..., a_N) = \ell_{a_i}(f),$$

where f is the flow induced by $a_1, a_2, ..., a_N$.

Now, we want to extend the payoff function - or the utility function - to mixed strategies. To do so, let $\mathcal{S}_{m_i}$ denote the simplex of dimension m_i which is the set of m_i-dimensional probability vectors:

$$\mathcal{S}_{m_i} = \{q_i = (q_{i,1}, ..., q_{i,m_i}) \in [0,1]^{m_i} : \sum_{s=1}^{m_i} q_{i,s} = 1\}. \tag{4}$$

For a player associated to commodity i, we write abusively $\mathcal{S}$ for $\mathcal{S}_{m_i}$, i.e. the set of its mixed strategies.

We denote by $K = \mathcal{S}^N$ the space of mixed strategies.

The payoff function d_i defined on pure strategies in equation (3) can be extended to function g_i on the space of mixed strategies K. Note that we consider this function as an expectation on the utility. This is because we assume that flows of information can not be split in our model.

We define the *mean* expected payoff function $g_i : \prod_{j=1}^{N} \mathcal{S}_{m_j} \rightarrow [0,1], 1 \leq i \leq N$, by:

$$\begin{aligned} g_i(q_1, ..., q_N) &= E[r_i | \, \textit{player } j \textit{ employs strategy } q_j, \ 1 \leq j \leq N] \\ &= \textstyle\sum_{j_1,...,j_N} d_i(a, ..., a_N) \times \prod_{z=1}^{N} q_{j,a_j} \end{aligned} \tag{5}$$

where $(q_1, ..., q_N)$ is the set of mixed strategies played by the set of players and E denotes a conditional expectation.

Now that we have defined the mean utility value, we can formally define Nash equilibria.

Definition 3.2. The N-tuple of mixed strategies $(\tilde{q}_1, ..., \tilde{q}_N)$ is said to be a Nash equilibrium (in mixed strategies), if for each i, $1 \leq i \leq N$, we have:

$$\overline{d}_i(\tilde{q}_1, ..., \tilde{q}_{i-1}, \tilde{q}_i, \tilde{q}_{i+1}, ..., \tilde{q}_N) \leq \overline{d}_i(\tilde{q}_1, ..., \tilde{q}_{i-1}, q, \tilde{q}_{i+1}, ..., \tilde{q}_N) \; \forall q \in \mathcal{S} \tag{6}$$

It is well known that every n-person game has at least one Nash equilibrium in mixed strategies [24].

We define $K^* = (\mathcal{S}^*)^N$ where $\mathcal{S}^* = \{\forall i, q_i \in \mathcal{S}_{m_i} |\; q_i$ is a m_i-dimensional probability vector with 1 component unity$\}$ as the corners of the strategy space K. Clearly, K^* can be put in one-to-one correspondence with pure strategies. A N-tuple of actions $(\tilde{a}_1, ..., \tilde{a}_N)$ can be defined to be a pure Nash Equilibrium similarly.

Now the learning problem can be stated as follows: Assume that we play a stochastic repeated game with incomplete information. $q_i[t]$ is the strategy employed by the i^{th} player at instant t. Let $a_i[t]$ and $c_i[t]$ be the action selected and the payoff obtained by player i respectively at time t $(t = 0, 1, 2, \dots)$. Find a decentralized learning algorithm T_i, where $q_i[t+1] = T_i(q_i[t], a_i[t], c_i[t])$, such that $q_i[t] \rightarrow \tilde{q}_i$ as $t \rightarrow +\infty$ where $(\tilde{q}_1, ..., \tilde{q}_N)$ is a Nash equilibrium of the game.

4. Distributed Algorithm

We consider the following learning algorithm, already considered in [20, 26]. The way of doing the update is also called the Linear Reward-Inaction (L_{R-I}) algorithm.

