Discrete-Time Selfish Routing Converging to the Wardrop Equilibrium

Antonio Pietrabissa[1] and Lorenzo Ricciardi Celsi[1]

Abstract—This paper presents a discrete-time, distributed and non-cooperative routing algorithm, which is proved, via Lyapunov arguments, to asymptotically converge to a specific equilibrium condition among the traffic flows over the network paths, known as Wardrop equilibrium. This convergence result improves the discrete-time algorithms in the literature, which achieve approximate convergence to the Wardrop equilibrium. Numerical simulations show the effectiveness of the proposed approach.

***Index Terms*— Wardrop equilibrium, Lyapunov stability, selfish routing.**

I. Introduction

THE paper presents a discrete-time selfish routing algorithm, proving the convergence to a Wardrop equilibrium, which is a game-theoretical concept originally introduced for network games when modelling transportation networks with congestion [1], and which can be informally described as the situation when "the journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route" [2]. In particular, routing is said to be *selfish* when, in selecting a path to travel from source to destination, each player does not take into account the additional congestion that it causes other players to experience as a result of its actions [3].

In routing scenarios, the system performance crucially depends on effectively dividing up traffic across the admissible network paths [4][5]. We therefore refer to the model of a time-invariant communication topology where a certain amount of traffic load, or *flow demand*, has to be routed from a source node to a destination node over a set of admissible paths in such a way as to ensure that the network stays *balanced*, i.e., the used network paths yield minimal latencies. Network traffic is modelled as an infinite stream of infinitely-many arriving agents, each being responsible for an infinitesimal amount of traffic, or *job*. Each agent is then a decision maker, yielding the *distributed* nature of the modelling setup.

Non-cooperative algorithms entail the presence of several decision makers which optimize their own response time independently of the other ones, since cooperation is not allowed. In case a finite number of agents is considered, a Nash equilibrium condition is reached when no agent can receive any further benefit by changing its own decision unilaterally. In other words, the stability of the network under said algorithms is analysed in terms of reaching a distribution of the traffic flow such that no single agent can move to any other path with a smaller number of jobs. In case the number of agents is infinite, a combination of flows such that no agent can improve its latency by deviating unilaterally yields a Wardrop equilibrium for the network. Indeed, a Nash equilibrium is said to become a Wardrop equilibrium whenever the number of agents is assumed to be infinite [6][7]. In turn, *dynamic* algorithms are required for settings where the load distribution is not known *a priori* as they succeed in performing the decision-making process based on the current state of the system, which is generally made available via feedback.

In this respect, we propose a discrete-time, distributed, non-cooperative, dynamic selfish routing algorithm, designed so that a Wardrop equilibrium is reached by the agents. To the authors' knowledge, the proposed algorithm is the first discrete-time selfish routing algorithm which is proved to converge asymptotically to the exact Wardrop equilibrium.

The paper is organized as follows: Section II presents the state of the art on selfish routing and Wardrop equilibrium, and highlights the proposed novelties; Section III introduces the routing problem; Section IV presents the proposed discrete-time control law and proves its convergence to the Wardrop equilibrium; Section V shows the results of numerical simulations; in Section VI the conclusions are drawn.

II. Related work and Proposed Innovation

This work falls within the scope of algorithmic game theory, which provides a fruitful modelling framework for many applications, e.g., in transport, network engineering and computer science problems, and currently represents a very active area of research [8]-[11].

Wardrop equilibria have been relied upon in order to deal with several different types of congested environments, such as: routing in road traffic networks [12][13], traffic engineering in communication networks [14][15], load balancing in distributed computational grids [16] and in wired and wireless networks [17][18]. In particular, as useful reads, we recall [15] for a clear definition of the concept of population, and [7] for a detailed discussion distinguishing between approximate and exact equilibria. In the classical problem formulation [19], a congested network represented by a graph with nodes and edges is considered. A non-decreasing latency function of traffic is associated with each edge, representing the cost of the edge (e.g., its congestion level). The network serves several commodities, characterized by a given amount of traffic load to be routed from a source to a destination and balanced over the set of admissible network paths. The job vector is the amount of traffic that is allocated for each commodity and for each path connecting the (source, destination) pairs. Each agent has the possibility to distribute its own flow among a set of admissible paths.

A job vector in which, for all commodities, the latencies of all used paths are equal is called Wardrop equilibrium. The Wardrop equilibrium can be computed by centralized algorithms in polynomial time [20]. This paper is aimed at achieving a Wardrop equilibrium via a discrete-time dynamic and distributed algorithm.

This concept has been adopted for solving selfish routing problems:

[1] A. Pietrabissa and L. Ricciardi Celsi are with the Department of Computer, Control and Management Engineering of the University of Rome "Sapienza", via Ariosto 25, 00162 Rome, Italy

email: {pietrabissa, ricciardicelsi}@diag.uniroma1.it.

in this respect, quite a few works in the literature resort to a similar approach as the one proposed in this paper, such as [21], [22], and [23] relative to a continuous-time setting, and, instead, [24], [25], [26], and [27] relative to a discrete-time setting where the authors only prove convergence to an approximated neighborhood of the Wardrop equilibrium. In [24], the authors develop a round-based distributed algorithm with a finite number of agents. Each agent is responsible for one commodity and has a set of admissible paths among which it may distribute its traffic. In this respect, the population of agents responsible for rerouting the traffic is instructed to *learn* a Wardrop equilibrium efficiently by relying on suitable adaptive sampling methods. A bulletin board is assumed to be available, where the traffic assignments are updated at every round. With a similar bulletin board scenario, in [25] a distributed routing algorithm is presented. At each round, each agent "samples" a different path and compares the latency of the newly chosen path with its own current latency. If the comparison shows that the agent can improve its latency by rerouting its own portion of network traffic, then it "migrates" to the better path with some probability depending on the latency improvement. Convergence results are given both when the agents base their decisions on up-to-date information and when the information is "stale" (i.e., considering delays in the bulletin board update). In [26], a distributed and asynchronous routing algorithm is proposed, relying on an estimation of the latencies of all paths and on a reinforcement learning algorithm to update the probabilities of transmission towards the different paths. In [27] a round-based version of the adaptive routing algorithm with stale information shown in [25] is proposed. All the cited algorithms converge to an approximate Wardrop equilibrium. Yet, none of these works yields exact convergence to a Wardrop equilibrium. In [28] and [29], the convergence to an exact Wardrop equilibrium is tackled in a no-regret learning framework by assuming the latency values to be discounted over time; such an assumption is limiting in non-atomic congestion games and in routing applications as it implies that players value future time less than current time.

The main innovation of the current work is therefore the design of a discrete-time selfish routing algorithm converging to an exact Wardrop equilibrium. In particular, the proposed algorithm is designed so as to dynamically learn a Wardrop equilibrium efficiently and in a distributed fashion**.** We adopt the problem formulation proposed in [17] and rely on the algorithm proposed in [27].

