October 2020, to appear in: ***Operations Research***

Atomic Dynamic Flow Games: Adaptive versus Nonadaptive Agents

Zhigang Cao
School of Economics and Management, Beijing Jiaotong University, Beijing 100044, China

Bo Chen
Warwick Business School, University of Warwick, Coventry, CV4 7AL, United Kingdom

Xujin Chen
Academy of Mathematics and Systems Science, Chinese Academy of Sciences;
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China

Changjun Wang
Faculty of Science, Beijing University of Technology, Beijing, 100124, China

We propose a game model for selfish routing of atomic agents, who compete for use of a network to travel from their origins to a common destination as fast as possible. We follow a frequently used rule that the latency an agent experiences on each edge is a constant transit time plus a variable waiting time in a queue. A key feature that differentiates our model from related ones is an edge-based tie-breaking rule for prioritizing agents in queueing when they reach an edge at the same time. We study both nonadaptive agents (each choosing a one-off origin-destination path simultaneously at the very beginning) and adaptive ones (each making an online decision at every nonterminal vertex they reach as to which next edge to take). On the one hand, we constructively prove that a (pure) Nash equilibrium (NE) always exists for nonadaptive agents, and show that every NE is weakly Pareto optimal and globally first-in-first-out. We present efficient algorithms for finding an NE and best responses of nonadaptive agents. On the other hand, we are among the first to consider adaptive atomic agents, for which we show that a subgame perfect equilibrium (SPE) always exists, and that each NE outcome for nonadaptive agents is an SPE outcome for adaptive agents, but not vice versa.

Key words: Selfish atomic routing; deterministic queuing; adaptive routing; subgame perfect equilibrium; Nash equilibrium.

---

1. Introduction

Selfish routing is a fundamental model for network traffic, with diverse applications (Wardrop 1952, Roughgarden and Tardos 2002, Roughgarden 2007). The problem is dynamic in essence. However, most of the literature is based on latency functions, which are good approximations of static flows, but not fully satisfactory due to the following weaknesses. First, a latency function is overly symmetric in that agents choosing the same road segment impede each other in the same way, which is usually not the case, as earlier agents may delay the later ones but not vice versa. Second, a latency function imposes the same delay upon all agents who travel along the road segment at any time, even if their travel periods along the segment do not overlap, which is unreasonable, as for example travel in peak hours takes more time than in off-peak hours. A well-recognized method to overcome the above weaknesses is to apply the *deterministic queuing* (DQ) rule (Vickrey 1969, Hendrickson and Kocur 1981, Koch and Skutella 2011, Cominetti et al. 2015, Scarsini et al. 2018). However, the previous DQ-based atomic models of selfish routing usually suffer from the problems of non-existence of a (pure strategy Nash) equilibrium or hardness in computing an equilibrium or a best response, especially when there are multiple origins.

One of the key features that differentiate various DQ-based atomic models is how to break ties when more agents than the capacity limit are trying to enter a road segment at the same time. In this paper, by introducing an edge-priority tie-breaking rule, we propose a new DQ-based atomic dynamic flow model, which we prove possesses several desirable properties and consequently leads to a solution of the aforementioned problems.

1.1. Atomic dynamic flows

Instead of a latency function, two integer parameters are used in DQ to characterize each network edge (road segment) e: its capacity c_e and length t_e. The travel cost that an agent bears for using edge e is a *variable* waiting time in the queue at edge e plus the *fixed* transit time t_e (i.e., the travel speed is normalized to 1). Time is discretized. At each time step, a (possibly empty) queue of completely ranked agents are waiting at the entrance of each edge. As many as possible up to c_e agents ranked highest in the queue start moving along different lanes of edge e, while the remaining agents (if any) still wait in the queue for the next time steps. Once an agent starts moving along edge e, he will reach e's terminal t_e time units later. In reality, one traffic paradigm that exhibits this atomic DQ feature is the expressway traffic. Imagine that an expressway road e consists of c_e lanes, and at the entrance of each lane, there is a toll booth collecting a toll from each car passing it. For each booth, at most one car can pass through it each time and begin to travel along the corresponding lane with a uniform speed (meaning the transit time of road e can be viewed as a constant t_e). In this paper, based on this atomic DQ rule, we propose a model that is very similar to the one in Scarsini et al. (2018) but has a crucial difference on the tie-breaking rules.

Network and inflows. We are given an acyclic directed network in which neighboring vertices may be joined by one or more edges. Since we allow for multiple edges, we may assume that each edge models a lane, and thus has a *unit capacity*. The network has one or more origins and a single common destination. At each time point and each origin, a (possibly empty) set of selfish agents enter the network, trying to reach the destination as quickly as possible. Initially, the agents who enter the network at the same time and from the same origin are associated with an original ranking among them, which is *temporarily valid* only when they enter the network.

Edge-priority tie-breaking rule. The queue at each edge is updated according to two criteria: (i) the *local first-in-first-out (FIFO) principle* — an agent who reaches the queue of an edge earlier also leaves the queue earlier, and (ii) the pre-specified *edge priorities* — if two agents reach the queue at the same time, their queue ranks are determined by the priorities of the preceding edges from which they enter the edge: higher priority gives higher rank. Our edge-priority tie-breaking is generalized from various real-world traffic regulation rules, such as right turning traffic should give way to oncoming traffic and side-road traffic should give way to main-road traffic.

1.2. Nonadaptive agents versus adaptive agents

We consider two types of selfish agents, referred to as nonadaptive and adaptive, respectively. Nonadaptive agents make their routing decisions only at the very beginning (i.e., time 0) as to which origin-destination path to take, no matter what time they enter the network. On the other hand, adaptive agents make routing decisions at *every* nonterminal vertex they reach as to which next edge to take. In particular, their decisions at a vertex may depend on the choices of other agents in the history.

In accordance, we investigate two submodels of the game, denoted as Γ^{N} and Γ^{A}, which are played by nonadaptive and adaptive agents, respectively. In terms of game theory, Γ^{N} is a normal-form game (a.k.a. a static game), whose standard solution concept is Nash equilibrium (NE), and Γ^{A} is an extensive-form game (a.k.a. a dynamic game), whose standard solution concept is subgame perfect equilibrium (SPE). (See Sections 4 and 5.1 for formal definitions of the equilibrium concepts.) The following warmup example illustrates what equilibria of the two submodels may look like, as well as their possible differences.

Example 1. In the network of Figure 1, every edge has a unit capacity and a unit length. Edge e_1 (resp. e_2) has a higher priority than e_3 (resp. e_4). The game has only two agents, 1 and 2, who enter the network via the common origin o at the same time 1, and make their ways to the common destination d. Agent 1 has a higher original rank than agent 2.

- Game Γ^{N} admits six NEs, where the two agents adopt edge-disjoint o-d paths, all bringing them the same travel cost of 3.
- In game Γ^{A}, agent 1 takes an adaptive strategy in the following sense. He initially chooses edge ov, and then chooses e_1 unless (at vertex v he finds that) agent 2 used edge ou_2, in which case he chooses edge e_2. Agent 2 always follows the upper path ou_1w_1d. It can be checked

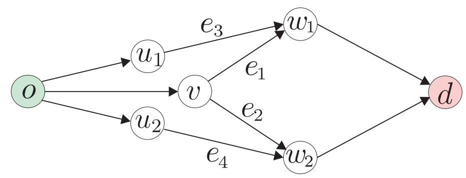


Figure 1 An SPE of game Γ^A may not induce an NE of game Γ^N

that these choices yield a strategy profile that is an SPE of Γ^A, where other off-equilibrium behaviors of the two agents can be easily defined, incurring a travel cost 3 to agent 1, and 4 to agent 2.

Note that the induced path profile by the above SPE of Γ^A, ovw_1d for agent 1 and ou_1w_1d for agent 2 (which have edge w_1d in common), is not an NE of game Γ^N.

1.3. Contributions

As in other models of atomic dynamic network flows, complicated and sometimes unpredictable chain effects form a great obstacle in our analysis. For example, a Braess-like paradox that resembles the one in Scarsini et al. (2018) (but with a different flavor) still exists in our model (see Example 4 in Section 4.2). Yet we are able to demonstrate that the proposed model admits the following positive results.

NE existence. We prove by construction that an NE for Γ^N is guaranteed to exist. It is well recognized that guaranteeing the existence of an equilibrium in dynamic flow models (especially those with multiple origins) is challenging, due to either inherent system instability or technical difficulties (Hoefer et al. 2009, Werth et al. 2014), even for nonatomic models (Anshelevich and Ukkusuri 2009, Koch and Skutella 2009, Meunier and Wagner 2010, Cominetti et al. 2015). To the best of our knowledge, no previous model of atomic dynamic flows has been proved to guarantee NE existence when multi-origin networks with local FIFO principle are considered.

SPE existence. Our work is among the first to consider adaptive agents and to establish the existence of an SPE. Although the standard game-theoretical concept of SPE (Selten 1965) is not new to the area of traffic flow games (c.f. Correa et al. 2019), no previous paper applies it to "doubly dynamic" flow games in that not only flows evolve over time but also agents make decisions over time at road segment intersections.

NE realization by SPE. We build a close connection between games Γ^N and Γ^A by showing that, the NE outcome set of Γ^N is a proper subset of the SPE outcome set of Γ^A. On the one hand, given any NE of Γ^N, we can construct an SPE of Γ^A whose realized path profile is exactly the given NE. On the other hand, an SPE outcome of Γ^A may not be an NE of Γ^N (see Example 1).

