Solving N-player dynamic routing games with congestion: a mean field approach

Theophile Cabannes[1,2,5], Paul Muller[4], Mathieu Laurière[3], Julien Perolat[4], Raphael Marinier[3], Sertan Girgin[3], Sarah Perrin[6] Olivier Pietquin[3], Alexandre M. Bayen[1,2], Eric Goubault[5], Romuald Elie[4]
[1]University of California, Berkeley, [2]Google Research, [3]Google Research, Brain team, [4]DeepMind
[5]Laboratoire d'Informatique de l'Ecole Polytechnique, Ecole Polytechnique
[6]Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL

Abstract— **The recent emergence of navigational tools has enabled new types of congestion-aware routing control like fastest path routing, eco-routing, or dynamic road pricing. Using the fundamental diagram of traffic flows – used in many macroscopic and mesoscopic traffic models – the article introduces a new N-player dynamic routing game with explicit congestion dynamics. The model is well-posed and can reproduce heterogeneous departure times and congestion spill back phenomena. However, as Nash equilibrium computations are PPAD-complete, solving the game becomes intractable for large but realistic numbers of vehicles N. Therefore, the corresponding mean field game is also introduced. Experiments were performed on several classical benchmark networks of the traffic community: the Pigou, Braess, and Sioux Falls networks with heterogeneous origin, destination and departure time tuples. The Pigou and the Braess examples reveal that the mean field approximation is generally very accurate and computationally efficient as soon as the number of vehicles exceeds a few dozen. On the Sioux Falls network (76 links, 100 time steps), this approach enables learning traffic dynamics with more than 14,000 vehicles.**

I. Introduction

A. Motivations

In 2019, the Texas A&M Transportation Institute estimated that the U.S. loses $166 billion per year due to the impact of congestion on fuel usage and productivity loss [1]. The average auto commuter spends 54 hours in congestion and wastes 21 gallons of fuel every year due to congestion at a cost of $1,010 in wasted time and fuel. With the emergence of navigational applications, the traffic patterns have evolved due to congestion-aware routing behaviors [2], [3]. Being able to model traffic and especially routing choice adequately would enable traffic control to leverage the new routing behaviors in order to improve the network efficiency.

However, solving realistic routing choice problems requires the consideration of large scale multi-agent systems, where the number of vehicles making strategic decisions is not tractable within classical algorithms [4]. This work focuses on finding a scalable approach to model the routing behavior of each vehicle in a dynamic road traffic environment as a large multi-agent dynamic system.

The current article summarizes the work that has been explain with more context in [3, Chapter 6].

B. Contributions and outline

The first main contribution of this work (see section II) is to introduce a novel dynamic mesoscopic traffic model viewed as a dynamic routing game with explicit congestion dynamics, i.e., congestion effects directly in the state evolution. We first propose a finite-player game which can easily be interpreted in terms of vehicles, and then derive the corresponding mean-field game (MFG). It is proved that Nash equilibria exist for both games. Furthermore, theoretical arguments supporting the MFG approximation are discussed. The reader can refer to [5] to understand how MFGs solve the curse of dimensionality in large population systems.

The second main contribution (see section III) is to demonstrate numerically that the MFG provides an efficient way to approximately solve the finite-player routing game with very large population: the MFG is much less costly to solve and yet provides a very good approximate Nash equilibrium policy. This is illustrated on small networks for which baselines are available and on the Sioux Falls network – a realistic network with 76 links and 100 time steps with 14,000 vehicles. Although this network is often used as a benchmark in the literature, to the best of our knowledge, it is the first time that a method is able to solve a dynamic routing game with a very large number of vehicles on this network.

C. Related work

Several works study mean-field routing games. First, continuous time models have been studied. The existence and uniqueness of the Nash equilibrium of a MFG with congestion on a graph has been shown in [6], where the state space is the set of nodes. Models in which the state space is given by the edges have been analyzed e.g. in [7], which proved existence and uniqueness for a forward -backward system of equations with suitable conditions at the vertices of the network. In [8], the authors analyzed an MFG model for traffic flow on networks by using an extended state space that includes the distribution of players on the network and they studied Wardrop equilibria. In existing discrete time MFG models for routing, the players move one edge per time step and pay a cost that increases with the proportion of players on the same edge. In [9], the authors analyzed an MFG model and studied the impact of adding or removing edges on the equilibrium traffic flow. Their work provides a discrete time resolution of a mean field routing game with on an 11 link network with 6 time steps. In [10], the authors proposed an MFG model that reduces to a linearly solvable Markov Decision Process and showed connections with Fictitious Play [11] in some cases.

The fact that existing models take congestion into account only through the cost functions leads to paradox such as an ambiguity about the definition of travel time: the graph traversal time and the player cost can differ. Such issues make these models hardly applicable for traffic engineering. Also, the main motivation for using an MFG-based routing method is to obtain an efficient equilibrium policy in the finite-player routing game, which has not been checked in existing works.