Hence, the scenario considered in this paper requires a non-cooperative dynamic selfish routing approach. Yet, for the sake of completeness, we recall that such a setup can also be adapted to load balancing purposes in game-theoretic frameworks, where the problem can be described as a dynamic load balancing game, with users distributing their loads in a non-cooperative and selfish fashion (e.g., see [30], [31], and [32]).

III. Selfish Routing Problem

As described above, this paper further develops a well-known model for selfish routing [27], where an infinite population of agents carries an infinitesimal amount of load each. Standard notation is used throughout the paper, with $|\cdot|$ denoting the cardinality operator.

We are given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the finite set of vertices or nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges or links. Let $\mathcal{I}$ denote a set of commodities with constant traffic demands $d^i > 0, \forall i \in \mathcal{I}$, generally expressed in jobs per unit of time, with total demand $d \coloneqq \sum_{i\in\mathcal{I}} d^i$. Let also $\mathcal{P}^i$ denote a set of paths (or providers), which serve the traffic flows for every commodity $i \in \mathcal{I}$ and let $\mathcal{P} = \bigcup_{i\in\mathcal{I}} \mathcal{P}^i$ be the set of all the network paths. Each source node (e.g., s^i for some $i \in \mathcal{I}$) is connected by the network to the destination node (i.e., t^i) through the set of paths $\mathcal{P}^i$. A given path $p \in \mathcal{P}^i$ includes a set of links; let $\mathcal{P}_e$ be the set of paths including the edge e, i.e., $\mathcal{P}_e \coloneqq \{p \in \mathcal{P} \mid e \in p\}, \forall e \in \mathcal{E}$, and let $\mathcal{E}^i$ be the set of edges included in the paths of $\mathcal{P}^i$, i.e., $\mathcal{E}^i \coloneqq \{e \in p | p \in \mathcal{P}^i\}$. Let us also define the maximum path length as $\lambda^i \coloneqq \max_{p\in\mathcal{P}^i} |p|$ and the maximum number of paths including an edge as $\eta^i \coloneqq \max_{e\in\mathcal{E}^i} |\mathcal{P}_e|$.

The definition of *agent* is also required. As defined, for instance, in [19], each agent is an infinitesimal portion of a specified commodity. Let x_p^i be the volume of the agents, or bandwidth, of commodity i relying on a path $p \in \mathcal{P}^i$. In the scenario considered in this paper, the vector $\boldsymbol{x} = (x_p^i)_{p\in\mathcal{P}^i, i\in\mathcal{I}}$ can be defined as the *flow vector* or *population share*, whose components specify the overall amount of traffic per unit of time flowing along path $p \in \mathcal{P}^i$ and associated with commodity $i \in \mathcal{I}$. Let $x_p \coloneqq \sum_{i\in\mathcal{I}} x_p^i$, $x_e^i \coloneqq \sum_{p\in\mathcal{P}^i\cap\mathcal{P}_e} x_p^i$ and $x_e \coloneqq \sum_{i\in\mathcal{I}} x_e^i$ denote the total traffic flow over path $p \in \mathcal{P}$, the traffic flow of commodity i over edge e and the total traffic flow over edge e, respectively.

Definition 1. The feasible state space, i.e., the closed set of feasible flow vectors, is

$$\mathcal{X} \coloneqq \{\boldsymbol{x} \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{I}|} \mid x_e \geq 0, \forall e \in \mathcal{E}, \textstyle\sum_{p\in\mathcal{P}^i} x_p^i = d^i, \forall i \in \mathcal{I}\}. \tag{1}$$

■

A metric of interest is the average response time required by the path $p \in \mathcal{P}$ for serving an amount of traffic equal to x_p.The response time grows with the considered traffic flow and thus it is a reliable indicator of the path congestion status. Hence, this quantity is defined as the *latency function* associated with the path $p \in \mathcal{P}$ which is a non-negative function $l_p(\boldsymbol{x}) : [0, d] \to \mathbb{R}_{\geq 0}$. The path latency is the sum of the latencies of the edges of the path, denoted with $l_e(x_e) : [0, d] \to \mathbb{R}_{\geq 0}$, i.e., $l_p(\boldsymbol{x}) = \sum_{e\in p} l_e(x_e), \forall p \in \mathcal{P}$. The shape of the latency functions depends on the application considered. One strength of the proposed approach is that the agents only rely on measures of the latency functions, which are not required to be explicitly modelled.

Note that the latency of a path $p \in \mathcal{P}$ is a function of the flow vector $\boldsymbol{x}$, and the latency of an edge is a function of its flow over edge e, i.e., x_e. The latency functions are assumed to have the following properties.

Assumption 1. The latency functions $l_e(\xi)$, $e \in \mathcal{E}$, are Lipschitz continuous and strictly increasing over the interval $[0, d]$. We also define β_e as the local Lipschitz constant of l_e, and $\beta_{max} \coloneqq \max_{e\in\mathcal{E}} \beta_e$.■

Assumption 1 is a reasonable restriction, since the response time of a path generally increases with the total amount of traffic flow routed onto that path.

The agents' aim is that of minimizing their personal latency selfishly without considering the impact on the global situation. One usually assumes that the agents will converge to some allocation in which no agent can improve its latency by deviating unilaterally. A useful characterization of such an equilibrium goes back to Wardrop. In particular, a single request of the flow is approximately considered as an agent: in fact, even if the number of requests is finite, if the flow rates are sufficiently high, the population acceptably approximates the infinite population constraint required by Wardrop theory [1].

The routing problem is formulated below as the problem of determining the strategies which will lead the flow vector to reach a Wardrop equilibrium. In Wardrop theory, *stable* flow assignments are the ones in which no agent (i.e., no "small" portion of a commodity directed from a source to a destination) can improve its situation by changing its strategy (i.e., the set of used paths) unilaterally. This objective is achieved if all agents reach a Wardrop equilibrium.

Definition 2 ([24]). A feasible flow vector $\boldsymbol{x}$ is at a Wardrop equilibrium for an instance of the considered routing game if, for each path $p \in \mathcal{P}^i$ such that $x_p^i > 0$, the following relation holds: $l_p(\boldsymbol{x}) \leq l_q(\boldsymbol{x}), \forall q \in \mathcal{P}^i, \forall i \in \mathcal{I}$. ■

In practice, at the Wardrop equilibrium, the latencies of all the loaded paths have the same value: therefore, provided that the latency functions properly represent the path performances, a fair exploitation of the network resources is achieved by driving the flows towards a Wardrop equilibrium.

In the framework of researches on Wardrop equilibria, a key role is played by the Beckmann, McGuire, and Winsten potential [20]:

$$\Phi(\boldsymbol{x}) \coloneqq \sum_{e \in \mathcal{E}} \int_0^{x_e} l_e(\xi) d\xi, \tag{2}$$

whose properties are summarised in Property 1 below.