The proper inclusion reaffirms the intuition that Γ^{A} is more flexible than Γ^{N} (see also Example 7) and builds a bridge between them. In particular, Γ^{N} is more technique-friendly than Γ^{A}; all results established for NEs of Γ^{N} automatically hold for a subset of SPEs of Γ^{A}.

NE characterization. We provide a characterization of all NEs of Γ^{N}. Given a path profile of nonadaptive agents, let them be batched according to their arrival times at the common destination. A path profile is called *iteratively batch-dominant* if there is no way for agents in a later batch (no matter how they coordinate) to affect any agent in an earlier batch, provided all earlier agents follow their routes in the path profile. We prove that a path profile is an NE of Γ^{N} if and only if it is iteratively batch-dominant. Applying this characterization, we show that each NE of Γ^{N} (and hence a significant proportion of SPEs of Γ^{A}) possesses many desirable properties, including:

- Strong NE: each NE is a strong NE, and thus weakly Pareto efficient, i.e., there are no routing choices that could make every agent strictly better off; and
- Global FIFO: if agent i enters the network earlier than agent j from the same origin, then i exits the network no later than j.

Note that the above characterization and properties are satisfied by all NEs of game Γ^{N} without any additional constraints on agent behaviors or network topologies, whereas the literature usually can establish the properties for only some special NEs or NEs on special networks (Harks et al. 2018, Scarsini et al. 2018). In particular, while the existence of a strong NE (which must be an NE by definition) is known in the literature of atomic dynamic flow games (e.g., Werth et al. 2014), we are the first to show the equivalence between NEs and strong NEs for a class of these games.

Computational results. We design algorithms that efficiently construct an NE of Γ^{N}, a best response of any agent to any strategy profile of Γ^{N}, and an SPE of Γ^{A}. Our algorithms exploit a somewhat surprising fact that a greedy Dijkstra-like approach, which takes maximum advantage of the edge priority rule, is able to identify a path that can circumvent the intricate chain effects. Such computability is in sharp contrast with previous hardness results on related games of atomic dynamic flows, e.g., NP-completeness for determining NE existence (Werth et al. 2014) and NP-hardness for computing a best response (Hoefer et al. 2009, 2011, Ismaili 2017).

To summarize, this paper offers modelling, theoretical, technical as well as computational contributions to the literature of atomic dynamic flow games. Given that there has been little consensus on the characteristics of a canonical model for atomic dynamic flow games due to the inherent intractability (c.f. Correa and Stier-Moses 2010), our model (or its variation) arguably may have a potential to serve as a candidate for standard models in future studies.

2. Related literature

Compared with the relatively mature theory of static flow games, the study for the dynamic flow games, a.k.a. routing games over time, is still in its early stage. Vickrey (1969) and Yagar (1971) initialize the investigation of dynamic flow games, where they focus on analyzing NEs for small-sized concrete examples. Subsequent studies are extensive since the last two decades, encompass various models to investigate equilibrium behaviors of selfish agents, and adopt a wide variety of methodologies from mathematical programming, optimal control, variational inequalities, algorithmic game theory, and simulations (see Peeta and Ziliaskopoulos 2001, Koch and Skutella 2009, Cominetti et al. 2017, and the references therein). Under dynamic queuing, little is known about general equilibrium properties, until recent exciting progress on deriving equilibrium existence, uniqueness, characterizations and constructions (Meunier and Wagner 2010, Koch and Skutella 2011, Cominetti et al. 2015, Scarsini et al. 2018). We discuss the study of equilibria for two major subbranches of dynamic flow games, atomic models and nonatomic models, in the following two subsections, respectively.

2.1. Atomic dynamic flow games

To the best of our knowledge, almost all of the related atomic models studied are of nonadaptive agents, and their solution concepts are NEs. A recent important development on DQ-based games of atomic dynamic flows is Scarsini et al. (2018), which is one of the most related references to our studies in this paper. This study has several notable differences from our work. First, to break ties, Scarsini et al. (2018) place priorities on agents rather than on edges, i.e., a fixed priority ordering of all the agents is applied globally. Second, they only study nonadaptive agents in single-origin single-destination networks. In fact, when agents are adaptive in their model, an SPE may not exist. Third, they focus on seasonal inflows and how the transient phases impact the long-run steady outcomes, whereas their notion of steady outcome does not apply in our model because the inflows we consider are not restricted to be seasonal. Finally, they concentrate on a special kind of NE named *uniformly fastest route* (UFR) equilibrium, for which they prove the existence on single-origin single-destination networks. Scarsini et al. (2018) also obtain a variant of Braess's paradox: adding some initial queues in the network may decrease the worst average travel cost at an NE. The paradox differs from ours in that it involves route changes (see Section 4.2). Under the model of Scarsini et al. (2018), Ismaili (2017) shows many negative results when multiple origin-destination pairs are involved, including non-existence of an NE, and the NP-hardness and inapproximability of computing a best response, etc.

In Werth et al. (2014), more variants of atomic dynamic flow games are considered under a discrete-time DQ model, where finitely many agents are ready to start from their origin(s) at the very beginning. Apart from the sum-type objective as considered in Scarsini et al. (2018) and in this paper, Werth et al. (2014) also study the bottleneck-type objective, where each agent tries to minimize his expense on the slowest edge of his chosen path. To break ties, the global priorities placed on agents as in Scarsini et al. (2018) are discussed for both the sum-objective and bottleneck-objective models, while the local priorities placed on edges as in this paper are investigated only for the bottleneck-objective model. Werth et al. (2014) focus on computational issues on NEs. On the positive side, a greedy algorithm is proposed to efficiently compute an NE in the single-origin single-destination game with sum-type objective and agent priorities. On the negative side, the multi-origin multi-destination game with bottleneck-type objective is shown to suffer from intractabilities, such as non-existence of an NE under either agent or edge priorities, the NP-completeness for determining NE existence and the co-NP-completeness for testing NE in an acyclic directed network under edge priorities.

Among the earliest papers studying atomic dynamic routing games, Hoefer et al. (2009, 2011) are concerned with computing NEs and best responses for a finite number of weighted agents (with sum-type objectives) in a unit-capacitated directed network, where in a continuous-time setting the transit speed of each agent is inversely proportional to his weight. When the local FIFO principle is coupled with the global agent priorities for tie-breaking, the game turns out to be a generalization of the sum-objective model of Werth et al. (2014), and for unweighted agents (i.e., those with a uniform transit speed) in a single-origin network, the game admits a strong NE, which can be computed efficiently. Somewhat surprisingly, computing best responses is NP-hard even in the case of single-origin single-destination networks with unweighted agents. Harks et al. (2018) study an atomic DQ-based dynamic flow game without the local FIFO principle. They analyze the impact of global agent priority ordering on the efficiency of NEs, and show that an NE is polynomially computable. Other related works include Koch (2012) and Kulkarni and Mirrokni (2015).

2.2. Nonatomic dynamic flow games

More previous works investigate nonatomic models, which are usually more tractable than their atomic counterparts. In the nonatomic setting, every agent, aiming at earliest arrival at his destination, represents an infinitesimal amount of flow (a.k.a. fluid), for which neither tie-breaking rules nor road lanes play a role. Different scholars generalize the Wardrop equilibrium (Wardrop 1952) to dynamic versions from different perspectives. These solution concepts resemble more or less NEs where each agent follows a dynamic shortest path (in various sense) that takes time-varying delay into account. The solutions and other dynamic equilibrium concepts developed for nonatomic dynamic flow games do not consider off-equilibrium situations — this is a key difference from SPE.

Since the emergence of the purely existential results (Meunier and Wagner 2010), significant efforts have been made to understand the structure and computational properties of dynamic equilibria in nonatomic queuing networks. Koch and Skutella (2009, 2011) are the first to apply a DQ rule with local FIFO principle to study nonatomic dynamic flow games. They investigate the continuous-time single-origin single-destination case (called a *temporal routing game*) with uniform inflow rates. They characterize the so-called *Nash flows over time* with the *universal FIFO condition* that no flow overtakes another, and equivalently with an analogue to the Wardrop principle that flow is only sent along dynamic shortest paths. Cominetti et al. (2015) prove by construction the existence and uniqueness of the Nash flow over time of temporal routing games in a more general setting with piecewise constant inflow rates. For the multi-origin multi-destination case, Cominetti et al. (2015) prove in a nonconstructive way that a Nash flow over time exists when the inflow rates belong to the space of p-integrable functions with $1 < p < \infty$. Macko et al. (2013) show that Braess's paradox happens more frequently in the temporal routing model than in its static counterpart. Anshelevich and Ukkusuri (2009) consider a dynamic routing game whose monotone increasing edge-latency functions are more general than DQ models but still obey the local FIFO principle. For the single-origin single-destination case, they show the existence, uniqueness and polynomial-time computability of the Nash flow over time. For the multi-origin multi-destination case, examples are presented to show that neither the existence nor the uniqueness can be guaranteed.

In the related works discussed above, as in our nonadaptive model, agents' strategies are their origin-destination paths. Sometimes these path-based models have alternative edge-based representations (c.f. the literature review in Long and Szeto (2019)). For other representations studied in the literature, the interested reader is referred to Long et al. (2013). In the rest of this subsection, we discuss more works that are closely related to agents' adaptive behaviors, where their strategies are richer than mere path selections.

Among these works, Graf and Harks (2019) is closest to our adaptive model in that agents also make decisions over time. However, agents in their model are not completely rational as in our model, but myopic in that whenever they face the choice of which next edge to take, they always choose the one that is on a currently shortest path that is evaluated by current travel times and queuing delays. Graf and Harks show that an equilibrium under these dynamic behaviors exists in multi-origin multi-destination networks with measurable inflow rates.