Definition 4.1 (Considered Algorithm)

(1) At every time step, each player chooses an action according to its current Action Probability Vector (APV). Thus, the i^{th} player selects path $s = a_i(t)$ at instant t with probability $q_{i,s}(t)$.

(2) Each player gets to know his utility, based on the set of all actions, for the game. $r_i(t) \in [0, 1] = 1 - \ell_{a_i(t)}(f(t))$.

(3) Each player updates his APV according to the rule:

$$q_i(t+1) = q_i(t) + b \times r_i(t) \times (e_{a_i(t)} - q_i(t)), i = 1, ..., N, \tag{7}$$

where $0 < b < 1$ is a precision parameter and $e_{a_i(k)}$ is a unit vector of dimension m_i with $a_i(t)^{th}$ component unity.

It is easy to see that decisions made by players are completely decentralized, at each time step player i only needs r_i and a_i, respectively his utility and last action, to update his APV.

Notice that, componentwise, Equation (7) can be interpreted as the following 2 case scenario:

$$q_{i,s}(t+1) = \begin{cases} q_{i,s}(t) \; - b \times r_i(t) \times q_{i,s}(t) & \text{if } \; a_i \neq s \\ q_{i,s}(t) \; + b \times r_i(t) \times (1 - q_{i,s}(t)) & \text{if } \; a_i = s \end{cases} \tag{8}$$

4.1. *Evolution over time*

Let $Q[t] = (q_1(t), ..., q_N(t)) \in K$ denote the state of all the players at instant t. Our interest is in the asymptotic behavior of $Q[t]$ and its convergence to a Nash Equilibrium. Clearly, under the learning algorithm specified by (7), $\{Q[t], t \geq 0\}$ is a Markov process.

Observe that this dynamic can also be put in the form

$$Q[t+1] = Q[t] + b \cdot G(Q[t], a[t], r[t]), \tag{9}$$

where $a[t] = (a_1(t), ..., a_N(t))$ denotes the actions selected by all the players at time t and $r[t] = (r_1(t), ..., r_N(t))$ their resulting utilities, for some function $G(., ., .)$ representing the updating, specified by equation (7), that does not depend on b.

Consider the piecewise-constant interpolation of $Q[t], Q^b(.)$, defined by

$$Q^b(k) = Q[t], k \in [tb, (t+1)b], \tag{10}$$

where b is the parameter used in (7).

$Q^b(.)$ belongs to the space of all functions from $\mathbb{R}$ into K. These functions are right continuous and have left hand limits. Now consider the sequence $\{Q^b(.) : b > 0\}$. We are interested in the limit $Q(.)$ of this sequence as $b \to 0$.

4.2. *Approximating the trajectory*

The following is proved in [26]:

Proposition 4.1. *The sequence of interpolated processes $\{Q^b(.)\}$ converges weakly, as $b \to 0$, to $Q(.)$, which is the (unique) solution of the Cauchy problem*

$$\frac{dQ}{dt} = \phi(Q), Q(0) = Q_0 \tag{11}$$

where $Q_0 = Q^b(0) = Q[0]$, and $\phi : K \to K$ is given by

$$\phi(Q) = E[G(Q[t], a[t], c[t])|Q[t] = Q],$$

where G is the function in Equation (9).

Recall that a family of random variable $(Y_t)_{t\in\mathbb{R}}$ weakly converges to a random variable Y, if $E[h(X_t)]$ converges to $E[h(Y)]$ for each bounded and continuous function h. This is equivalent to convergence in distributions.

The proof of Proposition 4.1 in [26], that works for general (even with stochastic payoffs) games, is based on constructions from [21], in turn based on [31], i.e. on weak-convergence methods, non-constructive in several aspects, and does not provide error bounds.

It is actually possible to provide a bound on the error between $Q(t)$ and the expectation of $Q^b(.)$ in some cases. We will say a few words about that before getting back to the evolution of the system over time in section 5.