II. DYNAMIC N-PLAYER AND MEAN FIELD ROUTING GAMES

This section introduces the dynamic routing game and the corresponding MFG. This dynamic routing game models the evolution of N vehicles on a road network. The vehicles are described by their current link location, the time they will spend on the link before exiting it, and their destination. The action of a vehicle is the successor link they want to reach when exiting a given link. Pure actions for a player on link ℓ, with a negative waiting time are the successors link of ℓ. When arriving on a link, the waiting time of the player is assigned based on the number of players on the link at this time. As time goes by, the waiting time of a vehicle decreases until it becomes negative, then the vehicle moves to a successor link and the waiting time gets reassigned. The total cost for the vehicle is its travel time. In the corresponding MFG, the vehicles of the N-player game are replaced by a representative vehicle and the probability distribution of the vehicles states.

A. Network and game set up

Time is represented as an interval $\mathcal{T} = [0, T]$ of $\mathbb{R}$. The road network is described by a directed graph $G = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ and $\mathcal{A} \subset \mathcal{V} \times \mathcal{V}$ respectively denote the sets of vertices and links of the road network. When exiting a link $\ell \in \mathcal{A}$, a vehicle chooses one of the possible successor links. In case the link has no successor, the vehicle stays on the link until the end of time. When joining a link, a vehicle get assigned a travel time on this link, that depends on the volume of traffic on the link. More specifically, congestion induces a travel time spent on link $\ell \in \mathcal{A}$ which is a function $c_\ell \in \mathbb{R}_{>0}^{[0,1]}$ of the proportion of vehicles on link ℓ. We assume that c_ℓ is continuous. The congestion functions $(c_\ell)_{\ell \in \mathcal{A}}$ encode the heterogeneity of the roads' sensitivity to traffic volume within the network. The following typical congestion function is based on an example provided by the 1964 traffic assignment manual of the U.S. Bureau of public road functions, see [12, table 1.1]: $c_\ell : \mu \mapsto t_0(1 + \alpha(\mu/\mu_{\ell,c})^\beta)$, where α and β are positive constants, t_0 is the free flow travel time (i.e., the travel time when the link is empty), and $\mu_{\ell,c}$ is the relative capacity of the link ℓ (which, in our context, is to be understood as a capacity in terms of proportion of players).

B. N-player dynamic routing game

Given the above network, this subsection defines a finite-player game. Most of the notations are chosen to ease the presentation of the MFG in the next subsection.

1) Traffic flow environment: For the sake of convenience, we assume that the time horizon T is large enough so that any driver will have time to travel through the network.

Let N be a positive integer. The set of players is $\mathcal{N} = \{1, 2, \ldots, N\}$. Because the congestion functions are defined in terms of ratio of the total number of players (to ease the corresponding MFG definition), the number of players in the game is not necessarily the total number of vehicles N_0 in the real-life scenario. Each player of the model corresponds to a proportion of the real number of vehicles, which allows to define a player as an infinitesimal portion of flow that does not impact network travel time in the MFG. In the case where $N = N_0$, a player is a vehicle. Player $i \in \mathcal{N}$ starts at an origin link $L_0^i \in \mathcal{A}$ with a departure time $W_0^i \in \mathcal{T}$, and has a destination link $D_0^i \in \mathcal{A}$. This is the initial state of the player. Intuitively the player wants to start moving at time W_0^i from L_0^i and tries to reach D_0^i. We assume that the players' initial state are distributed according to a finite-support distribution m_0. Both the origin and the destination are modeled as links, so that the location of the vehicle is always described as a link. In experimental setups, a origin link is added before each origin node and a destination link is added after each destination node. Being on the origin link means having not departed yet, and being on the destination link means having finished their trip.

Then, at any time step t, the state of a player i is not only the link L_t^i where they stand, but also their waiting time W_t^i before exiting this link together with their destination D_t^i. L_t^i and W_t^i are random variables due to the randomness in the action choices. Even though the destination is constant through time ($D_t^i = D_0^i$ for all t), including this information in the state allows to keep track of the objective in the player's policy. So the state space for each driver is $\mathcal{X} = \mathcal{A} \times \mathcal{T} \times \mathcal{A}$, where the first component is for the current location and the last one is for the destination (recall that the destination is represented by a link in our model). Then, the space of vehicle trajectories is $\boldsymbol{\mathcal{X}} = \mathcal{X}^{\mathcal{T}}$. The state trajectories are in the space of triples (location, waiting time, destination), which provide more information than the physical trajectories just in terms of locations. At the population level, the states of all the agents is a vector $\underline{X} = (X^i)_{i \in \mathcal{N}}$. The state space for the whole population is $\underline{\mathcal{X}} = \mathcal{X}^{\mathcal{N}}$, and the corresponding space of trajectories is $\underline{\boldsymbol{\mathcal{X}}} = \mathcal{X}^{\mathcal{T} \times \mathcal{N}}$. We respectively call game state and game trajectory the state and trajectory of the population.