Hamdouch et al. (2004) study a dynamic flow game on an edge-capacitated network. At the very beginning of this game, for every nonterminal vertex of the network, all agents simultaneously choose a time-dependent preference order, referred to as a list, of some of the outgoing edges. This is a random model, and the probability that an agent moves along an edge depends on his chosen list at the edge's tail vertex, the residual edge capacities, and the number of his competitors as well as their lists. The strategies of an agent in their model, though also adaptive in some sense, are not so adjustable as in our adaptive model and less demanding for the intelligence level of the agent. Using a variational inequality approach explored by Marcotte et al. (2004), Hamdouch et al. (2004) prove the existence of an NE, which is called a strategic equilibrium following Marcotte et al. (2004), but they do not consider SPE as in our paper.

A large body of related literature considers both path choice and departure time as decision variables, but none of these works applies SPE as a solution concept (the reader is referred to Guo et al. (2018) for a literature review along this line of research). Besides the usual approach of variational inequalities, the approach of differential equations also turns out to be successful in studying these problems. Based on conservation laws (which are popular in differential equations) where the flow speed function is density dependent or density and location dependent, Bressan and Han (2013) and Han et al. (2013) prove that NEs exist in multi-origin multi-destination networks under some constraints on functional properties and trip volumes.

The rest of this paper is organized as follows. In Section 3, we present a formal mathematical model of our atomic dynamic flows. In Sections 4 and 5, we study its normal-form game setting Γ^{N} for nonadaptive agents and its extensive-form setting Γ^{A} for adaptive agents, respectively. In Section 6, we conclude our paper with some remarks on future research directions. All proofs and further discussions are provided in the Electronic Companion.

3. The flow model

All paths discussed in this paper are directed and simple. In our model of atomic dynamic flows, we are given a finite *acyclic* directed multi-graph $G=(V,E)$, with V being the vertex set and E the edge set. There is a distinguished vertex d called the *destination*; for each vertex $v\in V$, there is at least one path from v to d, called a v-d path.

Further to our unit-capacity assumption discussed in Section 1.1, which can be viewed as part of our modeling related to edge priorities, we also assume unit edge-length throughout the paper for the convenience of our exposition. The generality of this additional assumption will be discussed in Section 6.

Unit Assumption. Each edge of the input network has a unit capacity and a unit length.

For each $v\in V$, a complete priority order $\prec_v$ is pre-specified over all edges incoming to v. We denote by $e_1\prec_v e_2$ if edge e_1 has a higher priority than edge e_2. Time is discretized as $0,1,2,\ldots$, and may be infinite. Initially, at time 0, there is a (possibly empty) initial ranked queue $\mathbb{Q}_e^0$ of agents at the tail part of each edge $e\in E$. (NB: This initial setting is slightly more general than usual empty networks.) At each integer time point $r\geq 1$ and each vertex $v\in V$, a (possibly empty) set $\Delta_{r,v}$ of finitely many agents enter G from their common origin v. They are associated with original ranks among them, which are *temporarily valid* only at time r. Henceforth, we assume $\Delta_{r,v}$ is an ordered set with agents ordered by their original ranks. Throughout this paper, $\Delta := (\cup_{e\in E}\mathbb{Q}_e^0)\cup(\cup_{r\geq 1,v\in V}\Delta_{r,v})$ denotes the set of all agents.

For brevity, we consider w.l.o.g. every vertex as a possible origin: If vertex v is not an origin in the usual sense, then all sets in $\{\Delta_{r,v} : r \geq 1\}$ are empty. Note that no agent in $\Delta_{r,d}$ has impact on the game, as he never touches any edge of the network. To ease our writing, we frequently write $v \in V$ instead of $v \in V\backslash\{d\}$.

Each agent in Δ goes through some path in G ending at destination d and leaves G from d. The starting vertex of this path is called the starting vertex of this agent. Each agent, when reaching a vertex v ($\neq d$), immediately enters an edge e outgoing from v without any delay. We assume that all agents (if any) in $\mathbb{Q}_e^0$ enter e at time 0. At any integer time $s \geq 0$, all agents (if any) who have entered e but not yet exited queue at the tail part of e, and only the unique head of the queue (namely the one with the highest rank) leaves (recall the Unit Assumption). This queue head spends one time unit in traversing e from its tail to its head and exits e at time $s+1$.

If both agents i and j go through edge e (with tail vertex u_e), they are ranked for entering and therefore exiting the queue at e according to the following ranking rules (R0)–(R4), exactly one of which applies (by checking sequentially in the same order as their indices).

(R0) If i and j are both in the initial queue $\mathbb{Q}_e^0$, then their ranks agree with the ranks in $\mathbb{Q}_e^0$.

(R1) If i enters e earlier than j, then i has a higher rank.

(R2) If they enter e at the same time through two different edges incoming to u_e, then ranks at e are determined by the priority order $\prec_{u_e}$ on the two edges, higher priority giving higher rank.

(Note that if neither i nor j takes u_e as his origin, then the queuing rules (R0)–(R2) are enough to rank the agents. Otherwise, the following (R3) or (R4) will be needed.)

(R3) If only one of them, say i, takes u_e as his origin, then i has a higher rank than j (who must have entered e through an edge incoming to u_e).

(R4) If they both take u_e as their common origin, then their ranks on e are determined by their original ranks.

The above flow regulations will be referred to as the *edge-priority DQ rule*. Following this rule, by assuming different rationality levels of agents, we study in the following two sections two submodels, denoted as Γ^{N} and Γ^{A}, for games of nonadaptive agents and adaptive ones, respectively. In both games, each agent tries to arrive at the common destination d as early as possible. We shall slightly abuse Γ^{N} and Γ^{A} to denote both game models and corresponding game instances.

4. The game of nonadaptive agents

The first submodel Γ^{N} assumes that agents are of a relatively low rationality level, or alternatively, they do not have updated information about other agents. Specifically, the agents are *nonadaptive* in that they each select a path from their own origins to the common destination d simultaneously at the very beginning, i.e., time 0. As soon as the agents enter the network G at the time points specified by the game input $(\mathbb{Q}_e^0)_{e\in E}$ and $(\Delta_{r,v})_{r\geq 1,v\in V}$, they will always follow the chosen paths and never deviate from them at any intermediary vertex. It is worth noting that agents in $\cup_{r\geq 1,v\in V}\Delta_{r,v}$ make their decisions before they enter the network.

For each agent $i\in\Delta$, let $\mathscr{P}_i$ denote his strategy set. If $i\in\mathbb{Q}_e^0$, then $\mathscr{P}_i$ is the set of paths in G starting from edge e and ending at destination d. If $i\in\Delta_{r,v}$, then $\mathscr{P}_i$ is the set of v-d paths in G. For any agent $i\in\Delta$ and path profile $\boldsymbol{p}=(P_j)_{j\in\Delta}$ with $P_j\in\mathscr{P}_j$ for all $j\in\Delta$, we use $t_i^d(\boldsymbol{p})$ to denote the arrival time of i at destination d under $\boldsymbol{p}$. We will use the terms "path profile" and "routing" interchangeably. In terms of game theory, Γ^{N} is a normal-form game, for which we apply the standard solution concept of Nash equilibrium (NE).

Definition 1 (NE). A path profile $\boldsymbol{p}$ of Δ is a *Nash equilibrium* (NE) of Γ^{N} if no agent can gain by uniliteral deviation, i.e., $t_i^d(\boldsymbol{p})\leq t_i^d(P_i',\boldsymbol{p}_{-i})$ for all $i\in\Delta$ and $P_i'\in\mathscr{P}_i$, where $\boldsymbol{p}_{-i}$ is the partial path profile of $\boldsymbol{p}$ for agents in $\Delta\backslash\{i\}$.

4.1. NE existence

In this subsection, we constructively prove that every game Γ^{N} admits an NE. Recall from game theory that a dominant NE is a strategy profile where every agent uses a dominant strategy in that it is always optimal for him regardless of how other agents act. Observe that the following strategy profile that extends the idea of dominance is still an NE: the agents can be ordered such that the strategy of the first agent is a dominant one; and for any $k\geq 2$, subject to the condition that the first $k-1$ agents follow their respective strategies, the strategy of agent k is optimal for him regardless of the choices of the remaining agents. To prove NE existence for game Γ^{N}, we refine this idea on usual type of dominance to be a more specific and stronger iterative dominance defined as follows.

For any nonnegative integer k, we write $[k]$ for the set of all positive integers no more than k. Before providing a formal definition, we call a path profile $\boldsymbol{p}$ an *iteratively dominant* NE, or an IDNE for short, if the agents can be reindexed (ordered) as $1,2,\ldots$ such that small-index agents dominate large-index agents in the following iterative sense:

- No matter what paths other agents in $\Delta\backslash\{1\}$ choose, by following his path in $\boldsymbol{p}$, agent 1 reaches every vertex of the path (including the destination d) at the earliest possible time that an agent in Δ can achieve among all path profiles.

- Iteratively for every $k = 2, 3, \ldots$, assume that the first $k-1$ agents $1, \ldots, k-1$ fix their paths as in $\boldsymbol{p}$. No matter what paths other agents outside $[k]$ choose, by following his path in $\boldsymbol{p}$, agent k reaches every vertex of the path (including the destination d) at the earliest possible time that an agent outside $[k-1]$ can achieve among all path profiles where the first $k-1$ agents follow the given paths in $\boldsymbol{p}$.

For convenience, we call the above reindex of agent an *iterative dominant order* for $\boldsymbol{p}$, and call the paths in $\boldsymbol{p}$ the agents' (*associated*) *dominant paths*. Formally, we have the following definition, where for path profile $\boldsymbol{p}$ and agent subset S, the partial path profile of $\boldsymbol{p}$ for agents in S is written as $\boldsymbol{p}_S$.