4.3. *Bounds on the approximation error*

Theorem 1. Let $Q[.]$ be a process defined by an equation of type (9), and let $Q^b(.)$ be the corresponding piecewise-constant interpolation, given by (10). Assume that $E[G(Q[t], a[t], c[t])] = \phi(E[Q[t]])$ for some function ϕ of class $\mathcal{C}^1$.

Let $\epsilon(k)$ be the error in approximating the expectation of $Q^b(k)$ by $Q(t)$:

$$\epsilon(k) = ||E[Q^b(k)] - Q(k)||,$$

where $Q(.)$ is the (unique) solution of the Cauchy problem

$$\frac{dQ}{dt} = \phi(Q), Q(0) = Q_0, \tag{12}$$

where $Q_0 = Q^b(0) = Q[0]$.

We have

$$\epsilon(k) \leq Mb\frac{e^{\Lambda k} - 1}{2\Lambda},$$

for k of the form $k = tb$, $t \in \mathbb{N}$, where $\Lambda = \max_{i,\ell} ||\frac{\partial \phi}{\partial q_{i,\ell}}||$, and M is a bound on the norm of $Q''(k) = \frac{d\phi(Q(k))}{dk}$.

Proof. The general idea of the proof is to consider the dynamic (9), as an Euler discretization method of the ordinary differential equation (12), and then use some classical numerical analysis techniques to bound the error at time t.

Indeed, by hypothesis we have

$$\begin{aligned} E[Q[t+1]] &= E[Q[t]] + b \cdot E[G(Q[t], a[t], c[t])] \\ &= E[Q[t]] + b\phi(E[Q[t]]). \end{aligned}$$

Suppose that $\phi(.)$ is Λ-Lipschitz, then we know that for some positive Λ,

$$||\phi(x) - \phi(x')|| \leq \Lambda ||x - x'||$$

From Taylor-Lagrange inequality, we can suppose $\Lambda = \max_{i,\ell} ||\frac{\partial \phi}{\partial q_{i,\ell}}||$, if ϕ is of class $\mathcal{C}^1$.

We can write,

$$\begin{aligned} \epsilon((t+1)b) &= ||E[Q^b((t+1)b)] - Q((t+1)b)|| \\ &\leq ||E[Q^b((t+1)b)] - E[Q^b(tb)] - b\phi(Q(tb))|| \\ &\quad +||E[Q^b(tb)] - Q(tb)|| + ||Q(tb) - Q((t+1)b) + b\phi(Q(tb))|| \\ &= ||b\phi(E[Q^b(tb)]) - b\phi(Q(tb))|| + \epsilon(tb) + ||b\phi(Q(tb)) - \int_{tb}^{(t+1)b} \phi(Q(t'))dt'|| \\ &\leq \Lambda b||E[Q^b(tb)] - Q(tb)|| + \epsilon(tb) + e(tb) \\ &\leq (1 + \Lambda b)\epsilon(tb) + e(tb) \end{aligned}$$

where $e(tb) = ||b\phi(Q(tb)) - \int_{tb}^{(t+1)b} \phi(Q(t'))dt'||$.

From Taylor-Lagrange inequality, we know that $e(tb) \leq H = M\frac{b^2}{2}$, where M is a bound on the norm of $Q''(k) = \frac{d\phi(Q(k))}{dk}$.

From an easy recurrence on t, (sometimes called Discrete Gronwall's Lemma, see e.g. [6]), using inequality $\epsilon((t+1)b) \leq (1+\Lambda b)\epsilon(tb) + H$, we get that

$$\begin{aligned} \epsilon(tb) &\leq (1+\Lambda b)^t \epsilon(0) + H\tfrac{(1+\Lambda b)^t - 1}{1+\Lambda b - 1} \\ &\leq H\tfrac{e^{t\Lambda b}-1}{\Lambda b} \\ &= Mb\tfrac{e^{t\Lambda b}-1}{2\Lambda} \end{aligned}$$

using that $(1+u)^t \leq e^{tu}$ for all $u \geq 0$, and $\epsilon(0) = 0$. □

5. Expected utilities and variation of the system over time

We defined in section 3 the utility functions $d_i(a[t])$ and $g_i(Q[t])$ respectively on pure and mixed strategies. In our settings, players use one unique path at each game, for considering that, we define a function that expresses the expected utility on a certain path for a player at a given game.