2) Routing policy: When at link ℓ, a player can try to move to another link among the successors of ℓ, and the transition is realized provided the waiting time is 0. The players are allowed to randomize their actions. We thus call strategy function and denote by π a function from $\mathcal{X}$ to $\mathcal{P}(\mathcal{A})$ such that for any $x = (\ell, w, d)$, $\pi(x)$ has support in the successors of ℓ. We denote by Π the set of such strategy functions. A (feedback or closed-loop) policy $\boldsymbol{\pi}$ is a function that associates to each time a strategy function, so it is an element of the set $\mathbf{\Pi} = \Pi^{\mathcal{T}}$ of policies. The notation $\boldsymbol{\pi}_t(\ell'|\ell, w, d)$ represents the probability at time t with which the agent would like to go from ℓ to ℓ' given the fact that their waiting time is w and their destination is d. A policy profile $\underline{\boldsymbol{\pi}}$ is a vector of policy functions with one policy for each player, i.e., it is an element of $\underline{\mathbf{\Pi}} = \mathbf{\Pi}^{\mathcal{N}}$. Studying this class of policies can be justified by the fact that it allows each player to take a decision based only on their own state, which is realistic if the players do not know the situation of the rest of the population. More information could be added in the inputs of the policy (e.g., the proportion of agents on the current link), but this is beyond the scope of this work.

3) State dynamics: Since the players' initial states and actions are randomized, their trajectories are stochastic. Given a policy profile $\underline{\boldsymbol{\pi}} \in \underline{\mathbf{\Pi}}$, $\underline{X}_t = (\underline{L}_t, \underline{W}_t, \underline{D}_t) \in \underline{\mathcal{X}}$ denotes the random variable corresponding to the links, waiting times and destinations for all the players at time $t \in \mathcal{T}$. The stochastic process of the population state is denoted $\underline{\boldsymbol{X}} = (\underline{X}_t)_{t\in\mathcal{T}} \in \underline{\boldsymbol{\mathcal{X}}}$. As indicated above, the players' interactions are through the travel time functions $(c_\ell)_{\ell\in\mathcal{A}}$ taking into account congestion levels. So the interaction between a driver and the rest of the vehicles is only through the proportion of vehicles on the same link. It is thus convenient to introduce the empirical distribution $\nu^N_{\underline{\ell}} \in \mathcal{P}(\mathcal{A})$ corresponding to a location profile $\underline{\ell} = (\ell^i)_{i\in\mathcal{N}} \in \mathcal{A}^N$: for every $\ell \in \mathcal{A}$, $\nu^N_{\underline{\ell}}(\ell') = \frac{1}{N}\#\{i \,|\, \ell^i = \ell'\} \in [0, 1]$, which is the proportion of players on the link ℓ', given $\underline{\ell}$. This is all the information one needs from the game state to compute the interactions between players at link ℓ'. Note that $\nu^N_{\underline{\ell}}(\ell')$ is invariant by permutation of the components of the vector $\underline{\ell}$.

Let us fix a policy profile $\underline{\boldsymbol{\pi}} \in \underline{\mathbf{\Pi}}$. We denote by $\underline{\boldsymbol{U}}$ the $\mathcal{A}^{\mathcal{T}\times\mathcal{X}\times\mathcal{N}}$-valued random variable assigned to the probability distribution given by the policy profile: for each $(t, x, i) \in \mathcal{T}\times\mathcal{X}\times\mathcal{N}$, $U^i_t(x)$ is an $\mathcal{A}$-valued random variable with distribution $\pi^i_t(\cdot|x)$.

The evolution of the state of the game $\underline{\boldsymbol{X}}_t = (\underline{L}_t, \underline{W}_t, \underline{D}_t)$ is given by the following dynamics. At initial time, (L^i_0, W^i_0, D^i_0), $i \in \mathcal{N}$ are given, and then the dynamics is:

$$
\begin{aligned}
t_{k+1} &= t_k + \min\{W^i_{t_k},\ i \in \mathcal{N}\}\\
L^i_{t_{k+1}} &= \begin{cases} U^i_{t_{k+1}}(X^i_{t_k}) & \text{if } i \in I_{t_{k+1}}\\ L^i_{t_k} & \text{otherwise;}\end{cases}\\
W^i_{t_{k+1}} &= \begin{cases} c_{L^i_{t_{k+1}}}\left(\nu^N_{\underline{L}_{t_{k+1}}}(L^i_{t_{k+1}})\right) & \text{if } i \in I_{t_{k+1}}\\ W^i_{t_k} - (t_{k+1} - t_k) & \text{otherwise;}\end{cases}\\
L^i_t &= L^i_{t_k} \qquad \forall k, \forall t \in [t_k, t_{k+1}[, \forall i \in \mathcal{N}\\
W^i_t &= W^i_{t_k} - (t - t_k) \qquad \forall k, \forall t \in [t_k, t_{k+1}[, \forall i \in \mathcal{N}\\
D^i_t &= D^i_0, \qquad t \in \mathcal{T},
\end{aligned}
$$