Definition 2 (IDNE). A path profile $\boldsymbol{p}$ of Δ is an *iteratively dominant NE* (IDNE) of Γ^{N} if the agents can be reindexed as $1, 2, \ldots$ such that for any index $k \geq 1$, vertex v on agent k's path in $\boldsymbol{p}$, and partial path profile $\boldsymbol{q}$ for agents in $\Delta \backslash [k]$, the following sequential optimality holds:

$$t_k^v(\boldsymbol{p}_{[k]}, \boldsymbol{q}) = \min\{t_j^v(\boldsymbol{p}_{[k-1]}, \boldsymbol{r}) : j \in \Delta \backslash [k-1], \boldsymbol{r} \text{ is a partial path profile for } \Delta \backslash [k-1]\}.$$

The goal of this subsection is to construct an IDNE of Γ^{N}, which proves the following main result.

Theorem 1. *Every normal-form game Γ^{N} admits an IDNE.*

Before going into details, it is worth noting that Definition 2 directly implies that every IDNE possesses the following properties, which are crucial for studying SPE in Section 5.2.

- No overtaking: If agents i and j enter G from the same origin, but i does so earlier than j, and they both pass through some vertex v under the NE, then i reaches and leaves v no later than j does. The property in the special case of $v = d$ is known as *Global FIFO*.
- Earliest arrival: Given the other agents' choices in the NE, each agent using his path in the NE reaches each vertex on the path (not only the destination d) at an earliest time among all of his possible choices.
- Sequential independence: For each $k \geq 1$, if all the agents with iterative dominant orders at most k fix their paths as in the NE, then their arrival times at all vertices along their paths (including destination d) are independent of the choices of all the agents with indices larger than k.

To better understand our construction of an IDNE, we first give an example to illustrate this solution concept.

Example 2. Consider game Γ^{N} with input network illustrated in Figure 2, where at vertices y_1, y_2 and d, the right, left and upper edges have higher priorities, respectively (i.e., $x_1y_1 \prec_{y_1} o_1y_1$,

$o_2y_2 \prec_{y_2} x_2y_2$, and $y_1d \prec_d y_2d$). At time 1, seven agents (represented by small rectangles beside the corresponding origins) are about to enter the network from origins o_1, o_2, o_g, o_h and o_i. Agents 1–4 each have a unique path to choose, and agents g, h, i each have two choices — upper and lower paths. An iterative dominant order of the agents is $(1, 2, 3, 4, g, h, i)$. The associated dominant paths for the first four agents are their unique paths, and those for the last three agents are their upper path $o_gv_1x_1y_1d$, lower path $o_ho_2y_2d$, and upper path $o_iu_1o_1y_1d$, respectively.

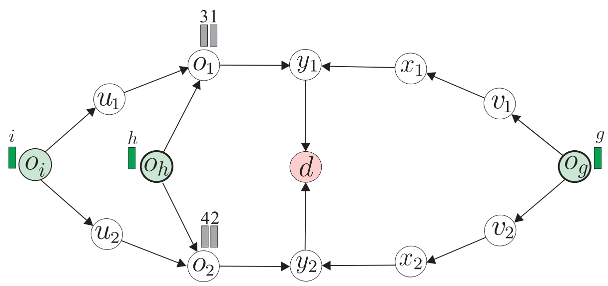


Figure 2 The existence of an IDNE for game Γ^{N} with multiple origins

Obviously, the first four agents are as indexed. We next show that g is the fifth agent associated with his upper path. Assuming agents 1–4 follow their trivial dominant routes, it is clear that the earliest possible time an agent in $\{g, h, i\}$ can reach the rth vertex of g's upper path is time r ($r = 1, \ldots, 5$). Moreover, no matter what routes agents h and i take, by following his upper path, agent g clearly reaches the first four vertices o_g, v_1, x_1, y_1 on his path at the earliest possible times $1, 2, 3, 4$, and subsequently reaches d at time 5 as desired (this is because his coming edge at y_1 has a higher priority, implying that agents h and i cannot overtake him and make his arrival time at d later than 5). Thus reindexing agent g as agent 5 does satisfy the condition for an IDNE. Now given that agents 1–5 follow their dominant paths, no matter how agent i routes, by following his lower path, agent h reaches o_h, o_2, y_2, d on the path at times $1, 2, 4$ and 5, each of which is the earliest possible that agent h or i can achieve. Thus agent h associated with his lower path is qualified to be agent 6 in the order. Finally, agent i is the last one in the order; by using his upper path, he can reach all vertices o_i, u_1, o_1, y_1, d on the path at his earliest possible times $1, 2, 3, 4, 6$, given the dominant path choices of others.

From the above example, we see that there may be multiple IDNEs — neither the iterative dominant order nor the dominant path profile is unique. Specifically, the order between agents 1 and 2 (as well as between agents 3 and 4 and between agents g and h) can be swapped, and in any case agent i's either path can be his dominant path.

We briefly describe the idea of how to make full use of edge priorities (e.g., edge priority w.r.t. the destination d in Example 2 that has been ignored in our previous discussion) to pin down a special IDNE (a unique iterative dominant order combined with a unique dominant path profile), which always exists and hence proves Theorem 1.

Suppose now the first $k-1$ agents as well as their associated dominant paths have been determined. We are to identify the kth agent, whom we relabel as k, and his associated dominant path $P_k \in \mathscr{P}_k$. Define the "ideal arrival time" at any vertex for each of the remaining agents as the earliest time when this agent can reach the vertex, under the assumption that all other agents in the network are the identified $k-1$ ones, who follow their associated dominant paths previously determined. This ideal arrival time is defined as infinity if the vertex is unreachable by this agent. In the following, we will first choose a set $C = C(k)$ of candidate pairs (j, P_j) with $j \in \Delta\backslash[k-1]$ and $P_j \in \mathscr{P}_j$, and then prune C by backtracking the path P_j of one of the candidate pairs, starting from d edge by edge, and eliminating unqualified candidates discovered during the process, until only one candidate pair is left. The corresponding agent and path are thus identified as the kth agent and his dominant path, respectively.

A more detailed pruning process goes as follows. Initially, pair (j, P_j) is a candidate in C if and only if j is one of the remaining unidentified agents and $P_j \in \mathscr{P}_j$, i.e., P_j is a path from his starting vertex to d. Let $u = d$ and proceed with the following three steps in sequence:

(S1) A candidate $(j, P_j) \in C$ is retained in C if and only if the ideal arrival time of agent j at u is the earliest among all candidate agents in the current C, and P_j is a path along which j achieves this ideal arrival time at u;

(S2) A candidate $(j, P_j) \in C$ is retained in C if and only if the incoming edge to u on P_j has the highest edge priority among all candidate paths in the current C;

(S3) If there are more than one candidate left in the current C, then the candidate paths in the current C must share the same incoming edge e to u (whose tail vertex is denoted as u_e) and we backtrack along e: update u with u_e and go back to step (S1).

It can be seen that either the above process is terminated at some step when only one candidate is left in C, in which case we are done, or all agents corresponding to the current candidate pairs are in the same initial queue or enter G simultaneously at the same origin (thus with the same candidate path). In this case, we identify among all the candidates the one, (j, P_j), such that agent j has the highest initial queue rank or original rank among all candidate agents in the current C.

It turns out that the path profile $(P_k)_{k\in\Delta}$ constructed as above is indeed an IDNE. We call it a special IDNE. As an illustration, the order $(1, 2, \ldots, 7)$ of agents and their dominant paths given in Example 2 constitute the special IDNE. For example, the order between agents 1 and 2 (resp. 3 and 4) is determined by the edge priorities w.r.t. d; the order between agents 2 and 3 (resp. 4 and $g\,(=5)$) is determined by the arrival times at d; the order between $g\,(=5)$ and $h\,(=6)$ is determined by backtracking from d to y_1 and checking the edge priorities at y_1. The precise algorithmic description for the aforementioned process together with a formal proof of Theorem 1 is presented in Section EC.2 of the Electronic Companion.

We would like to remark that placing priorities on edges is crucial to the NE existence of the game model Γ^{N} with multiple origins. Example 3 below shows that, if priorities were placed on agents (i.e., when two agents enter an edge at the same time, the agent with a higher priority will be ranked higher in the queue), one could not guarantee the NE existence when there are more than one origins, though an NE does exist in the single-origin single-destination case (Scarsini et al. 2018). A more detailed discussion about why this critical tie-breaking rule matters is provided in Section EC.11 of the Electronic Companion.

Example 3. Consider an example modified from Figure 2, where global priorities are placed on the agents in a way that i ranks higher than g and g higher than h. Agents i, g, h are our focus, as the remaining four agents do not make substantial decisions, and are not affected by i, g, h. The lowest ranked agent h reaches o_1 (or o_2) one time unit earlier than the highest ranked agent i if they choose to pass the same vertex o_1 (or o_2); otherwise, they reach y_1, y_2 at the same time as agent g reaches y_1 or y_2. It can be verified that h, i and g have a Rock-Paper-Scissors-like relationship, and hence this game does not have any NE.

4.2. Braess-like paradoxes

A well-known phenomenon in selfish routing is the Braess's paradox: building a new road may make the network more congested. Recently, Scarsini et al. (2018) discovered a Braess-like paradox under their model of atomic dynamic flows: removing an initial queue may reduce the system performance. This paradox occurs in a single-origin single-destination extension-parallel network, which has been known to be free of the classical Braess's paradox based on latency functions. An apparent cause is that removing an initial queue may bring about route changes of agents, leading them to a less efficient NE (despite of the presence of a more efficient one). This type of paradox is also present in our model as shown in Section EC.12 of the Electronic Companion by an adaptation from the example of Scarsini et al. (2018).