Let $h_{i,s}(Q)$ be the expected utility for player i if he decides to play the pure strategy (path) s, and players $j \neq i$ play (mixed) strategy q_j. Formally,

$$h_{i,s}(q_1, ..., q_{i-1}, s, q_{i+1}, ..., q_n) = E[r_i \mid Q^{-i}, a_i = s] \tag{13}$$

where Q^{-i} stands the set of actions of all players but player i.

Let $\overline{h_i}(Q)$ be the mean value of $h_{i,s}(Q)$ according to q_i, in the sense that

$$\overline{h_i}(Q) = \sum_{s'} q_{i,s'} h_{i,s'}(Q).$$

5.1. *Evolution of players over time*

By Theorem 1, the limit Q of the interpolated process Q^b given by (10) satisfies the ODE (11). Recall that Q consists of N probability vectors. Solutions of the ODE live in the set K defined previously. As discussed in section 3, the points in K^* represent pure strategies and are referred to as corners of K. If we note m_i the number of paths available to player i, we have that Q contains $m_1 + m_2 + ... + m_N$ components which are denoted by $q_{i,s}, 1 \leq i \leq N, 1 \leq s \leq m_i$. ϕ also has the same number of components which we denote as $\phi_{i,s}$.

Using (8), we can rewrite $\phi_{i,s}$ in the general case as follows.

$$\begin{aligned} \phi_{i,s} &= q_{i,s}(1-q_{i,s})h_{i,s} - \textstyle\sum_{s' \neq s} q_{i,s'} q_{i,s} h_{i,s'} \\ &= q_{i,s}\left(\textstyle\sum_{s' \neq s} q_{i,s'} h_{i,s} - \textstyle\sum_{s' \neq s} q_{i,s'} h_{i,s'}\right) \\ &= q_{i,s} \textstyle\sum_{s'} (h_{i,s} - q_{i,s'} h_{i,s'}), \end{aligned}$$

using the fact that $\sum_{s' \neq s} q_{i,s'} = 1 - q_{i,s}$.

We obtain

$$\phi_{i,s} = \frac{dq_{i,s}}{dt} = q_{i,s}(h_{i,s}(Q) - \overline{h_i}(Q)). \tag{14}$$

Hence, the dynamics given by the ODE (11) can be rewritten, componentwise as:

$$\frac{dq_{i,s}}{dt} = q_{i,s} \sum_{s'} q_{i,s'}(h_{i,s}(Q) - h_{i,s'}(Q)). \tag{15}$$

Note that this is a replicator equation, a well-known and studied dynamic in evolutionary game theory [17, 33].

5.2. *Convergence in linear Wardrop games*

The following so-called folk theorem characterizes the solutions of the ODE and hence characterizes the long term behavior of the learning algorithm [17, 26].

Theorem 2. The following are true for the solutions of the replicator equation (15):

- All corners of space K are stationary points.
- All Nash equilibria are stationary points.
- All strict Nash equilibria are asymptotically stable.
- All stable stationary points are Nash equilibria.

From this theorem, we can conclude that, if we put aside the case of trivial (constant) dynamics, the dynamic (15), and hence the learning algorithm when b goes to 0, will never converge to a point in K which is not a Nash equilibrium.

However, for general games, there is no convergence in the general case [26].

We will now show that for linear Wardrop games, there is always convergence. It will then follow that the learning algorithm we are considering here converges towards Nash equilibria, i.e. solves the learning problem for linear Wardrop games.