where $I_{t_{k+1}} := \{i \in \mathcal{N},\ W^i_{t_k} + t_k - t_{k+1} = 0\}$ and using $(t_k)_{k\in\mathbb{N}}$ the sequence of times where one of the vehicles changes link with $t_0 = 0$ and $t_k = T$ if all the players have arrived their destination. The destination is constant through time and is not affected by the policy's randomness. Note that $\boldsymbol{U}^i = (U^i_t)_{t\in\mathcal{T}}$ is defined for all t but used only when the player moves from one link to the next one, i.e., when the waiting time has vanished. This enables reducing any pure (i.e. deterministic) policy as a path choice.

4) Cost function: Given a policy profile $\underline{\boldsymbol{\pi}} \in \underline{\mathbf{\Pi}}$, the cost for player i is the average arrival time which can be defined as:

$$
\begin{aligned}
J^N_i(\boldsymbol{\pi}^i, \boldsymbol{\pi}^{-i}) &= \mathbb{E}_{\underline{\boldsymbol{\pi}}}\left[\min\{t \in \mathcal{T},\ L^i_t = D^i\}\right]\\
&= \mathbb{E}_{\underline{\boldsymbol{\pi}}}\left[\int_{t\in\mathcal{T}} r(X^i_t)dt\right]
\end{aligned}
$$

where $\boldsymbol{\pi}^{-i} = (\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^{i-1}, \boldsymbol{\pi}^{i+1}, \ldots, \boldsymbol{\pi}^N)$, and the instantaneous cost is defined as: for every $x = (\ell, w, d)$, $r(x) = \mathbb{1}_{\ell\neq d}$. Note that the running cost is independent of the (rest of the) population state, contrary to other models for routing or crowd motion in which the interactions are not in the dynamics but in the cost function.

Furthermore, the population is homogeneous (all players have the same dynamics evolution and same running cost), and player i interacts with the other players only through $\boldsymbol{\nu}^N$ and for this reason, the cost function J^N_i does not depend directly on the index i but only on $\boldsymbol{\pi}^i$: as a function, $J^N_i = J^N_{i'}$ for all i'. The policy profile $\boldsymbol{\pi}^{-i}$ for the rest of the population is used only to compute $\boldsymbol{\nu}^N = (\nu^N_t)_{t\in\mathcal{T}}$. Although $\boldsymbol{\pi}^{-i}$ is necessary, it is not sufficient because $\boldsymbol{\nu}^N$ is also influenced by the policy $\boldsymbol{\pi}^i$ chosen by the player under consideration. However, the influence of each player decays as N increases, which will be the basis for the mean-field approach presented in § II-C.

5) Nash equilibrium: Considering that all the players are individually optimizing their own cost leads to the following notion of solution for the game. We refer to e.g. [13] for more details.

Definition 1 (Nash equilibrium): A Nash equilibrium is a policy profile $\underline{\boldsymbol{\pi}}^\star = (\boldsymbol{\pi}^{i\star})_{i\in\mathcal{N}} \in \underline{\mathbf{\Pi}}$ such that:

$$\forall i \in \mathcal{N},\ \forall \boldsymbol{\pi} \in \mathbf{\Pi},\ J^N_i(\boldsymbol{\pi}^{i\star}, \boldsymbol{\pi}^{-i\star}) \leqslant J^N_i(\boldsymbol{\pi}, \boldsymbol{\pi}^{-i\star}).$$

The following result says that, in our model, such equilibria exist.

Theorem 1 (Existence of N-player Nash equilibrium): Assuming the continuity of the cost function with respect to the policy profiles, there exists a Nash equilibrium in the N-player routing game.

Proof: The proportion of players on each link is always a multiple of $1/N$. Since the number of links and the time horizon are finite, there is a finite set of times at which a vehicle can switch link. The set of policy profiles can thus be restricted to a finite set. Therefore, the game can be restated as a game with finite state and action spaces. Assuming the continuity of the cost function with respect to the policy, it has a Nash equilibrium (Kakutani-Fan-Glisckberg theorem [14]). ∎

Besides the above definition, another way to express that a policy profile $\underline{\boldsymbol{\pi}}$ is a Nash equilibrium is to say that the deviation incentive is 0 for every player, where the deviation incentive for player i is:

$$D_i^N(\boldsymbol{\pi}^i, \boldsymbol{\pi}^{-i}) = J_i^N(\boldsymbol{\pi}^i, \boldsymbol{\pi}^{-i}) - \operatorname*{argmin}_{\boldsymbol{\pi}' \in \boldsymbol{\Pi}} J_i^N(\boldsymbol{\pi}', \boldsymbol{\pi}^{-i}).$$