The following example demonstrates a different paradoxical phenomenon in the sense that it stems from unpredictable chain effects of agents' interactions. In our example, no route changes are involved.

Example 4. Consider an instance of game Γ^{N} as depicted in Figure 3. There are a total of ten agents (shown as small rectangles on edges), with agents 1, 2 and 3 being our focus. Figure 3 shows

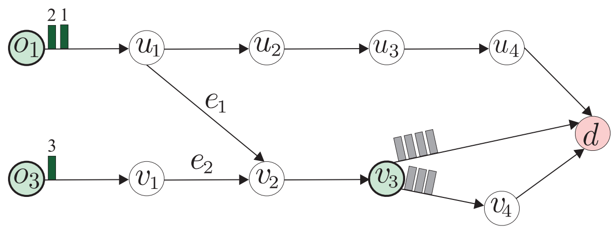


Figure 3 Removal of agent 1 from the system weakly harms other agents

the locations of agents at time 1. Edge e_1 has a higher priority than e_2. Suppose that agent 1 chooses the top path $o_1u_1u_2u_3u_4d$, agent 2 chooses the middle path $o_1u_1v_2v_3d$, agent 3 follows the bottom path $o_3v_1v_2v_3v_4d$, and other seven agents follow their trivial paths. It can be checked that all the three agents 1, 2, 3 reach destination d at time 6.

Now let us remove agent 1 from the game and suppose that all other agents keep their paths as above. Removal of agent 1 makes agent 2 arrive at vertices u_2, v_3 and v_4 one time unit earlier. Since agent 2 has to spend one extra unit of waiting time at edge v_3d, he reaches d still at time 6. However, agent 2's earlier arrival at vertex v_2 delays agent 3, making him reach d at time 7. Note that both path profiles in the above two scenarios are NEs of the corresponding games.

A macro-level explanation for the above counterintuitive example is that, when an agent disappears, some agents may benefit *temporarily* in that they enter some edges earlier; however, they have to spend more time waiting at some of these edges. As a result, their arrival times at the destination are not affected at all, but their earlier entries into some edges may add *everlasting* delays to some other agents who go through the same edges.

In studying NE properties, we need to frequently analyze what happens if one agent unilaterally deviates by choosing a different path. As demonstrated in the above example, this is a quite tricky issue in general. Agents may affect one another in unpredictable ways due to the intricate chains of interactions. Despite this complication, we are able to show in the remainder of this section that the NEs in our model possess many desirable properties.

4.3. NE characterization

In this subsection, we characterize all NEs for game Γ^{N}. The characterization not only shows that a general NE bears many similarities to the IDNEs discussed in Section 4.1, but also helps us establish a close connection between a game of nonadaptive agents and that of adaptive agents (see Section 5.3).

Batching agents according to their arrival times at the destination d is useful for our analysis on the NEs. For any path profile $\boldsymbol{q}$ of Γ^{N}, let $\tau(\boldsymbol{q},1)<\tau(\boldsymbol{q},2)<\tau(\boldsymbol{q},3)<\cdots$ be the arrival times of all agents at d under $\boldsymbol{q}$. For each integer $k\geq 1$, let

$$\Delta(\boldsymbol{q},k):=\{i\in\Delta\,|\,t_i^d(\boldsymbol{q})=\tau(\boldsymbol{q},k)\}$$

denote the set of agents in Γ^{N} who reach d under $\boldsymbol{q}$ at the kth earliest time $\tau(\boldsymbol{q},k)$; we often refer to $\Delta(\boldsymbol{q},k)$ as the *kth batch*. We use

$$\Delta(\boldsymbol{q},[k]):=\cup_{j\in[k]}\Delta(\boldsymbol{q},j)$$

to denote the set of agents reaching d no later than time $\tau(\boldsymbol{q},k)$, i.e., those in the first k batches. For notational convenience, we set $\Delta(\boldsymbol{q},[0]):=\emptyset$ to be the 0th batch, and let $\Delta(\boldsymbol{q},[\infty]):=\Delta$ denote the disjoint union of all batches.

It can be shown that the interactions between agents of different batches at an NE are hierarchal. That is, every NE is *iteratively batch-dominant* in that there is no way for agents in a later batch (no matter how they coordinate) to affect any agent in an earlier batch, provided all earlier agents follow their routes in the NE. This iterative batch-dominance, formally defined below, actually characterizes all NEs of game Γ^{N}.

Definition 3 (Iterative batch-dominance). A path profile $\boldsymbol{q}=(Q_h)_{h\in\Delta}$ of Γ^{N} is *iteratively batch-dominant* if, for any batch index $k\geq 1$, agent $i\in\Omega:=\Delta(\boldsymbol{q},[k])$, vertex $v\in Q_i$, agent $j\in\Delta\backslash\Omega$, and partial path profile $\boldsymbol{r}_{-\Omega}$ for agents in $\Delta\backslash\Omega$, the following inequalities hold:

$$t_i^v(\boldsymbol{q})=t_i^v(\boldsymbol{q}_\Omega,\boldsymbol{r}_{-\Omega})\leq t_j^v(\boldsymbol{q}_\Omega,\boldsymbol{r}_{-\Omega})\quad\text{and}\quad t_j^d(\boldsymbol{q}_\Omega,\boldsymbol{r}_{-\Omega})\geq\tau(\boldsymbol{q},k+1)>t_i^d(\boldsymbol{q}_\Omega,\boldsymbol{r}_{-\Omega}).$$

Theorem 2. *A path profile is an NE for game Γ^{N} if and only if it is iteratively batch-dominant.*

Using Theorem 2, we can establish that all NEs are strong NEs (see definition below) and global FIFO (see Theorem EC.6).

Definition 4 (Strong NE). A path profile $\boldsymbol{p}$ of Δ is a *strong NE* of Γ^{N} if no group of agents can gain by deviation, i.e., there exists no group $S\subseteq\Delta$ and partial path profile $\boldsymbol{p}'_S$ of agents in S such that $t_i^d(\boldsymbol{p}'_S,\boldsymbol{p}_{-S})<t_i^d(\boldsymbol{p})$ for all $i\in S$, where $\boldsymbol{p}_{-S}$ is the partial path profile (determined by $\boldsymbol{p}$) of agents not in S.

Theorem 3. *All NEs of every game Γ^{N} are strong NEs and global FIFO.*

Since each strong NE is also an NE, Theorem 3 actually establishes the equivalence between an NE and a strong NE in our model. Note that the strong NE property implies that every NE of Γ^{N} is weakly Pareto optimal. Moreover, it follows from Theorem 2 that every NE of game Γ^{N} has a hierarchal structure that resembles the sequential structure of IDNEs. More specifically, we have the following properties:

- Every NE is *hierarchically independent* in that, for every $k \geq 1$, if agents in the first k batches all follow their NE routes, then their arrival times at any vertex are independent of other agents' choices.
- Every NE is *hierarchically optimal* in that, for every $k \geq 1$, the arrival time of each agent in the kth batch under the NE is the smallest among the arrival times of all agents outside the first $k-1$ batches under any routing in which agents in the first $k-1$ batches follow their NE routes.

Two other properties of the IDNEs, no overtaking and earliest arrival, do not hold in general for the NEs of game Γ^{N} (see Section EC.12 for examples). This is in contrast to nonatomic dynamic flow games for which every NE must be no overtaking and earliest arrival (Koch and Skutella 2011). On the other hand, every NE of game Γ^{N} is *temporally overtaking* in that, if agent i enters the network G earlier than j from the same origin, but j overtakes i at some vertex $v \in V \backslash \{d\}$ (i.e., j reaches v earlier than i), then they must reach the destination d at the same time. Omitted proofs and more properties possessed by the NEs of game Γ^{N} are presented in Section EC.7 of the Electronic Companion.

4.4. Computations

In this subsection we show that a best response and an NE of game Γ^{N} can be computed efficiently. The computation works with a kind of "naive" greedy idea, which is validated using the notion of preemption. Throughout this subsection, i denotes a fixed agent and $\boldsymbol{q}_{-i} = (Q_j)_{j \in \Delta \backslash \{i\}}$ a partial path profile of all other agents. We consider the scenario where only agent i is allowed to change his path and the others always follow $\boldsymbol{q}_{-i}$.

4.4.1. Preempt relations We say that agent i *preempts* agent j at vertex v if either (i) the earliest time i reaches v is earlier than the earliest time j reaches v (among all path profiles in which all agents but i adopt the same paths as in $\boldsymbol{q}_{-i}$), or (ii) their earliest times are the same and additionally, either (ii.a) agent i can reach v via an edge that has a higher priority (w.r.t. v) than an edge that j uses to reach v, or (ii.b) vertex v is agent i's but not j's starting vertex, or (ii.c) vertex v is the starting vertex of both agents i and j, and i has a higher initial queue rank or original rank than j. This notion of preemption (see Definition EC.1 in Section EC.4 for a formal description) combines the *optimization* (minimization) on arrival times and the *speciality* on the best available choice w.r.t. edge priorities.

The preempt relation, together with the following lemma, plays a critical role in designing our algorithm for computing best responses.