Property 1 ([25]). Under Assumption 1, the potential (2) is continuous and the following properties hold:

i) there exists a unique flow $\boldsymbol{x}_{min}$, over the set of feasible flows, minimizing Φ;
ii) correspondingly, there exists a unique positive minimum $\Phi_{min} = \Phi(\boldsymbol{x}_{min})$;
iii) in $\boldsymbol{x}_{min}$, no agent can improve its own latency unilaterally, i.e., $\boldsymbol{x}_{min}$ is at Wardrop equilibrium. ■

Property 1 states that, in the considered scenario, the set $\mathcal{X}_\mathcal{W}$ collapses into a unique Wardrop equilibrium with jobs vector $\boldsymbol{x}_{min}$, hereafter denoted with $\boldsymbol{x}_\mathcal{W}$.

IV. Discrete-Time Control Law and Algorithm Convergence

For the sake of presentation clarity, we choose to limit the analysis to the single-commodity case only, i.e., $|\mathcal{I}| = 1$, and, therefore, the index i will be neglected.

A. Proposed Control Law

The system dynamics is therefore expressed component-wise by

$$x_p[k+1] = x_p[k] + \tau \sum_{q \in \mathcal{P}} \left(r_{qp}[k] - r_{pq}[k] \right), \\ k = 0,1, \dots, \forall p \in \mathcal{P}, \tag{3}$$

where τ is the sampling period and $r_{pq}[k]$ is the so-called *migration rate* from path $p \in \mathcal{P}$ to path $q \in \mathcal{P}$.

Inspired by the algorithm in [24], the migration rate is defined as

$$r_{pq}[k] = x_p[k] \sigma_{pq}[k] \mu_{pq}[k], \tag{4}$$

where

$$\sigma_{pq}[k] > 0, \forall p, q \in \mathcal{P}, \tag{5}$$

is the control gain, which sets the rate with which the population share of path p migrates to path q, and $\mu_{pq}[k]$ is the so-called *migration policy*, representing the decision whether (and in which percentage) the population share assigned to path p migrates to path q.

Throughout the paper, we consider the following initial conditions:

$$\begin{cases} x_p[0] \geq 0, \forall p \in \mathcal{P}, \\ \sum_{p \in \mathcal{P}} x_p[0] = d. \end{cases} \tag{6}$$

There is a variety of migration policies used in the literature for continuous-time algorithms, e.g., the *better response* policy

$$\mu_{pq}[k] = \begin{cases} 0 & \text{if } l_p(\boldsymbol{x}[k]) - l_q(\boldsymbol{x}[k]) \leq 0, \\ 1 & \text{otherwise}, \end{cases}$$

and the *linear migration* policy

$$\mu_{pq}[k] = \begin{cases} 0 & \text{if } l_p(\boldsymbol{x}[k]) - l_q(\boldsymbol{x}[k]) \leq 0, \\ \frac{l_p(x_p[k]) - l_q(x_q[k])}{l_{max}} & \text{otherwise}, \end{cases}$$

where l_{max} is the maximum latency value. By using these migration policies, it can be shown that convergence cannot be guaranteed, however small the sampling time τ is chosen (see, e.g., [25]).

The proposed migration policy is then a modified *better response* policy, defined as:

$$\mu_{pq}[k] = \begin{cases} 0 & \text{if } l_p(\boldsymbol{x}[k]) - l_q(\boldsymbol{x}[k]) \leq \alpha\delta[k], \\ 1 & \text{otherwise}, \end{cases} \tag{7}$$

where $\alpha \in (0,1)$ is a tuning parameter which affects the algorithm convergence velocity, as examined in the following Section, and $\delta[k]$ is defined as the difference between the maximum measured latency value of the loaded paths and the minimum measured latency value, at time k, i.e.,

$$\delta[k] \coloneqq \max_{p \in \mathcal{P} \mid x_p[k] > 0} l_p(\boldsymbol{x}[k]) - \min_{q \in \mathcal{P}} l_q(\boldsymbol{x}[k]). \tag{8}$$

The proposed control law is expressed by equations (4), (5) and (7), with the following control gain upper bound:

$$\sigma_{pq}[k] \leq \frac{\alpha\delta[k]}{\tau\lambda\beta_{max} d(|\mathcal{P}| + \eta)}, \tag{9}$$

The system dynamics (3), under the control law (4), (5), (7), (8), and (9) – hereafter referred to as *Discrete-Time Selfish Routing* (DTSR) dynamics – resembles the algorithm proposed in [24]; besides the different expression (9) for $\sigma_{pq}[k]$, the main difference is the tolerance $\alpha\delta[k]$ introduced in equation (7).

In particular, the DTSR dynamics yields, in vector form, the closed-loop nonlinear system dynamics

$$\boldsymbol{x}[k+1] = f(\boldsymbol{x}[k]), \quad \boldsymbol{x}(0) \in \mathcal{X}, \tag{10}$$

with respect to which we conduct the Lyapunov stability analysis reported in Section IV.B. Such a map from a population game to a set of difference equations is referred to in the literature as *deterministic evolutionary dynamics* [8].

B. Convergence Proof

Lemma 1. Under Assumption 1 on the DTSR dynamics, with initial conditions (6), the latency variation of an edge e in time steps is bounded according to the following relation:

$$-\frac{|\mathcal{P}|}{\lambda(|\mathcal{P}| + \eta)} \alpha\delta[k] \leq l_e(x_e[k+1]) - l_e(x_e[k]) \leq \frac{\eta}{\lambda(|\mathcal{P}| + \eta)} \alpha\delta[k]. \tag{11}$$

■

Proof. In the worst case, at time k, no paths migrate their own population to a path p such that $e \in p$:

$$l_e(x_e[k+1]) = l_e\big(x_e[k] + \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{m\in\mathcal{P}\setminus\mathcal{P}_e}\big(r_{mp}[k] - r_{pm}[k]\big)\big)$$
$$\geq l_e\big(x_e[k] - \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{m\in\mathcal{P}\setminus\mathcal{P}_e}\big(r_{pm}[k]\big)\big). \quad (12)$$

Since β_e is the Lipschitz constant of the function l_e, it follows that

$$l_e(x_e[k+1]) \geq l_e(x_e[k]) - \beta_e \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{m\in\mathcal{P}\setminus\mathcal{P}_e} r_{pm}[k]. \quad (13)$$

Considering equations (4), (7) and (9), the last term of equation (13) is written as

$$\beta_e \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{m\in\mathcal{P}\setminus\mathcal{P}_e} r_{pm}[k] =$$
$$= \beta_e \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{m\in\mathcal{P}\setminus\mathcal{P}_e} \sigma_{pm}[k]\mu_{pm}[k]x_p[k]$$
$$\leq \beta_e \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{m\in\mathcal{P}\setminus\mathcal{P}_e} \sigma_{pm}[k]x_p[k]$$
$$\leq \frac{\alpha\delta[k]}{\lambda d(|\mathcal{P}|+\eta)} \textstyle\sum_{p\in\mathcal{P}_e} x_p[k] \sum_{m\in\mathcal{P}\setminus\mathcal{P}_e} 1, \quad (14)$$

where the last inequality holds since $\beta_e \leq \beta_{max}$. Since there are at most $|\mathcal{P} \setminus \mathcal{P}_e| \leq |\mathcal{P}|$ terms in the last summation of equation (14) and since $\sum_{p\in\mathcal{P}_e} x_p[k] \leq d$, it holds that

$$\beta_e \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{m\in\mathcal{P}\setminus\mathcal{P}_e} r_{pm}[k] \leq \frac{|\mathcal{P}|}{\lambda(|\mathcal{P}|+\eta)}\alpha\delta[k]. \quad (15)$$