This also serves as a basis to assess the convergence of algorithms towards a Nash equilibrium using the average deviation incentive:

$$\overline{D}^N(\underline{\boldsymbol{\pi}}) = \frac{1}{N}\sum_{i=1}^{N} D_i^N(\boldsymbol{\pi}^i, \boldsymbol{\pi}^{-i}). \quad (1)$$

C. Mean Field approximation

As mentioned in the introduction, solving the above N-player game is infeasible as N is very large. We thus turn to an MFG version of the above routing game, which can be used to provide approximate Nash equilibria and whose quality improves as $N \to +\infty$. This is based on considering the interactions between a typical player and a distribution representing the rest of the population. This is possible thanks to the anonymity and the symmetry in the interactions, which allows us to focus on symmetric Nash equilibria. Intuitively, the law of large numbers allows to consider the state distribution instead of a large number of random variables induced by it.

1) Traffic flow environment: The state of a typical player at time t is a random variable denoted by $X_t = (L_t, W_t, D_t)$ which takes values in $\mathcal{X} = \mathcal{A} \times \mathcal{T} \times \mathcal{A}$. At time 0, the population's state distribution is m_0 and is known to the players.

2) Routing policy: The space of policies is still $\boldsymbol{\Pi}$. For a policy $\boldsymbol{\pi}$, we denote by $\boldsymbol{\pi}_t(\ell'|\ell, w, d)$ the probability with which a typical player using policy $\boldsymbol{\pi}$ would like to go from ℓ to ℓ' given that their waiting time is w and their destination is d. The routing random variable is denoted by U.

3) State dynamics: Assume that an infinitesimal agent uses policy $\boldsymbol{\pi}$ while the rest of the population uses $\boldsymbol{\pi}'$. Let $\boldsymbol{\nu} = (\nu_t)_{t\in\mathcal{T}} \in \mathcal{P}(\mathcal{A})^{\mathcal{T}}$ be the flow of distributions on $\mathcal{A}$ induced by the population that uses $\boldsymbol{\pi}'$. The evolution of a typical player's state is given by the following dynamics. Let $t_0 = 0$ and let (L_0, W_0, D_0) be a given initial state. Then, the dynamics follow:

$$\begin{aligned}
t_{k+1} &= W_{t_k} + t_k \\
L_{t_{k+1}} &= U_{t_k}(X_{t_k}) \\
W_{t_{k+1}} &= c_{L_{t_{k+1}}}\left(\nu_{t_{k+1}}(L_{t_{k+1}})\right) \\
L_t &= L_{t_k} \qquad \forall k, \forall t \in [t_k, t_{k+1}[\\
W_t &= W_{t_k} - (t - t_k) \qquad \forall k, \forall t \in [t_k, t_{k+1}[\\
D_t &= D_0, \qquad t \in \mathcal{T}.
\end{aligned}$$

Here $(t_k)_{k\in\mathbb{N}}$ denotes the sequence of times where the representative player changes link (we take $t_k = T$ when there are no more changes), and $\nu_t(\ell) \in [0,1]$ is the proportion of the mean field population on link ℓ at time t.

4) Cost function: The cost of the typical player using policy $\boldsymbol{\pi}$ when the population uses policy $\boldsymbol{\pi}'$ is defined as:

$$J(\boldsymbol{\pi}, \boldsymbol{\pi}') = \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\pi}'}\left[\int_{t\in\mathcal{T}} r(X_t)dt\right]$$

where the state of the representative player $\boldsymbol{X} = (X_t)_{t\in\mathcal{T}}$ has the above dynamics with policy $\boldsymbol{\pi}$, and the instantaneous cost function r is the same function as in the finite player game (see § II-B.4). Analogously to the N-player game, the policy $\boldsymbol{\pi}'$ is used only to deduce $\boldsymbol{\nu} = (\nu_t)_{t\in\mathcal{T}}$ that appears in the evolution of $\boldsymbol{W}$. So the cost function J could alternatively be written as a function of $(\boldsymbol{\pi}, \boldsymbol{\nu})$ instead of $(\boldsymbol{\pi}, \boldsymbol{\pi}')$. In contrast with the finite player regime, we highlight that here $\boldsymbol{\pi}'$ completely determines $\boldsymbol{\nu}$ because the player under consideration is infinitesimal and hence their policy $\boldsymbol{\pi}$ does not affect the flow $\boldsymbol{\nu}$ of distributions of locations of the population.

5) Nash equilibrium: The counterpart of the N-player Nash equilibrium in the mean-field regime can now be introduced.