Lemma 1. *For any agent $j \in \Delta \backslash \{i\}$ and vertex $v \in Q_j$, if there exist paths $P_i, P_i' \in \mathscr{P}_i$ such that $t_j^v(P_i, \boldsymbol{q}_{-i}) \neq t_j^v(P_i', \boldsymbol{q}_{-i})$, then i preempts j at v under $\boldsymbol{q}_{-i}$.*

The above lemma implies in particular that if agent i's unilateral change from P_i to P_i' can affect the arrival time of agent j at vertex v, making it earlier or later, then by using some path in $\mathscr{P}_i$ (not necessarily P_i or P_i'), agent i is able to reach v no later than j under $(P_i, \boldsymbol{q}_{-i})$ and under $(P_i', \boldsymbol{q}_{-i})$. In contrast, this property does not necessarily hold in the model where ties are broken using agent priorities (see Remark EC.1 in Section EC.4). Furthermore, we can show that if agent i preempts j at vertex $v \in Q_j$, then i preempts j at all vertices on the subpath of Q_j from v to d (see Corollary EC.1 in Section EC.4).

Lemma 1 enables us to classify all agents but i into two categories, the "slow" ones $\mathcal{S}$ whom agent i can preempt and the "fast" ones $\mathcal{F}$ whom agent i cannot preempt.

(C1) Agents of $\mathcal{F}$ are always no later than agent i at any vertex along their paths (regardless of the choice of i). Hence the flows resulting from the travels of $\mathcal{F}$ agents can be viewed as an exogenous environment for i.

(C2) In contrast, when agent i follows a special optimal path (denoted O_i^*), whose existence can be deduced from Lemma 1, he reaches each vertex of a final segment of O_i^* no later than any agent of $\mathcal{S}$ under path profile $(P_i, \boldsymbol{q}_{-i})$ for any $P_i \in \mathscr{P}_i$, which results in that agent i is "faster" than all agents of $\mathcal{S}$ in the final segment. To put it differently, no agent of $\mathcal{S}$ can influence i in the final segment when he follows path O_i^*, which attains the optimality w.r.t. the exogenous environment of $\mathcal{F}$ agents and possesses the speciality w.r.t. edge priorities.

In summary, when agent i follows the special optimal path, the intricate chain effects are decoupled in the above sense and our analysis is greatly alleviated. We remark that Lemma 1 is also an important tool for us to establish the NE characterization discussed in Section 4.3.

4.4.2. Computing best responses When we talk about algorithm efficiency, only the case of finite agent set and finite network is concerned unless otherwise stated. Given a game Γ^{N} with agent set Δ, an agent $i \in \Delta$, and a partial path profile $\boldsymbol{q}_{-i} = (Q_j)_{j \in \Delta \setminus \{i\}}$, our algorithm computes a special best response of i to $\boldsymbol{q}_{-i}$ defined as follows.

Definition 5 (EE best-response). Given a partial path profile $\boldsymbol{q}_{-i}$, a path $Q_i^* \in \mathscr{P}_i$ of agent i is called the *edge-priority-oriented earliest-arrival best-response* (EE best-response) if for each non-starting vertex v of Q_i^*,

- Agent i's arrival time at v when he goes along Q_i^* is the earliest he can achieve;
- The incoming edge to v on Q_i^* has the highest priority among the incoming edges to v on all paths of $\mathscr{P}_i$ along each of which i reaches v at the earliest time.

By definition, agent i has a *unique* EE best-response to any given $\boldsymbol{q}_{-i}$. It can be seen that the EE best-response is exactly the special optimal path Q_i^* discussed in (C2) in Section 4.4.1. Our algorithm for finding Q_i^* resembles the classical Dijkstra algorithm for computing a shortest path. However, its correctness proof is nontrivial.

Definition 6. For each edge $e \in E$ and time $r \geq 0$, let $\mathcal{Q}_e^r$ denote its queue at time r produced by routing $\boldsymbol{q}_{-i} = (Q_j)_{j \in \Delta \setminus \{i\}}$. For any edge e' (if any) incoming to the tail vertex of e, let $\mathcal{Q}_{e,e'}^r$ denote the subset of agents in $\mathcal{Q}_e^r$ who enter e at r from edges with priorities no higher than e'.

We slightly abuse notation $\mathcal{Q}_{e,e'}^r$ in the following two settings: (i) If i is in the initial queue $\mathbb{Q}_e^0$ with $e = v_1 v_2$, we abuse $\mathcal{Q}_{e,\hat{e}_{v_1}}^0$ to denote the set of agents in $\mathbb{Q}_e^0$ who queue after i. (ii) If $i \in \Delta_{v_1,r}$ with some $r \geq 1$, then for any edge $e \in E$ with tail vertex v_1, we abuse $\mathcal{Q}_{e,\hat{e}_{v_1}}^r$ to denote the set of agents in $\mathcal{Q}_e^r$ who either enter e at time r from edges incoming to v_1, or belong to $\Delta_{v_1,r}$ and have original ranks lower than i.

Note that $\mathcal{Q}_e^0 = \mathbb{Q}_e^0 \setminus \{i\}$ for all $e \in E$. Given $v \in V$, let τ^v denote the earliest time when agent i can reach v provided other agents follow $\boldsymbol{q}_{-i}$. Let $Y := \{v \in V | \tau^v < +\infty\}$ denote the set of vertices in G that i can reach through some paths, i.e., i's reachable vertices. In particular, if i is in the initial queue $\mathbb{Q}_e^0$, the tail vertex of e belongs to Y. Since G is acyclic, one can find in polynomial time a complete "acyclic" order on the vertices in Y such that for each edge with both end-vertices in Y, its tail vertex has an order smaller than its head vertex. Let $v_1, v_2, \ldots, v_{|Y|}$ be the vertices in Y ranked by such an order. Then it must be the case that $v_{|Y|} = d$, and v_1 is i's starting vertex, i.e., either $i \in \mathbb{Q}_{v_1 v_2}^0$ or i enters G from v_1 at some time $r \geq 1$. For any vertex $v \in Y \setminus \{v_1\}$, let $\hat{e}_v$ denote the incoming edge to v with the highest priority (w.r.t. $\prec_v$) that agent i can use to reach v at time τ^v, provided $\boldsymbol{q}_{-i}$ is fixed. Now we are ready to describe our Dijkstra-like algorithm.

Algorithm 1 (Dijkstra-like algorithm for EE best-response)

1. `Simulation of the dynamic process generated by` $\boldsymbol{q}_{-i}$: for every time $r \geq 0$ when the network is nonempty, for every edge e and any edge e' incoming to the tail vertex of e, compute the queues $\mathcal{Q}_e^r$ and $\mathcal{Q}_{e,e'}^r$.
2. `Initiation:` If $i \in \mathbb{Q}_{v_1 v_2}^0$, then $\tau^{v_1} \leftarrow 0$; If $i \in \Delta_{v_1,r}$ for some $r \geq 1$, then $\tau^{v_1} \leftarrow r$.
3. $k \leftarrow 2$, $E_i^* \leftarrow \emptyset$
4. **While** $k \leq |Y|$ **Do**
   - $\tau^{v_k} \leftarrow \min_{e = v_h v_k \in E : h < k} \left\{ \tau^{v_h} + |\mathcal{Q}_e^{\tau^{v_h}} \setminus \mathcal{Q}_{e,\hat{e}_{v_h}}^{\tau^{v_h}}| + 1 \right\}$.
   - $\hat{e}_{v_k} \leftarrow$ the edge in $\operatorname{argmin}_{e = v_h v_k \in E : h < k} \left\{ \tau^{v_h} + |\mathcal{Q}_e^{\tau^{v_h}} \setminus \mathcal{Q}_{e,\hat{e}_{v_h}}^{\tau^{v_h}}| + 1 \right\}$ that has the highest priority
   - $E_i^* \leftarrow E_i^* \cup \{\hat{e}_{v_k}\}$, $k \leftarrow k + 1$

   End-While
5. `Output:` Return i's EE best-response, i.e., the unique path from v_1 to d that can be formed by some edges in E_i^*.

For $k = 1, 2, \ldots, |Y|$, in the kth iteration of the while-loop, the algorithm computes agent i's earliest arrival time τ^{v_k} at vertex v_k in the same spirit as the Dijkstra algorithm. If agent i uses edge

$e = v_h v_k$ to travel from v_h to v_k, the fastest way is that he reaches v_h at the earliest possible time τ^{v_h} (which has been derived in a previous iteration), waits at e for $|\mathcal{Q}_e^{\tau^{v_h}} \setminus \mathcal{Q}_{e,\hat{e}_{v_h}}^{\tau^{v_h}}|$ time units, and then spends 1 unit of transit time going through e to v_k. Thus τ^{v_k} is obtained by taking minimum over all possible edges e incoming to v_k, as stated in the first item of Step 4. The nontrivial part of our algorithm is determining the queuing time to be $|\mathcal{Q}_e^{\tau^{v_h}} \setminus \mathcal{Q}_{e,\hat{e}_{v_h}}^{\tau^{v_h}}|$. Among the agents who queue at edge e at time τ^{v_h} (i.e., those in $\mathcal{Q}_e^{\tau^{v_h}}$), the ones whom i can overtake at e are those in $\mathcal{Q}_{e,\hat{e}_{v_h}}^{\tau^{v_h}}$ and they are exactly the agents in $\mathcal{Q}_e^{\tau^{v_h}}$ who are preempted by i at v_h.

THEOREM 4. *Given any partial path profile* $\boldsymbol{q}_{-i}$, *the EE best-response of agent* i *can be computed by the Dijkstra-like algorithm efficiently.*

The correctness proof of the algorithm can be found in Section EC.5. We discuss here the time efficiency. To compute the queues in Step 1, we simulate the transit and queuing process in a way that we only keep records for all nonempty queues during the process. Note that the queue state varies only when an agent just reaches an edge or just leaves a queue. So the number of records contributed by an agent is at most twice the number of edges on his path. It follows that all queues stated in Step 1 can be found in polynomial time. On the other hand, the while-loop at Step 4 is a standard dynamic program and its time complexity is $O(|V|^2)$. Therefore, the EE best-response of each agent is polynomially computable if the agent set and network are finite. Otherwise, the computational efficiency is achieved via ignoring agents who are sufficiently far from the destination d or enter the network at time points sufficiently later (i.e., those who are doubtlessly not in $\mathcal{F}$).