First, we specialize the dynamic for our routing games, regarding latency functions, we have

$$\ell_{a_i}(f) = \sum_{e \in a_i} \ell_e(\lambda_e) = \sum_{e \in a_i} [\beta_e + \alpha_e w_i + \alpha_e \sum_{j \neq i} \mathbf{1}_{e \in a_j} w_j] \tag{16}$$

where $\mathbf{1}_{e \in a_j}$ is 1 whenever $e \in a_j$, 0 otherwise. $e \in a_i$ means edge e belongs to the sequence of edges - or path - induced by a_i. Let us also introduce the following notation:

$$p(e, q_i) = \sum_{s=1}^{m_i} q_{i,s} \times \mathbf{1}_{e \in s} \tag{17}$$

which denotes the probability for player i of using edge e according to his probability vector q_i.

Using this, we can write the expected utility for player i when using path s as:

$$h_{i,s}(Q) = \sum_{e \in s} \left[\alpha_e \left(w_i + \sum_{j \neq i} p(e, q_j) w_j \right) + \beta_e \right]$$

Now let us provide a sufficient condition on $Q(t)$ for the game to converge to some point in K with the following theorem.

Theorem 3 (Extension of Theorem 3.3 from [26]) *Suppose there is a non-negative function*

$$F : K \to \mathbb{R}$$

such that for some constants $x_i > 0$*, for all* i*,* s*,* Q*,*

$$\frac{\partial F}{\partial q_{i,s}}(Q) = x_i \times h_{i,s}(Q). \tag{18}$$

Then the learning algorithm, for any initial condition in $K - K^*$*, always converges to a Nash Equilibrium.*

Proof. Consider the variation of F along the trajectories of the ODE. We have

$$\begin{aligned} \tfrac{dF(Q(t))}{dt} &= \textstyle\sum_{i,s} \frac{\partial F}{\partial q_{i,s}} \frac{dq_{i,s}}{dt} \\ &= \textstyle\sum_{i,s} \frac{\partial F}{\partial q_{i,s}}(Q) q_{i,s} \sum_{s'} q_{i,s'}[h_{i,s}(Q) - h_{i,s'}(Q)] \\ &= \textstyle\sum_{i,s} x_i h_{i,s}(Q) q_{i,s} \sum_{s'} q_{i,s'}[h_{i,s}(Q) - h_{i,s'}(Q)] \\ &= \textstyle\sum_i x_i \sum_s \sum_{s'} q_{i,s} q_{i,s'}[h_{i,s}(Q)^2 - h_{i,s}(Q) h_{i,s'}(Q)] \\ &= \textstyle\sum_i x_i \sum_s \sum_{s'>s} q_{i,s} q_{i,s'}[h_{i,s}(Q) - h_{i,s'}(Q)]^2 \\ &\geq 0 \end{aligned} \tag{19}$$

Thus F is non decreasing along the trajectories of the ODE and, due to the nature of the learning algorithm, all solutions of the ODE (15), for initial conditions in K will be confined to K.

Hence from the Lyapunov Stability theorem (see e.g. [16], page 194), if we note Q^* an equilibrium point, we can define $L(Q) = F(Q) - F(Q^*)$ as a Lyapunov function of the game. Asymptotically, all trajectories will be in the set $K' = \{Q^* \in K : \frac{dF(Q^*)}{dt} = 0\}$.

From (19), we know that $\frac{dF(Q^*)}{dt} = 0$ implies $q_{i,s}q_{i,s'}[h_{i,s}(Q) - h_{i,s'}(Q)] = 0$ for all i, s, s'. Such a Q^* is consequently a stationary point of the dynamics.

Since from Theorem 2, all stationary points that are not Nash equilibria are unstable, the theorem follows. □

Given its nature, this kind of function is often referred to as a *potential* function in game theory [2, 27, 23]. In what follows, we call that function F a potential function for our game.