Similarly, in the worst case, at time k, every path q such that $e \in q$ does not migrate its population to any other paths:

$$l_e(x_e[k+1]) \leq l_e(x_e[k]) + \beta_e \tau \textstyle\sum_{q\in\mathcal{P}_e}\sum_{n\in\mathcal{P}\setminus\mathcal{P}_e} r_{nq}[k]. \quad (16)$$

Also, given definition (9), since $\beta_{max} \geq \beta_e, \forall e \in \mathcal{E}$, and $\sum_{n\in\mathcal{P}\setminus\mathcal{P}_e} x_n[k] \leq d$, it holds that:

$$\beta_e \tau \textstyle\sum_{q\in\mathcal{P}_e}\sum_{n\in\mathcal{P}\setminus\mathcal{P}_e} r_{nq}[k] \leq \frac{\alpha\delta[k]}{\lambda(|\mathcal{P}|+\eta)} \textstyle\sum_{q\in\mathcal{P}_e} \mu_{nq}[k]. \quad (17)$$

Since $\eta \geq |\mathcal{P}_e|, \forall e \in \mathcal{E}$, and $\mu_{pq}[k] \leq 1, \forall p, q \in \mathcal{P}$, it holds that

$$\beta_e \tau \textstyle\sum_{q\in\mathcal{P}_e}\sum_{n\in\mathcal{P}\setminus\mathcal{P}_e} r_{nq}[k] \leq \frac{\eta}{\lambda(|\mathcal{P}|+\eta)}\alpha\delta[k]. \quad (18)$$

■

It is shown below that the state space $\mathcal{X}$ is a positively invariant set.

Lemma 2. Under Assumption 1, $\mathcal{X}$ is a positively invariant set for the DTSR dynamics, with the initial conditions specified in (6).

Proof.

It is shown in the following that, since the initial flow vector of the considered system dynamics, $\boldsymbol{x}[0]$, lies in $\mathcal{X}$, the flow vector $\boldsymbol{x}[k]$ lies in $\mathcal{X}$ too, i.e., that a) $\sum_{p\in\mathcal{P}} x_p[k] = d$ and b) $x_e[k] \geq 0, \forall e \in \mathcal{E}, \forall k \geq 0$.

a) It follows from equation (3) that

$$\textstyle\sum_{p\in\mathcal{P}}\big(x_p[k+1] - x_p[k]\big) = \tau \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{P}}\big(r_{qp}[k] - r_{pq}[k]\big)$$
$$= \tau\big(\textstyle\sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{P}} r_{qp}[k] - \sum_{q\in\mathcal{P}}\sum_{p\in\mathcal{P}} r_{qp}[k]\big) = 0, \quad (19)$$

and, therefore, that $\sum_{p\in\mathcal{P}} x_p[k] = \sum_{p\in\mathcal{P}} x_p[0] = d, \forall k \geq 0$.

b) By induction, since $x_e[0] \geq 0, \forall e \in \mathcal{E}$, it is proven below that $x_e[k] \geq 0, \forall e \in \mathcal{E}, \forall k \geq 0$. Assuming that $x_e[k] \geq 0, \forall e \in \mathcal{E}$, for a given k, it is sufficient to prove that

$$x_e[k+1] = x_e[k] + \tau \textstyle\sum_{\substack{q\in\mathcal{P}|e\notin q \\ p\in\mathcal{P}|e\in p}}\big(r_{qp}[k] - r_{pq}[k]\big) \geq 0, \forall e \in \mathcal{E}. \quad (20)$$

Since $x_e[k] \geq 0$, from equation (4) it follows that $r_{pq}[k] \geq 0$. Thus, the following inequality holds (in the worst case, no paths migrate part of their population to a path p which includes e):

$$x_e[k+1] \geq x_e[k] - \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{q\in\mathcal{P}\setminus\mathcal{P}_e} r_{pq}[k], \forall e \in \mathcal{E}. \quad (21)$$

A sufficient condition for inequality (20) to hold is then

$$x_e[k] - \tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{q\in\mathcal{P}\setminus\mathcal{P}_e} r_{pq}[k] \geq 0, \forall e \in \mathcal{E} \quad (22)$$

Recalling equations (4) and (8) and the definition of $x_e[k]$, equation (22) can be written as

$$\textstyle\sum_{p\in\mathcal{P}_e} x_p[k] - \tau \sum_{p\in\mathcal{P}_e}\sum_{q\in\mathcal{P}\setminus\mathcal{P}_e} r_{pq}[k] = \sum_{p\in\mathcal{P}_e} x_p[k] -$$
$$\tau \textstyle\sum_{p\in\mathcal{P}_e}\sum_{q\in\mathcal{P}\setminus\mathcal{P}_e} x_p[k]\sigma_{pq}[k]\mu_{pq}[k]$$
$$= \textstyle\sum_{p\in\mathcal{P}_e}\Big(x_p[k]\big(1 - \tau \sum_{q\in\mathcal{P}\setminus\mathcal{P}_e} \sigma_{pq}[k]\mu_{pq}[k]\big)\Big) \geq 0, \forall p \in \mathcal{P}. \quad (23)$$

The inequality (23) holds if each term of the summation is positive, which, given equation (9), is written as:

$$x_p[k]\big(1 - \tau \textstyle\sum_{q\in\mathcal{P}\setminus\mathcal{P}_e} \sigma_{pq}[k]\mu_{pq}[k]\big)$$
$$\geq x_p[k]\Big(1 - \frac{\alpha\delta[k]}{\lambda\beta_{max}d(|\mathcal{P}|+\eta)} \textstyle\sum_{q\in\mathcal{P}\setminus\mathcal{P}_e} \mu_{pq}[k]\Big)$$
$$> x_p[k]\Big(1 - \frac{\alpha\delta[k]}{\lambda\beta_{max}d}\Big) \geq 0, \forall p \in \mathcal{P}, \quad (24)$$

where the second inequality holds since, considering that $\mu_{qq}[k] = 0, \forall k \geq 0$, the inner summation has at most $(|\mathcal{P}| - 1)$ terms equal to 1. If $x_p[k] = 0$, the inequality (24) is verified. Instead, if $x_p[k] > 0$, equation (24) holds if

$$\frac{\alpha\delta[k]}{\lambda\beta_{max}d} \leq 1. \quad (25)$$

Given that $0 < \alpha < 1$ and given definition (8), it follows that

$$\frac{\alpha\delta[k]}{\lambda\beta_{max}d} < \frac{\max_{p\in\mathcal{P}} l_p(x[k])}{\lambda\beta_{max}d} < 1, \quad (26)$$

where the last inequality holds since, by the definitions of the β_e's, of β_{max} and of λ, it holds that $l_p(\boldsymbol{x}) = \sum_{e\in p} l_e(x_e) \leq \sum_{e\in p} \beta_e d \leq \lambda\beta_{max}d, \forall p \in \mathcal{P}$. ■

We are now in a position to prove the following theorem.