EXAMPLE 5. Let us reconsider the game shown in Example 2. Now suppose agent h chooses his upper path $o_h o_1 y_1 d$ and agent g chooses his lower path $o_g v_2 x_2 y_2 d$, while agents 1–4 just move forward along their unique paths. We illustrate how to compute agent i's EE best-response using the Dijkstra-like algorithm. First, by simulating the flow produced by $\boldsymbol{q}_{-i}$, we have the critical queue-size information in Table 1.

	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
$\lvert\mathcal{Q}^r_{o_1y_1}\rvert$	2	2	1	0	0
$\lvert\mathcal{Q}^r_{o_2y_2}\rvert$	2	1	0	0	0
$\lvert\mathcal{Q}^r_{y_1d}\rvert$	0	1	1	1	0
$\lvert\mathcal{Q}^r_{y_1d,o_1y_1}\rvert$	0	1	1	0	0
$\lvert\mathcal{Q}^r_{y_2d}\rvert$	0	1	1	1	0
$\lvert\mathcal{Q}^r_{y_2d,o_2y_2}\rvert$	0	1	1	1	0

Table 1 Critical queue-size information on $\boldsymbol{q}_{-i}$

It is apparent that $(o_i, u_1, u_2, o_1, o_2, y_1, y_2, d)$ is a complete acyclic order of agent i's reachable vertices. Using the above queue-size information, we can compute the earliest arrival time at each vertex for agent i as follows: $\tau^{o_i}=1,\tau^{u_1}=2,\tau^{u_2}=2,\tau^{o_1}=3,\tau^{o_2}=3,\tau^{y_1}=5,\tau^{y_2}=4$ and $\tau^d=\min\left\{\tau^{y_1}+\left|\mathcal{Q}^5_{y_1d}\backslash\mathcal{Q}^5_{y_1d,o_1y_1}\right|+1,\tau^{y_2}+\left|\mathcal{Q}^4_{y_2d}\backslash\mathcal{Q}^4_{y_2d,o_2y_2}\right|+1\right\}=\min\{5+0+1,4+(1-1)+1\}=5$, $\hat{e}_d=y_2d$. Thus agent i's EE best-response is his lower path $o_iu_2o_2y_2d$.

4.4.3. Computing the special IDNE In Section 4.1, we have constructed the special IDNE for game Γ^{N}. Now let us show that this construction can be executed efficiently. It suffices to show that, when the partial path profile $(P_1,\dots,P_{i-1})$ for agents $1,\dots,i-1$ has been computed, we can efficiently find the next agent, whom we label as i, and his associated dominant path P_i.

It is worth noting that P_i is actually the EE best-response of i to $(P_1,\dots,P_{i-1})$. Therefore, to identify the agent i, we employ the Dijkstra-like algorithm to compute the EE best-response P_j of each agent $j\in\Delta\backslash[i-1]$ to $(P_1,\dots,P_{i-1})$ and their earliest arrival times at each vertex. (Note that when we simulate the flow generated by $(P_1,\dots,P_{i-1})$ for every agent $j\in\Delta\backslash[i-1]$, the subsets of agents preempted by j are always empty, since no agent in $\Delta\backslash[i-1]$ can affect the agents in $[i-1]$ according to the iterative dominance property, i.e., there is no intricate chain effects under this circumstance.) Starting with a candidate set $\{(j,P_j):j\in\Delta\backslash[i-1]\}$, we repeatedly implement steps (S1)–(S3) in Section 4.1 to prune the set until only one candidate is left. This candidate consists of the desired agent i and path P_i. Therefore, the total number of times we run the Dijkstra-like algorithm is $\sum_{i=1}^{|\Delta|}(|\Delta|-(i-1))=(1+|\Delta|)|\Delta|/2$. As mentioned earlier, if infinitely many agents are involved, our computation may ignore agents whose entry times to G are sufficiently late.

We remak that there is another natural algorithm to efficiently compute the special IDNE by making the utmost of the above EE best-responses and the iterative dominance property. Given an arbitrary initial path profile $\boldsymbol{q}^{(0)}=(Q_i^{(0)})_{i\in\Delta}$ with $Q_i^{(0)}\in\mathscr{P}_i$, define a sequence of path profiles $\boldsymbol{q}^{(k)}=(Q_i^{(k)})_{i\in\Delta}$, $k=1,2,\dots,|\Delta|$, where $Q_i^{(k)}$ is agent i's EE best-response to $\boldsymbol{q}_{-i}^{(k-1)}$, i.e., at each round k, every agent makes EE best-response to other agents' strategies in the preceding round. For our game Γ^{N}, using this iterative approach of simultaneous EE best-responses, the path profiles converge to the special IDNE quickly (in at most $|\Delta|$ rounds). To see the convergence, note that regardless of the initial paths of other agents, the EE best-response of agent 1 in the first round must be exactly his path in the special IDNE, which will never change in subsequent rounds. Similar observations are applicable to the EE best-response of agent 2 from the second round onwards, and so on and so forth. Finally, in the $|\Delta|$th round, all agents choose the paths as in the special IDNE, where everyone's path is his EE best-response to others. We illustrate the algorithm using a simple example as follows.

Example 6. Following Example 5, suppose $(o_iu_2o_2y_2d,o_ho_1y_1d,o_gv_2x_2y_2d)$ is an initial partial path profile for agents i,h and g, respectively, and the other four agents follow their unique paths. Table 2 lists the EE best-responses of agents i,h,g in each round and the convergence process. Note that

	Agent i	Agent h	Agent g
Round 0	$o_iu_2o_2y_2d$	$o_ho_1y_1d$	$o_gv_2x_2y_2d$
Round 1	$o_iu_2o_2y_2d$	$o_ho_1y_1d$	$\boldsymbol{o_gv_1x_1y_1d}$
Round 2	$o_iu_2o_2y_2d$	$\boldsymbol{o_ho_2y_2d}$	$\boldsymbol{o_gv_1x_1y_1d}$
Round 3	$\boldsymbol{o_iu_1o_1y_1d}$	$\boldsymbol{o_ho_2y_2d}$	$\boldsymbol{o_gv_1x_1y_1d}$
Round 4	$\boldsymbol{o_iu_1o_1y_1d}$	$\boldsymbol{o_ho_2y_2d}$	$\boldsymbol{o_gv_1x_1y_1d}$

Table 2 The iterative process of simultaneous EE best-responses

$(\boldsymbol{o_gv_1x_1y_1d}, \boldsymbol{o_ho_2y_2d}, \boldsymbol{o_iu_1o_1y_1d})$ is the partial path profile for agents g, h, i in the special IDNE for this example.

Note that, in the above process, we do not need to identify the iterative dominant order of the agents. It resembles a natural learning process in the real world, where agents update their strategies in a distributed way from an arbitrary initial path profile. This kind of algorithms are quite common in the study of day-to-day models, where convergence within a finite number of steps is rare and even convergence may not be guaranteed (Guo et al. 2018).

5. The game of adaptive agents

Our second submodel Γ^{A} of dynamic atomic flow game assumes that agents are of a relatively high rationality level. Specifically, agents are adaptive in that they make routing decisions at *every* nonterminal vertex they reach as to which next edge to take. Their decisions at a vertex may depend on the choices of other agents in the history. The following example demonstrates that it is natural to assume that agents use adaptive strategies, when they have updated information about others, and they may gain by using more flexible adaptive strategies than simply choosing fixed origin-destination paths at the very beginning.

Example 7. Consider the network in Figure 4, where e_1 has a higher priority over e_2, and e_3 over e_4. Two agents 1 and 2 set off from their respective origins o_1 and o_2.

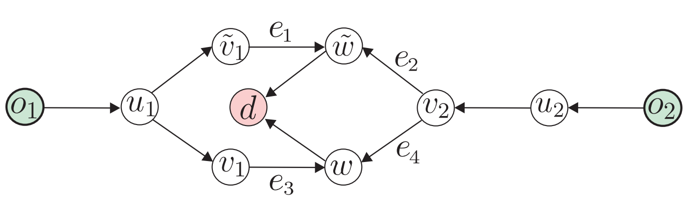


Figure 4 Nonadaptive vs. adaptive agents

While agent 1 does not care about what agent 2 selects (because e_1 and e_3 have higher priorities over e_2 and e_4, respectively), agent 2 does care about what agent 1 selects, because he may be blocked and delayed by agent 1 at w or $\tilde{w}$. But how could agent 2 be sure that agent 1 will select the upper (or lower) path? Suppose now agent 2 postpones his decision making on vertex v_2 to the time he reaches it, then he will select e_4 if he observes that agent 1 has chosen the upper path and e_2 otherwise. In fact, this is exactly what adaptive agent 2 does in both SPEs of the game Γ^{A}.

5.1. Game setting

For the extensive-form game Γ^{A}, the notion of a strategy is much more complicated than for the normal-form game Γ^{N}. While a nonadaptive agent in Γ^{N} has only one decision point, at which he selects an origin-destination path, an adaptive agent in Γ^{A} typically has multiple decision points. On the other hand, while the choice of a nonadaptive agent is an origin-destination path, the choice of an adaptive agent at each decision point is an edge. A strategy of an adaptive agent is a "complete plan" that is responsive to all possible scenarios, i.e., a profile of decisions at all decision points. We next present a rigorous definition of a strategy in the extensive-form game Γ^{A} in terms of "configurations" and "histories".