5.3. *A potential function*

We now provide a potential function for our game that satisfies Equation (18).

Proposition 5.1. *For the definition we gave in this paper of linear Wardrop games, the following function F satisfies the hypothesis of the previous theorem:*

$$\begin{aligned} F(Q) = \textstyle\sum_{e\in E}\Bigg[\ & \beta_e\Bigg(\textstyle\sum_{j=1}^{N} w_j \times p(e,q_j)\Bigg)+ \\ & \tfrac{\alpha_e}{2}\Bigg(\textstyle\sum_{j=1}^{N} w_j \times p(e,q_j)\Bigg)^2 + \\ & \alpha_e\Bigg(\textstyle\sum_{j=1}^{N} w_j^2 \times p(e,q_j) \times (1-\tfrac{p(e,q_j)}{2})\Bigg)\Bigg] \end{aligned} \tag{20}$$

Notice that the hypothesis of affine cost functions is crucial here.

Proof. We first rewrite the potential function F as $F(Q) = \sum_{e\in E} A_e(Q)$ in order to lighten the next few lines.

$\frac{\partial F}{\partial q_{i,s}}(Q) \quad = \quad \frac{\partial \sum_{e\in E} A_e(Q)}{\partial q_{i,s}} = \sum_{e\in E} \frac{\partial A_e(Q)}{\partial q_{i,s}}$, which can be rewritten as

$\frac{\partial F}{\partial q_{i,s}}(Q) \quad = \quad \sum_{e\in E} \frac{\partial A_e(Q)}{\partial p(e,q_i)} \times \frac{\partial p(e,q_i)}{\partial q_{i,s}}$.

From (17), we get that $\frac{\partial p(e,q_i)}{\partial q_{i,s}} = \mathbf{1}_{e\in s}$, we can rewrite

$$\frac{\partial F}{\partial q_{i,s}}(Q) = \sum_{e\in E} \frac{\partial A_e(Q)}{\partial p(e,q_i)} \times \mathbf{1}_{e\in s} = \sum_{e\in s} \frac{\partial A_e(Q)}{\partial p(e,q_i)} \tag{21}$$

Let us now develop the derivative of each term of the sum and come back to (21) in the end, we have

$$\begin{aligned} \tfrac{\partial A_e(Q)}{\partial p(e,q_i)} &= \beta_e \times w_i + \alpha_e \times w_i(\textstyle\sum_{j=1}^{N} w_j \times p(e,q_j)) + \alpha_e(w_i^2(1-p(e,q_i))) \\ &= \beta_e \times w_i + \alpha_e \times w_i(\textstyle\sum_{j\neq i} w_j \times p(e,q_j)) + \alpha_e w_i^2, \end{aligned} \tag{22}$$

which corresponds to the expected price on edge e for player i when he knows he is going to use this edge (it belongs to the path he chooses).

For $\frac{\partial F}{\partial q_{i,s}}$, this finally leads to:

$$\frac{\partial F}{\partial q_{i,s}} = \sum_{e\in s} \frac{\partial A_e(Q)}{\partial p(e,q_i)} = \sum_{e\in s} \beta_e \times w_i + \alpha_e \times w_i(\sum_{j\neq i} w_j \times p(e,q_j)) + \alpha_e w_i^2$$

$$\frac{\partial F}{\partial q_{i,s}}(Q) = w_i \times h_{i,s}(Q) \tag{23}$$

We showed that Equation (18) holds here. The constants x_i of the theorem are simply, in this case, the weights w_i of the players. This completes the proof and confirms that F is a convenient function for the game. □

We now provide a short example on why latency functions need to be linear with these settings.