Theorem 1. Under Assumption 1, $\boldsymbol{x}_{\mathcal{W}}$ is a globally asymptotically stable equilibrium point for the DTSR dynamics, with initial conditions (6) and with total traffic demand $d > 0$. ■

Proof. The following proof is based on Lyapunov's direct method. Let $\mathcal{L}(\boldsymbol{x}) \coloneqq \Phi(\boldsymbol{x}) - \Phi_{min}$ be the candidate Lyapunov function, where $\Phi(\boldsymbol{x})$ is the potential (2) and Φ_{min} is its minimum value, which is unique thanks to Assumption 1.

Let $\Delta\mathcal{L}(\boldsymbol{x}[k])$ denote the difference of the Lyapunov function $\mathcal{L}(\boldsymbol{x})$ along the solutions of the DTSR system:

$$\Delta\mathcal{L}(\boldsymbol{x}[k+1]) \coloneqq \mathcal{L}(\boldsymbol{x}[k+1]) - \mathcal{L}(\boldsymbol{x}[k])$$
$$= \textstyle\sum_{e\in\mathcal{E}} \int_{x_e[k]}^{x_e[k+1]} l_e(\xi)d\xi$$
$$\leq \textstyle\sum_{e\in\mathcal{E}} (x_e[k+1] - x_e[k]) l_e(x_e[k+1]), \quad (27)$$

where the inequality holds from geometric considerations, as the latency functions are strictly increasing (see the Appendix).

Considering that

$$\sum_{e\in\mathcal{E}} x_e[k] l_e(x_e[k]) = \sum_{e\in\mathcal{E}} \sum_{p\in\mathcal{P}_e} x_p[k]\, l_e(x_e[k]) = \sum_{p\in\mathcal{P}} \sum_{e\in p} x_p[k]\, l_e(x_e[k]) = \sum_{p\in\mathcal{P}} x_p[k] \sum_{e\in p} l_e(x_e[k]) = \sum_{p\in\mathcal{P}} x_p[k] l_p(\boldsymbol{x}[k]), \quad (28)$$

from equations (27) and (28), it follows that

$$\begin{aligned}\Delta\mathcal{L}(\boldsymbol{x}[k+1]) &\le \sum_{p\in\mathcal{P}} \big(x_p[k+1] - x_p[k]\big) l_p(\boldsymbol{x}[k+1]) \\ &= \sum_{p\in\mathcal{P}} \tau\big(\sum_{q\in\mathcal{P}} r_{qp}[k] - \sum_{q\in\mathcal{P}} r_{pq}[k]\big) l_p(\boldsymbol{x}[k+1]) \\ &= \tau \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{qp}[k]\, l_p(\boldsymbol{x}[k+1]) - \tau \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k]\, l_p(\boldsymbol{x}[k+1]) \\ &= \tau \sum_{q\in\mathcal{P}} \sum_{p\in\mathcal{P}} r_{pq}[k]\, l_q(\boldsymbol{x}[k+1]) - \tau \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] l_p(\boldsymbol{x}[k+1]) \\ &= \tau \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] \Big(l_q(\boldsymbol{x}[k+1]) - l_p(\boldsymbol{x}[k+1])\Big). \quad (29)\end{aligned}$$

We now prove that the terms $r_{pq}[k]\Big(l_q(\boldsymbol{x}[k+1]) - l_p(\boldsymbol{x}[k+1])\Big)$ of the last summation are either negative or null, for any $p, q \in \mathcal{P}$.

a) If $r_{pq}[k] = 0$, the term is null.

b) It is shown below that, if $r_{pq}[k] > 0$, it holds that $l_p(\boldsymbol{x}[k+1]) - l_q(\boldsymbol{x}[k+1]) > 0$, i.e., the considered term in (29) is negative.

By Lemma 1, it holds that

$$\begin{aligned}&l_p(\boldsymbol{x}[k+1]) - l_q(\boldsymbol{x}[k+1]) \\ &\ge \Big(l_p(\boldsymbol{x}[k]) - \sum_{e\in p} \frac{|\mathcal{P}|}{\lambda(|\mathcal{P}|+\eta)} \alpha\delta[k]\Big) - \Big(l_q(\boldsymbol{x}[k]) + \sum_{e\in q} \frac{\eta}{\lambda(|\mathcal{P}|+\eta)} \alpha\delta[k]\Big) \\ &\ge \Big(l_p(\boldsymbol{x}[k]) - \frac{|\mathcal{P}|}{|\mathcal{P}|+\eta} \alpha\delta[k]\Big) - \Big(l_q(\boldsymbol{x}[k]) + \frac{\eta}{|\mathcal{P}|+\eta} \alpha\delta[k]\Big) \\ &= l_p(\boldsymbol{x}[k]) - l_q(\boldsymbol{x}[k]) - \alpha\delta[k], \quad (30)\end{aligned}$$

where the second inequality follows from the fact that $|p| \le \lambda, \forall p \in \mathcal{P}$. Since we are considering the case $r_{pq}[k] > 0$, it follows from equation (7) that

$$l_p(\boldsymbol{x}[k]) - l_q(\boldsymbol{x}[k]) > \alpha\delta[k], \quad (31)$$

which, from equation (30), yields that $l_p(\boldsymbol{x}[k+1]) - l_q(\boldsymbol{x}[k+1]) > 0$.

Now we can prove the asymptotic convergence to $\boldsymbol{x}_{\mathcal{W}}$ by showing that $\Delta\mathcal{L}(\boldsymbol{x}[k+1]) = 0$ if and only if $\boldsymbol{x}[k] = \boldsymbol{x}_{\mathcal{W}}$.

If $\boldsymbol{x}[k] = \boldsymbol{x}_{\mathcal{W}}$, for any couple $p, q \in \mathcal{P}$, it holds either that $l_p(\boldsymbol{x}[k]) \le l_q(\boldsymbol{x}[k])$ with $x_p[k] > 0$ or that $x_p[k] = 0$. In the latter case, it follows from equation (4) that $r_{pq}[k] = 0$. In the former case, equation (7) yields that $\mu_{pq}[k] = 0$ and, therefore, $r_{pq}[k] = 0$. From the discussion above, it follows that $\Delta\mathcal{L}(\boldsymbol{x}[k+1]) = 0$.