Definition 2 (Mean field Nash equilibrium): Definition 3.1. of [15]: A mean field Nash equilibrium (MFNE) is a policy $\boldsymbol{\pi}^\star \in \boldsymbol{\Pi}$ such that: $J(\boldsymbol{\pi}^\star, \boldsymbol{\pi}^\star) \leqslant J(\boldsymbol{\pi}', \boldsymbol{\pi}^\star)$ for all $\boldsymbol{\pi}'$, or equivalently:

$$\boldsymbol{\pi}^\star \in \operatorname*{argmin}_{\boldsymbol{\pi}\in\boldsymbol{\Pi}} J(\boldsymbol{\pi}, \boldsymbol{\pi}^\star).$$

Another way to express that $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ is a MFNE is to say that the average deviation incentive $\overline{D}$ vanishes, where:

$$\overline{D}(\boldsymbol{\pi}) = J(\boldsymbol{\pi}, \boldsymbol{\pi}) - \operatorname*{argmin}_{\boldsymbol{\pi}'\in\boldsymbol{\Pi}} J(\boldsymbol{\pi}', \boldsymbol{\pi}).$$

Theorem 2 (Existence of mean field Nash equilibrium): Assuming the continuity of the cost function with respect to the policy profiles, and assuming that the support of the initial distribution of the waiting time is a finite set, there exists a mean field Nash equilibrium.

Proof: The set of pure policies for the representative player can be restricted to the choice of a path given a departure time. This set is finite as long as the support of the initial distribution of waiting time is. Therefore the argmin map (also called Best response map) is a Kakutani-Fan-Glisckberg map providing the continuity of the cost with respect to the policy of the representative player and of the mean field. ■

One of the advantages of considering a mean field setting, is that any MFNE is automatically a dynamic Wardrop equilibrium.

Theorem 3 (Dynamic Wardrop equilibrium [16]): For any mean field Nash equilibrium, all induced trajectories of players with the same initial state (origin, waiting time, destination), have the same travel time (i.e., the same total cost).

Proof: In case a trajectory used by the representative player has a higher travel time than another one, then the player has an incentive to deviate, and the game is not a Nash equilibrium. ■

Any mean field Nash equilibrium policy $\boldsymbol{\pi}^\star$ can be used by the players in an N-player game. Intuitively, the larger N

is, the closer the population is to the mean field regime. In fact, it can be shown under suitable conditions that $\underline{\boldsymbol{\pi}}^\star = (\boldsymbol{\pi}^\star, \dots, \boldsymbol{\pi}^\star) \in \boldsymbol{\Pi}^N$ is an approximate Nash equilibrium whose quality improves with N in the sense that:

$$\overline{D}^N(\underline{\boldsymbol{\pi}}^\star) \to 0, \qquad \text{as } N \to +\infty.$$

So if all the agents use the mean field Nash equilibrium policy, then any single player's incentive to deviate decreases when the population becomes larger. For example, [15] prove in their setting that: if $\boldsymbol{\pi}^\star$ is an MFNE, then for every $\epsilon > 0$, there exists $N_0 \in \mathbb{N}$ such that for every $N \geqslant N_0$, the N-player policy profile $(\boldsymbol{\pi}^\star, \dots, \boldsymbol{\pi}^\star) \in \boldsymbol{\Pi}^N$ satisfies: $\overline{D}^N(\underline{\boldsymbol{\pi}}) \leqslant \epsilon$.

Next, to illustrate this property in our model, an explicit computation is carried out in the simple Pigou network and then is empirically verified on both Pigou and Braess networks.

III. Experiments

This section shows experimentally that (1) computing the mean field equilibrium is easier than computing the N-player Nash equilibrium using state of the art algorithms (sampled counterfactual regret minimization [17]) and (2) it gives an excellent approximation of the N-player equilibrium when N is large (above 30 in the case of the Pigou [18] and the Braess [19] network). The experiments also show that (3) online mirror descent algorithm [20] enables computing the mean field equilibrium on the Sioux Falls network [21], a classic use case in road traffic network games, with 14,000 vehicles (across two origin-destination pairs) and realistic congestion function.

A. Context

All the experiments are conducted within the OpenSpiel framework [22], an open source library that contains a collection of environments and algorithms to apply reinforcement learning and other optimization algorithms in games. The code is publicly available on GitHub[1].

Networks. As classical network games consider demand between nodes, we add artificial origin and destination links before and after each node in the network (Pigou [18], Braess [19] and Sioux Falls [21]). This enables defining vehicle location only using links, and defining state of not having begun a trip and having finished it.

The *Pigou network* [18] has two links ℓ, ℓ' and two nodes (an origin and a destination one) which come from the conversion of the origin and the destination nodes. A time discretisation of 0.01, with a time horizon of 2 is used. The cost functions are $c_\ell(x) = 2$, $c_{\ell'}(x) = 1 + 2x$ and all the demand leaves the origin link at time 0 and head towards the destination link.