Proposition 5.2. *Suppose for example that cost functions were quadratic :*

$$\ell_e(\lambda_e) = \alpha_e \lambda_e^2 + \beta_e \lambda_e + \gamma_e,$$

with $\alpha_e, \beta_e, \gamma_e \geq 0$*,* $\alpha_e \neq 0$*.*

There can not exist a function F *of class* $\mathcal{C}^2$ *that satisfies* (18) *for all* i*,* s*,* Q*, and general choice of weights* w_i*.*

Proof. By Schwartz theorem, we must have

$$\frac{\partial}{\partial q_{i',s'}}(\frac{\partial F}{\partial q_{i,s}}) = \frac{\partial}{\partial q_{i,s}}(\frac{\partial F}{\partial q_{i',s'}}),$$

and hence

$$W_i \frac{\partial h_{i,s}}{\partial q_{i',s'}} = W_{i'} \frac{\partial h_{i',s'}}{\partial q_{i,s}},$$

for all i, i', s, s', for some constants $W_i, W_{i'}$. It is easy to see that this doesn't hold for general choice of Q and weights $(w_i)_i$ in this case. □

Coming back to our model (with affine costs), we obtain the following result:

Theorem 4. For linear Wardrop games, for any initial condition in $K - K^*$, the considered learning algorithm converges to a (mixed) Nash equilibrium.

6. Conclusion

In this paper we considered a game based on the classical Wardrop traffic model, with finitely any players and we introduced some specific dynamical aspects.

We considered an update algorithm proposed by [26] and we proved that the learning algorithm depicted is able to learn mixed Nash equilibria of the game, extending several results of [26].

To do so, we proved that the learning algorithm is asymptotically equivalent to an ordinary differential equation, which turns out to be a replicator equation. Using a folk theorem from evolutionary game theory, one knows that if the dynamics converges, it will be towards some Nash equilibria. We proved using a Lyapunov function argument that the dynamic converges in our considered settings.

We established some bounds on the error in approximating the discrete process by a continuous function, based on the analysis of the dynamics and numerical analysis arguments in some special cases. Our next intent is to be more specific about the convergence time and to lower it.

We believe that this paper exhibits a very nice example of distributed systems whose study is done through a continuous view of a discrete system. With the analysis of the distributed learning algorithm players use and considering how they adapt their strategies, we obtain a trajectory for the whole system. Whereas the agents' rules are quite simple and based on local views, we showed that if they all apply the learning algorithm, the system will eventually reach an equilibrium state.

We also intend to pursue our investigations on the computational properties of distributed systems through similar continuous time dynamical system views.

References

[1] E. Altman, Y. Hayel, and H. Kameda. Evolutionary Dynamics and Potential Games in Non-Cooperative Routing. In *Wireless Networks: Communication, Cooperation and Competition (WNC3 2007)*, 2007.

[2] M. Beckmann, C. B. McGuire, and C. B. Winsten. Studies in the economics of transportation. 1956.

[3] Petra Berenbrink, Tom Friedetzky, Leslie Ann Goldberg, Paul Goldberg, Zengjian Hu, and Russell Martin. Distributed Selfish Load Balancing. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 354–363, New York, NY, USA, 2006. ACM.

[4] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. How much can taxes help selfish routing? In *Proceedings of the 4th ACM Conference on Electronic Commerce (EC-03)*, pages 98–107, New York, June 9–12 2003. ACM Press.

[5] R. Cominetti, J.R. Correa, and N.E. Stier-Moses. Network Games with Atomic Players. *Automata, Languages and Programming: Proceedings of the 33rd International Colloquium, Venice, Italy*, 4051:525–536, 2006.

[6] Jean-Pierre Demailly. *Analyse Numérique et Equations Différentielles*. Presses Universitaires de Grenoble, 1991.

[7] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence Time to Nash Equilibria. *30th International Conference on Automata, Languages and Programming (ICALP)*, pages 502–513, 2003.

[8] Eyal Even-Dar, Alexander Kesselman, and Yishay Mansour. Convergence Time to Nash equilibrium in Load Balancing. *ACM Transactions on Algorithms*, 3(3), 2007.