If $\boldsymbol{x}[k] \ne \boldsymbol{x}_{\mathcal{W}}$**,** there exist one or more pairs of paths $p, q \in \mathcal{P}$, such that $x_p[k] > 0$ and $l_p(\boldsymbol{x}[k]) - l_q(\boldsymbol{x}[k]) \ge 0$. Let $\bar{p}$ and $\bar{q}$ be the paths such that $\bar{p} = \underset{p\in\mathcal{P}\,|\,x_p[k]>0}{\operatorname{argmax}}\, l_p(\boldsymbol{x}[k])$ and $\bar{q} = \underset{q\in\mathcal{P}}{\operatorname{argmin}}\, l_q(\boldsymbol{x}[k])$, respectively, i.e., by definition (8), such that $l_{\bar{p}}(\boldsymbol{x}[k]) - l_{\bar{q}}(\boldsymbol{x}[k]) = \delta[k]$. Given that $0 < \alpha < 1$, it holds that $l_{\bar{p}}(\boldsymbol{x}[k]) - l_{\bar{q}}(\boldsymbol{x}[k]) > \alpha\delta[k]$ and, from equations (4) and (7), that $r_{\bar{p}\bar{q}}[k] > 0$. From the discussion above, it follows that $\Delta\mathcal{L}(\boldsymbol{x}[k+1]) < 0$. ■

Remark 1. In the control law (4), the σ_{pq}'s are interpreted as the control gains. Lemma 1 and Theorem 1 show that equation (9) sets dynamic upper bounds on such control gains, with the twofold objective of keeping the dynamics feasible and of driving the system trajectories towards $\boldsymbol{x}_{\mathcal{W}}$. ■

Remark 2. Concerning the migration policy in equation (7), the key elements for the convergence of the discrete-time case are the facts (i) that, at any time step k, equation (7) sets a minimum latency separation $\alpha\delta[k]$ for the migration between two paths, and (ii) that such latency separation vanishes with k, as the latency values converge. ■

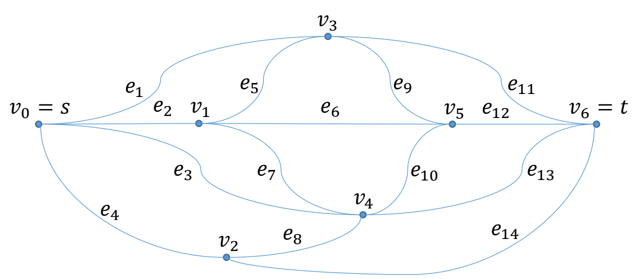
Remark 3. As regards the convergence velocity of the considered closed-loop dynamics (10), the value chosen for α sets a trade-off between the gain value – the upper bound (9) on the σ_{pq}'s is proportional to α – and the number of flows selected for migrations – according to equation (7), at a given time step k, the μ_{pq}'s are positive only for the path pairs whose latencies differ by a quantity that is larger than $\alpha\delta[k]$ – see also the example simulation runs in Section V. ■

V. Numerical Simulations

In this Section, the proposed algorithm is evaluated via numerical simulations performed with MATLAB®. The example scenario has $|\mathcal{E}| = 14$ edges and $|\mathcal{P}| = 12$ paths, as shown in Fig. 1, which support the total traffic $d = 1$. The figure shows that $\lambda = 4$ and $\eta = 6$. The following convex latency functions[2] were selected:

$$l_e(x_e) = e^{-\beta_e x_e/d} - 1, \quad (32)$$

with $\beta_e = 0.2$, for $e \le 5$, $\beta_e = 0.3$ for $6 \le e \le 9$, $\beta_e = 0.1$ for $e \ge 10$. Equation (32) defines three groups of edges, each one characterized by the same latency function.



$p_1 = \{e_1, e_{11}\}$	$p_5 = \{e_2, e_6, e_{12}\}$	$p_9 = \{e_3, e_{13}\}$
$p_2 = \{e_1, e_9, e_{12}\}$	$p_6 = \{e_2, e_7, e_{10}, e_{12}\}$	$p_{10} = \{e_4, e_8, e_{13}\}$
$p_3 = \{e_2, e_5, e_{11}\}$	$p_7 = \{e_2, e_7, e_{13}\}$	$p_{11} = \{e_4, e_8, e_{10}, e_{12}\}$
$p_4 = \{e_2, e_5, e_9, e_{12}\}$	$p_8 = \{e_3, e_{10}, e_{12}\}$	$p_{12} = \{e_4, e_{14}\}$

Figure 1. Example network and list of paths from the source node s to the destination node t (v_i: vertex i; e_i: edge i; p_i: path i).

The first set of simulation runs addresses the tuning of the parameter α of the control action, whereas the second one shows some latency and population dynamics examples as well as a qualitative comparison between the proposed algorithm and the one in [27].

[2] Analogous results were achieved by simulating the algorithm with monotonically increasing but non-convex functions, e.g., the logarithmic one; the case of convex latency functions is presented here since it is meaningful for routing problems.

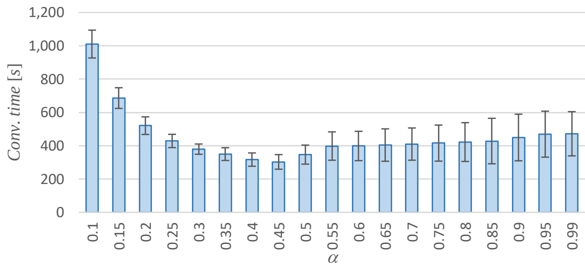


Figure 2. Convergence time (mean and std. deviation) vs. α

To evaluate the effect of the parameter α on the convergence time, simulation runs starting from random initial states were simulated for values of α ranging from 0.1 to 0.99. The convergence time was defined as the time instant k_c such that the maximum latency mismatch $\delta[k_c]$ lies within a tolerance $\varepsilon = 10^{-3}$.

Fig. 2 shows the obtained convergence times, averaged over 20 simulation runs, highlighting that the convergence time

- is larger for small values of α and then decreases with α until $\alpha = 0.45$, due to the fact that the σ_{pq}'s, i.e., the control gains, increase with α;
- increases for values of α from $\alpha = 0.45$ onwards, since fewer paths are selected for migrations (see the migration policy (7)); for instance, for values of α sufficiently close to 1, only the paths with the largest and smallest latency values can exchange their population.

The second simulation set presents examples of simulation runs under the proposed algorithm, for different values of α, and under the algorithm in [27]. The latter is a round-based algorithm: at every time step k, each path $p \in \mathcal{P}$ is activated with probability γ. Each activated path decides to migrate to a given path $q \in \mathcal{P}$ based on the rule:

$$\Pr\{q \text{ is selected}\} = \begin{cases} \frac{1}{|\mathcal{P}|} & \text{with probability } \beta_F \\ \frac{x_q[k]}{d} & \text{with probability } (1-\beta_F), \end{cases} \tag{33}$$

where $\beta_F \in (0,1)$ is the probability with which a random path is selected for the migration (*exploration*) instead of selecting the path proportionally to its population (*exploitation*). The migration from an activated path to the selected one is still given by equation (4), with migration rate $\sigma_p = \frac{1}{|\mathcal{P}|}$ and migration probability $\mu_{pq}[k] = \max\left\{\frac{l_p[k]-l_q[k]}{\bar{\eta}(l_p[k]+\zeta)}, 0\right\}$, where $\bar{\eta}$ is the maximum value of the elasticity of the latency functions (defined in [27] as $\eta(\xi) \coloneqq \frac{x \cdot dl(\xi)/d\xi}{l(\xi)}$) and ζ is a positive value needed to avoid a division by 0 if $l_p[k] = 0$. The parameters, tuned according to [27], were set as follows: $\sigma_{pq} = \frac{1}{12}$, $\zeta = 0.01$, $\gamma = \frac{1}{32}$, $\beta_F = 0.002$, $\bar{\eta} = 2$.