The *Braess network* game is the dynamic extension of the game described in [19]. The network has 5 links AB, AC, BC, BD and CD, one origin node A converted to an origin link OA and a destination node D converted to a destination link DE. The cost functions are $c_{AB}(x) = 1+x$, $c_{AC}(x) = 2$, $c_{AB}(x) = 0.25$, $c_{BD}(x) = 2$, $c_{CD}(x) = 1+x$. All the demand leaves the origin link at time 0 and head towards the destination link. We use a time step of 0.05 and a time horizon of 5.

The *Sioux Falls network* game is used by the traffic community for proof of concepts on network with around 100 links. The network (76 links without the origin and destination links), the link congestion functions, and an origin-destination traffic demand are open source [23]. As the classical routing game [24, Chapter 18] is a static game, the demand is only a list of tuple origin, destination and counts, and does not provide any departure time. We use the network data (including the congestion functions) and generate a demand specific to the game. We model 7,000 vehicles departing at time 0 from node 1 to node 19, and 7,000 vehicles departing at time 0 from node 19 to node 1. We use a time step of 0.5 and a time horizon of 50.

B. Mean field game solves the curse of dimensionality in the number of players

In this section, the mean field equilibrium policy is computed for both the Braess and the Pigou network games. In addition to being considerably faster to compute compared to the N-player Nash equilibrium, the mean field equilibrium provides an excellent approximation when N is above 30.

In the Braess mean field Nash equilibrium policy, the travel time on the three possible paths are equals, which encodes the Nash equilibrium condition of the MFG provided that the travel time on each link is a multiple of the time step, accordingly to theorem 3.

1) While solving N-player game is intractable for large number of players, this can be done for the mean field game: We compare the running time of the algorithms for solving the N-player game and the mean field player game depending on the number of players it models. The counterfactual regret minimization with external sampling (ext CFR) is used in the N-player game, as it is the fastest algorithms to solve the dynamic routing N-player game within the OpenSpiel library of algorithms (comparison done within the OpenSpiel framework are not reported here). Online mirror descent (OMD) is used in the MFG. As the mean field Nash equilibrium does not depends on the number of vehicles the MFG models, the computation time of 10 iterations of OMD is independent of the number of vehicles modeled. On the other hand, the computational cost of 10 iterations of ext CFR increases exponentially with the number of players, making the computation of a Nash equilibrium with a large number of players intractable with the algorithms of the OpenSpiel library.

2) The mean field equilibrium policy is a good approximation of the N-player equilibrium policy whenever N is large enough: In the Pigou network game, the mean field equilibrium policy is almost a Nash equilibrium in the N-player game as soon as N is larger than 20 players, see fig. 1.

[1] `https://github.com/deepmind/open_spiel`

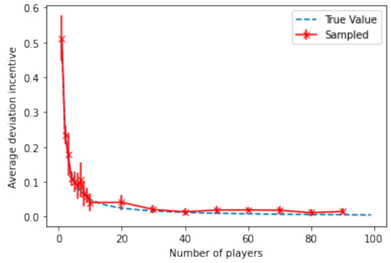

Fig. 1. Average deviation incentive of the Nash equilibrium mean field policy in the N-player game as a function of N in the case of the Pigou game. The sampled value is the value computed in OpenSpiel by testing all the possible pure best responses, and sampling game trajectories to get the expected returns.

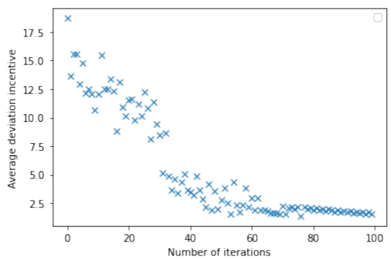

Fig. 2. Online mirror descent average deviation incentive in the Sioux Falls MFG as a function of the number of iterations of the descent algorithm.

In the Braess network game, the mean field equilibrium policy is almost a Nash equilibrium in the N-player game as soon as N is larger than 30 players.

C. Mean field game approach can be extended to more complex set ups

The MFG approach solves the curse of dimensionality in the number of players (whenever the number of possible states is much below the number of players). It can be extended to more complex setups than the Pigou and Braess networks such as realistic traffic networks with demand. This section focuses on the extension of MFG approach to one of the classical benchmark network game used by the traffic community: the Sioux Falls network [21].

The experiment shows the ability to learn the mean field equilibrium policy on this 76 links network, with 14,000 vehicles going to two different destinations. Using online mirror descent, we see that the average deviation incentive decreases to 1.55 (for a travel time of 27) over 100 iterations, see fig. 2. We use a fixed learning rate of 1 in the 30 first iterations of the algorithm, 0.1 in the 31 to the 60 first iterations and a fixed learning rate of 0.01 in the 40 remaining iterations to produce fig. 2.