[9] Eyal Even-Dar and Yishay Mansour. Fast Convergence of Selfish Rerouting. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 772–781, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[10] S. Fischer, H. Räcke, and B. Vöcking. Fast Convergence to Wardrop Equilibria by Adaptive Sampling Methods. *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 653–662, 2006.

[11] S. Fischer and B. Vocking. On the Evolution of Selfish Routing. *Algorithms–ESA 2004: 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004, Proceedings*, 2004.

[12] S. Fischer and B. Vöcking. Adaptive Routing with Stale Information. *Proceedings of the twenty-fourth annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 276–283, 2005.

[13] L. Fleischer. Linear Tolls Suffice: New Bounds and Algorithms For Tolls in Single Source Networks. *Theoretical Computer Science*, 348(2-3):217–225, 2005.

[14] Paul W. Goldberg. Bounds for the Convergence Rate of Randomized Local Search in a Multiplayer Load-Balancing Game. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 131–140, New York, NY, USA, 2004. ACM.

[15] C. Harris. On the Rate of Convergence of Continuous-Time Fictitious Play. *Games and Economic Behavior*, 22(2):238–259, 1998.

[16] Morris W. Hirsch, Stephen Smale, and Robert Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos.* Elsevier Academic Press, 2003.

[17] J. Hofbauer and K. Sigmund. Evolutionary Game Dynamics. *Bulletin of the American Mathematical Society*, 4:479–519, 2003.

[18] J. Hofbauer and S. Sorin. Best Response Dynamics for Continuous Zero-Sum Games. *Discrete and Continuous Dynamical Systems-Series B*, 6(1), 2006.

[19] E. Koutsoupias and C. Papadimitriou. Worst-case Equilibria. In *STACS99*, pages 404–413, Trier, Germany, 4–6March 1999.

[20] M.A.L. Thathachar K.S. Narendra. Learning Automata: An Introduction. *Englewood Cliffs: Prentice Hall*, 1989.

[21] H. J. Kushner. *Approximation and Weak Convergence Methods for Random Processes, with Applications to Stochastic Systems Theory.* Cambridge, MA: MIT Press, 1984.

[22] L. Libman and A. Orda. Atomic Resource Sharing in Noncooperative Networks. *Telecommunication Systems*, 17(4):385–409, 2001.

[23] D. Monderer and L. S. Shapley. Potential games. *Games and Economics Behavior*, 14:124–143, 1996.

[24] John F. Nash. Equilibrium Points in n-person Games. *Proc. of the National Academy of Sciences*, 36:48–49, 1950.

[25] A. Orda, R. Rom, and N. Shimkin. Competitive Routing in Multi-user Communication Networks. *IEEE/ACM Transactions on Networking (TON)*, 1(5):510–521, 1993.

[26] M.A.L. Thathachar P.S. Sastry, V.V. Phansalkar. Decentralized Learning of Nash Equilibria in Multi-Person Stochastic Games With Incomplete Information. *IEEE transactions on system, man, and cybernetics*, 24(5), 1994.

[27] Robert W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International journal of Game Theory 2*, pages 65–67, 1973.

[28] T. Roughgarden. How unfair is optimal routing? *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 203–204, 2002.

[29] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, March 2002.

[30] L. Olbrich S. Fischer and B. Vöcking. Approximating Wardrop Equilibria with Finitely Many Agents. *DISC07*, 2007.

[31] D.W. Stroock and SRS Varadhan. *Multidimensional Diffusion Processes.* Springer, 1979.

[32] J. Wardrop. Some Theoretical Aspects of Road Traffic Research. *Proceedings of the Institution of Civil Engineers, Part II*, 1(36):352–362, 1952.

[33] Jörgen W. Weibull. *Evolutionary Game Theory.* The MIT Press, 1995.