Fig. 3 and Fig. 4 show the dynamics of the latency and population values obtained with the proposed algorithm, with $\alpha = 0.45$, and with the algorithm in [27], respectively, starting from the same initial population. The figures highlight that, while the convergence velocities appear similar, the dynamics of the proposed algorithm is much smoother, since the algorithm in [27] relies on migration probabilities. For the same reason, even in the long run, the algorithm in [27] produces oscillations around the value at the equilibrium (in fact, it achieves an approximate equilibrium, see [27] for details).

Fig. 5 shows the algorithm behaviour at values of α close to 1, highlighting that, at each time step, only the paths with the largest and smallest latency values exchange their populations.

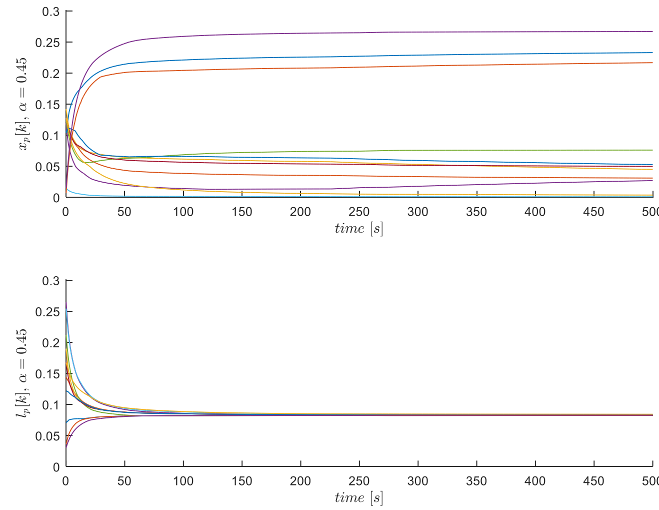


Figure 3. Population dynamics (higher plot) and latency dynamics (lower plot) with $\alpha = 0.45$.

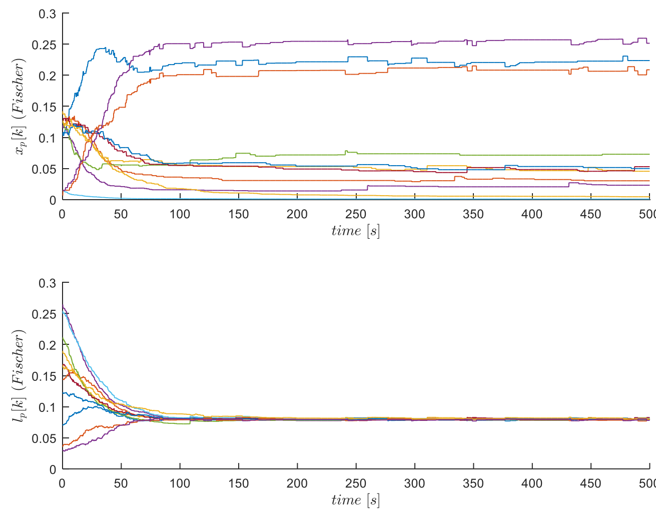


Figure 4. Population dynamics (higher plot) and latency dynamics (lower plot) with the algorithm in [27].

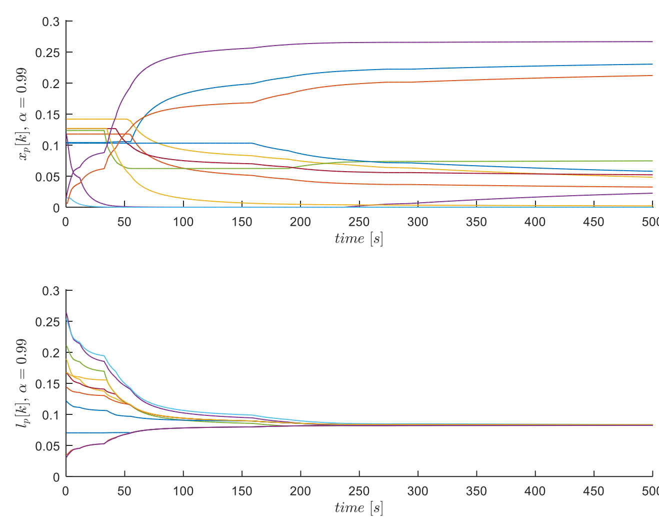


Figure 5. Population dynamics (higher plot) and latency dynamics (lower plot) with $\alpha = 0.99$.

VI. Conclusions and Future Work

This paper proposes a distributed non-cooperative selfish routing algorithm, which adopts the problem formulation proposed in [25] and introduces the design of a discrete-time algorithm. The proposed algorithm is proved to asymptotically converge to the exact Wardrop equilibrium, in contrast with the existing discrete-time algorithms, which converge to approximate Wardrop equilibria. The convergence proof is given by relying on Lyapunov's second method.

Future work is aimed (i) at extending the proposed algorithm to the multi-commodity scenario, (ii) at analysing the effects of time-delays and of communication constraints and losses and (iii) at implementing the algorithm in real applications, such as load balancing in smart-grids and in cloud networks, where the load and the latency functions may vary with time.

Appendix

If $x_e[k+1] > x_e[k]$, the upper plot of Fig. 6 shows that the quantity $(x_e[k+1] - x_e[k]) l_e(x_e[k+1])$, equal to the area of the rectangle with bold lines, is larger than the integral $\int_{x_e[k]}^{x_e[k+1]} l_e(\xi) d\xi$, equal to the grey area. If $x_e[k+1] < x_e[k]$, the lower plot shows that $(x_e[k] - x_e[k+1]) l_e(x_e[k+1])$, equal to the area of the rectangle with bold lines, is smaller than the integral $\int_{x_e[k+1]}^{x_e[k]} l_e(\xi) d\xi$, equal to the grey area.

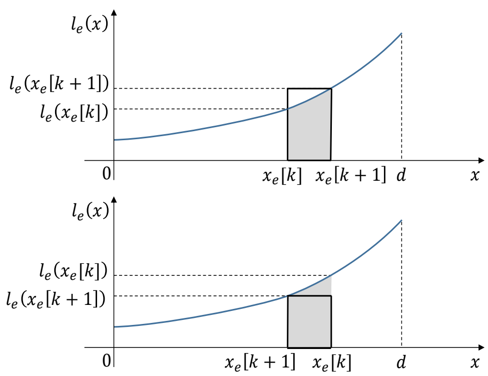


Figure 6. Geometrical considerations on the latency functions.