The resulting mean field policy is not exactly the Nash equilibrium policy of the MFG as its average deviation incentive is 1.55 (for a travel time of 27.5).

Average deviation of the learned mean field policy cannot be computed numerically in the 14,000 player game, due to the large number of players.

REFERENCES

[1] D. Schrank, B. Eisele, and T. Lomax, "2019 Urban Mobility Report," *Texas A&M Transportation Institute*, no. August, 2019.

[2] T. Cabannes, M. A. S. Vincentelli, A. Sundt, H. Signargout, E. Porter, V. Fighiera, J. Ugirumurera, and A. M. Bayen, "The impact of GPS-enabled shortest path routing on mobility: a game theoretic approach," *Transportation Research Board 97th Annual Meeting*, 2018.

[3] T. Cabannes, "The impact of information-aware routing on road traffic. from case studies to game-theoretical analysis and simulations," Ph.D. dissertation, University of California, Berkeley, 2022.

[4] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 195–259, 2009.

[5] K. Cui, A. Tahir, G. Ekinci, A. Elshamanhory, Y. Eich, M. Li, and H. Koeppl, "A survey on large-population systems and scalable multi-agent reinforcement learning," *arXiv preprint arXiv:2209.03859*, 2022.

[6] O. Guéant, "Existence and uniqueness result for mean field games with congestion effect on graphs," *Applied Mathematics & Optimization*, vol. 72, no. 2, pp. 291–303, 2015.

[7] Y. Achdou, M.-K. Dao, O. Ley, and N. Tchou, "Finite horizon mean field games on networks," *Calculus of Variations and Partial Differential Equations*, vol. 59, no. 5, pp. 1–34, 2020.

[8] D. Bauso, X. Zhang, and A. Papachristodoulou, "Density flow in dynamical networks via mean-field games," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1342–1355, 2016.

[9] R. Salhab, J. Le Ny, and R. P. Malhamé, "A mean field route choice game model," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1005–1010.

[10] T. Tanaka, E. Nekouei, A. R. Pedram, and K. H. Johansson, "Linearly solvable mean-field traffic routing games," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 880–887, 2020.

[11] G. W. Brown, "Iterative solution of games by fictitious play," *Activity analysis of production and allocation*, vol. 13, no. 1, pp. 374–376, 1951.

[12] M. Patriksson, *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015.

[13] R. B. Myerson, *Game theory*. Harvard university press, 2013.

[14] I. L. Glicksberg, "A further generalization of the kakutani fixed point theorem, with application to nash equilibrium points," *Proceedings of the American Mathematical Society*, vol. 3, no. 1, pp. 170–174, 1952.

[15] N. Saldi, T. Basar, and M. Raginsky, "Markov–nash equilibria in mean-field games with discounted cost," *SIAM Journal on Control and Optimization*, vol. 56, no. 6, pp. 4256–4287, 2018.

[16] J. G. Wardrop and J. I. Whitehead, "Correspondence. Some theoretical aspects of road traffic research," *ICE Proceedings: Engineering Divisions 1*, 1952.

[17] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione, "Regret minimization in games with incomplete information," *Advances in neural information processing systems*, vol. 20, pp. 1729–1736, 2007.

[18] A. C. Pigou, *The economics of welfare*. Palgrave Macmillan, 1920.

[19] D. Braess, A. Nagurney, and T. Wakolbinger, "On a paradox of traffic planning," *Transportation science*, vol. 39, no. 4, pp. 446–450, 2005.

[20] J. Perolat, S. Perrin, R. Elie, M. Laurière, G. Piliouras, M. Geist, K. Tuyls, and O. Pietquin, "Scaling up mean field games with online mirror descent," *arXiv preprint arXiv:2103.00623*, 2021.

[21] M. Abdulaal and L. J. LeBlanc, "Continuous equilibrium network design models," *Transportation Research Part B: Methodological*, vol. 13, no. 1, pp. 19–32, 1979.

[22] M. Lanctot, E. Lockhart, J.-B. Lespiau, V. Zambaldi, S. Upadhyay, J. Pérolat, S. Srinivasan, F. Timbers, K. Tuyls, S. Omidshafiei, D. Hennes, D. Morrill, P. Muller, T. Ewalds, R. Faulkner, J. Kramár, B. D. Vylder, B. Saeta, J. Bradbury, D. Ding, S. Borgeaud, M. Lai, J. Schrittwieser, T. Anthony, E. Hughes, I. Danihelka, and J. Ryan-Davis, "OpenSpiel: A framework for reinforcement learning in games," *CoRR*, vol. abs/1908.09453, 2019. [Online]. Available: http://arxiv.org/abs/1908.09453

[23] Transportation Networks for Research Core Team, "Transportation networks for research, free of use," https://github.com/bstabler/TransportationNetworks, accessed: 2021-09-01.

[24] N. Nisam, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press, 2007.