References

[1] J. G. Wardrop, "Some Theoretical Aspects of Road Traffic Research," *ICE Proc. Eng. Div.*, vol. 1, no. 3, pp. 325–362, Jan. 1952.

[2] J. R. Correa and N.E. Stier-Moses, "Wardrop Equilibria," in *Wiley Encyclopedia of Operations Research and Management Science*, J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh, and J. C. Smith, Eds. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011.

[3] N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, 2007.

[4] E. Anshelevich, D. Kempe, and J. Kleinberg, "Stability of Load Balancing Algorithms in Dynamic Adversarial Systems," *SIAM Journal on Computing*, vol. 37, no. 5, pp. 1656-1673, 2008.

[5] G. Como, E. Lovisari, K. Savla, "Throughput Optimality and Overload Behavior of Dynamical Flow Networks Under Monotone Distributed Routing," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 1, pp. 57-67, 2015.

[6] A. Haurie and P. Marcotte, "On the relationship between Nash-Cournot and Wardrop equilibria," *Networks*, vol. 15, no.3, 1985.

[7] V.S. Borkar, "Cooperative dynamics and Wardrop equilibria," *Systems & Control Letters*, vol. 58, no. 2, pp. 91-93, 2009.

[8] W. Sandholm, *Population games and evolutionary dynamics*, MIT press, 2010.

[9] M. Zhua and E. Frazzoli, "Distributed robust adaptive equilibrium computation for generalized convex games," *Automatica*, vol. 63, pp. 82-91, 2016.

[10] D. Bauso, X. Zhang, A. Papachristodoulou, "Density Flow in Dynamical Networks via Mean-Field Games, *IEEE Transactions on Automatic Control*, 62 (3), art. no. 7499816, pp. 1342-1355, 2017.

[11] T. Tatarenko, *Game-Theoretic Learning and Distributed Optimization in Memoryless Multi-Agent Systems*, Springer, 2017.

[12] S.-W. Chiou, "Optimization of a nonlinear area traffic control system with elastic demand," *Automatica*, vol. 46, no. 10, pp. 1626–1635, Oct. 2010.

[13] G. Como, K. Savla, D. Acemoglu, M.A. Dahleh, E. Frazzoli, "Robust distributed routing in dynamical networks – Part I: Locally responsive policies and weak resilience," *IEEE Transactions on Automatic Control*, 58 (2), art. no. 6247471, pp. 317-332, 2013.

[14] S. Fischer, N. Kammenhuber, and A. Feldmann, "REPLEX: dynamic traffic engineering based on Wardrop routing policies," in *Proceedings of the 2nd Conference on Future Networking Technologies (CoNext)*, 2006, pp. 6–17.

[15] E. Altman, T. Boulogne, R. El-Azouzi, T. Jimenez, and L. Wynter, "A survey on networking games in telecommunications," *Computers & Operations Research*, vol. 33, no. 2, pp. 286-311, 2006.

[16] D. Grosu and A. T. Chronopoulos, "Algorithmic Mechanism Design for Load Balancing in Distributed Systems," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 34, no. 1, pp. 77–84, Feb. 2004.

[17] J. Anselmi, U. Ayesta, and A. Wierman, "Competition yields efficiency in load balancing games," *Perform. Eval.*, vol. 68, no. 11, pp. 986–1001, Nov. 2011.

[18] G. Oddi, A. Pietrabissa, F. Delli Priscoli, and V. Suraci, "A decentralized load balancing algorithm for heterogeneous wireless access networks," in *World Telecommunications Congress 2014 (WTC2014)*, 2014, pp. 1-6.

[19] D. Barth, O. Bournez, O. Boussaton, and J. Cohen, "Distributed learning of Wardrop equilibria," in *Lecture Notes in Computer Science, vol. 5204*, 2008, pp. 19–32.

[20] M. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the economics of transportation*. New Haven: Yale University Press, 1959.

[21] F. Benmansour, G. Carlier, G. Peyré, F. Santambrogio, "Numerical approximation of continuous traffic congestion equilibria," *Networks and Heterogeneous Media*, 4 (3), pp. 605-623, 2009.

[22] Y.O. Lee, B. Rengarajany, A.L. Narasimha Reddy, "Reaching approximate Wardrop equilibrium at reduced costs of link state updates," *3rd International Conference on Communication Systems and Networks, COMSNETS 2011*, art. no. 5716481, 2011.

[23] G. Como, K. Savla, D. Acemoglu, M.A. Dahleh, E. Frazzoli, "Robust distributed routing in dynamical networks – Part II: Strong resilience, equilibrium selection and cascaded failures," *IEEE Transactions on Automatic Control*, 58 (2), art. no. 6248170, pp. 333-348, 2013.

[24] S. Fischer, L. Olbrich, and B. Vöcking, "Approximating Wardrop equilibria with finitely many agents," *Distrib. Comput.*, vol. 21, no. 2, pp. 129–139, Mar. 2008.

[25] S. Fischer and B. Vöcking, "Adaptive routing with stale information," *Theor. Comput. Sci.*, vol. 410, no. 36, pp. 3357–3371, Aug. 2009.

[26] V. S. Borkar and P. R. Kumar, "Dynamic Cesaro-Wardrop equilibration in networks," *IEEE Trans. Automat. Contr.*, vol. 48, no. 3, pp. 382–396, Mar. 2003.

[27] S. Fischer, H. Räcke, and B. Vöcking, "Fast Convergence to Wardrop Equilibria by Adaptive Sampling Methods," *SIAM J. Comput.*, vol. 39, no. 8, pp. 3700–3735, Jan. 2010.

[28] A. Blum, E. Even-Dar, and K. Ligett, "Routing without regret: On convergence to Nash equilibria of regret-minimizing algorithms in routing games," in *Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing*, 2006.

[29] W. Krichene, B. Drighes, and A. Bayen, "On the convergence of no-regret learning in selfish routing," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no. 2, pp. 163-171, 2014.

[30] H. Ackermann, S. Fischer, M. Hoefer, and M. Schöngens, "Distributed algorithms for QoS load balancing," *Distrib. Comput.*, vol. 23, no. 5–6, pp. 321–330, Dec. 2010.

[31] F. Cimorelli, F. Delli Priscoli, A. Pietrabissa, L. Ricciardi Celsi, V. Suraci, and L. Zuccaro, "A Distributed Load Balancing Algorithm for the Control Plane in Software Defined Networking," in *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED 2016)*, pp. 1033-1040, June 21-24, 2016, Athens, Greece, DOI: 10.1109/MED.2016.7535946.

[32] A. Pietrabissa and V. Suraci, "Wardrop equilibrium on time-varying graphs," *Automatica*, vol. 84, pp. 159-165, 2017, doi: 10.1016/j.automatica.2017.07.021.