Given time point $r \geq 0$, we use $\mathbb{Q}_e^r$ to denote the queue at edge e at time r, which will be considered as both a sequence of agents and the corresponding set. We call $\mathsf{c}_r = (\mathbb{Q}_e^r)_{e\in E}$ a *configuration w.r.t. time* r if $\mathbb{Q}_e^r \cap \mathbb{Q}_{e'}^r = \emptyset$ for different edges e and e'. In particular,

- Let $\mathsf{c}_0 = (\mathbb{Q}_e^0)_{e\in E}$ denote the unique *initial configuration* given by the input (see Section 3);
- Let $\Delta(\mathsf{c}_r) := (\cup_{e\in E}\mathbb{Q}_e^r) \cup (\cup_{v\in V}\Delta_{r+1,v})$ denote the set of agents involved in configuration c_r and inflows at time $r+1$;
- Let $\mathscr{D}(\mathsf{c}_r) := (\cup_{e\in E}\mathbb{Q}_e^r) \cup (\cup_{v\in V, s\geq r+1}\Delta_{s,v})$ denote the set of agents involved in configuration c_r and afterwards.

We say that configurations c_r and c_{r+1} are *consecutive* if c_{r+1} is reachable from c_r after one time unit under the given inflows and the edge-priority DQ rule (recalling Section 3). A precise definition of consecutiveness is provided in Section EC.8 of the Electronic Companion, using a notion of action profiles.

Definition 7 (History/Decision point). For each time point $r \geq 0$, a sequence of consecutive configurations $\mathsf{h}_r = (\mathsf{c}_0, \ldots, \mathsf{c}_r)$ starting from the initial configuration c_0 is called a *history* at time r. In particular, $\mathsf{h}_0 = (\mathsf{c}_0)$ is called the *initial history*. The set of all histories at time r is denoted as H_r. Each history h_r corresponds to a *decision point* of all agents in $\Delta(\mathsf{c}_r)$.

Definition 8 (Strategy). A *strategy* of agent $i \in \Delta$ is a mapping σ_i that maps each history $\mathsf{h}_r = (\mathsf{c}_0, \ldots, \mathsf{c}_r)$ with $i \in \Delta(\mathsf{c}_r)$ to $\sigma_i(\mathsf{h}_r)$ such that, based on c_r and the edge-priority DQ rule, either $\sigma_i(\mathsf{h}_r)$ is the "next" edge along which i travels (i.e., i could stand at its tail part at time $r+1$) or $\sigma_i(\mathsf{h}_r)$ is a null element when under c_r agent i will exit G at time $r+1$. The *strategy set* of agent i is denoted as Σ_i. A vector $\sigma = (\sigma_i)_{i\in\Delta}$ is called a *strategy profile* of Γ^{A}.

Note in the above definition that when an agent is not the head of a queue, the "next" edge he "chooses" must be the same edge he is queuing at, i.e., he waits for at least one more time unit.

REMARK 1. The number of decision points of an adaptive agent is generally much larger than the number of vertices he passes. Taking Example 1 as an illustration, each agent has 4 decision points before arriving at vertex w: 1 point at origin o, and 3 points at vertex v (corresponding to the opponent choosing ou_1, ou_2 and ov, respectively). Suppose agent 2 has decided to choose edge ou_1 at origin o. Agent 2 still needs to specify in his strategy the choices at vertex v in 3 different scenarios, even if he will never reach v when he follows the strategy. This is a remarkable difference between a strategy in an extensive-form game and a strategy in daily languages.

A strategy profile is an SPE if and only if each agent has no incentive to deviate from his strategy at any decision point, assuming that other agents do not deviate. A more rigorous definition is given in terms of "game tree" as follows. The *game tree* of Γ^{A} is a tree with nodes corresponding to Γ^{A}'s histories (i.e., decision points of agents). At each game tree node (history) $\mathsf{h}_r = (\mathsf{c}_0, \mathsf{c}_1, \ldots, \mathsf{c}_r)$, agents in $\Delta(\mathsf{c}_r)$ need to make their own decisions simultaneously, and the collection of these decisions forms their action profile, which leads to a new node (history) h_{r+1} as a child (continuation) of h_r. For each history $\mathsf{h}_r = (\mathsf{c}_0, \mathsf{c}_1, \ldots, \mathsf{c}_r)$, the subtree of the game tree rooted at h_r can be viewed as a separate game (with agent set $\mathscr{D}(\mathsf{c}_r)$ starting from c_r at time r), which is referred to as a *subgame* of Γ^{A}. A subtree is also called a subgame tree.

Given a strategy profile $\sigma = (\sigma_i)_{i\in\Delta}$ of Γ^{A}, the restriction of each strategy σ_i with $i \in \mathscr{D}(\mathsf{c}_r)$ to a subgame tree rooted at $\mathsf{h}_r = (\mathsf{c}_0, \mathsf{c}_1, \ldots, \mathsf{c}_r)$ is also a strategy of agent i in the corresponding subgame starting from h_r. All these restricted strategies form a strategy profile for the subgame. Under the routing *induced* by the strategy profile for the subgame, the time when agent $i \in \Delta(\mathsf{c}_r)$ exits G is denoted as $t_i(\sigma|\mathsf{h}_r)$.

DEFINITION 9 (SPE). A strategy profile $\sigma = (\sigma_i)_{i\in\Delta}$ is a *subgame perfect equilibrium* (SPE) of Γ^{A} if for any $r \geq 0$ and any history $\mathsf{h}_r \in \mathsf{H}_r$, $t_i(\sigma|\mathsf{h}_r) \leq t_i(\sigma_i', \sigma_{-i}|\mathsf{h}_r)$ holds for all $i \in \Delta(\mathsf{c}_r)$ and all $\sigma_i' \in \Sigma_i$ such that strategy profile (σ_i', σ_{-i}) still leads to history h_r, where σ_{-i} is the partial strategy profile of σ for agents in $\Delta\backslash\{i\}$.

5.2. SPE existence

The standard way to prove the existence of an SPE is by backward induction (and usually the one-deviation property). However, in game Γ^{A}, time horizon is typically infinite and more than one agent may move at each time step, hence the usual approach does not work here in general. In this subsection, we establish the SPE existence for Γ^{A} using a constructive approach.

THEOREM 5. *Each extensive-form game Γ^{A} admits an SPE.*

The basic idea for constructing an SPE of Γ^{A} is to assemble the special IDNEs of various game instances Γ^{N} that are associated with configurations of Γ^{A}. To be more specific, given a game instance Γ^{A} with input (G, Δ), every configuration c_r of the extensive-form game Γ^{A} is associated with a normal-form game instance under model Γ^{N}, denoted as $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$, which starts at time r on network G with initial queues $\mathsf{c}_r = (\mathbb{Q}_e^r)_{e\in E}$, and is played by nonadaptive agents in $\mathscr{D}(\mathsf{c}_r) = (\cup_{e\in E}\mathbb{Q}_e^r) \cup (\cup_{v\in V, s\geq r+1}\Delta_{s,v})$. Using the method in Section 4.1, we obtain the special IDNE of game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$. In our assembling, at every configuration c_r, the action that each agent of $\Delta(\mathsf{c}_r)$ takes in Γ^{A} is determined by his path in the special IDNE of $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$: just choose the first edge of the path when he is the head of the current queue or keep staying in the queue otherwise. It can be shown that such a strategy profile is an SPE for Γ^{A}. The proof (see Section EC.9) relies on the properties of these NEs for all games $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$: the iterative dominance as well as the sequential independence and optimality discussed in Section 4.1.

The realized paths of the SPE constructed above form a path profile that is exactly the special IDNE we construct for game $\Gamma^{\mathrm{N}}(\mathsf{c}_0)$. Hence, this special SPE possesses all the nice properties discussed in Section 4.1, and also satisfies the following additional properties:

- Markovian: the action each agent takes at each decision point depends *only* on the current configuration and its associated time r, *not* on earlier configurations in the history. (Note that different histories, which correspond to different decision points of agents, may lead to the same configuration w.r.t. the same time point.)
- Anonymous: the action each agent takes at each decision point does not depend on the identities of other agents.

As a corollary of the efficient computation of NEs discussed in Section 4.4.3, we can efficiently find the action profile at any history under this SPE of game Γ^{A} assembled from special IDNEs.

Remark 2. If priorities were placed on agents, then an SPE may not exist even if the network has only one single origin. This can be seen from Example 3, which can be considered as a part of some more complex single-origin network game (we omit the construction of the whole game).

5.3. NE realization from an SPE

Each strategy profile σ of game Γ^{A} induces a path profile, which is a strategy profile of the corresponding game Γ^{N}. Recall from Example 1 that an SPE of Γ^{A} does not necessarily induce an NE of Γ^{N}. A natural question arises: are all NEs of Γ^{N} inducible by SPEs of Γ^{A}? The answer is yes, as formally presented in the following theorem, whose technical proof, relegated to Section EC.10, relies on both the hierarchical independence of general NEs (see Section 4.3) and the iterative dominance of special NEs (see Section 4.1).

Theorem 6. *For every NE profile $\boldsymbol{p}$ of game Γ^{N}, there exists an SPE σ of game Γ^{A} such that the path profile induced by the initial history h_0 and σ is exactly $\boldsymbol{p}$.*

Combined with Example 1, the above theorem shows that the NE outcome set of Γ^{N} is typically a proper subset of the SPE outcome set of Γ^{A}, reaffirming the intuition that model Γ^{A} is more flexible than Γ^{N}. Since model Γ^{N} is relatively easier to study, also natural and more frequently analyzed in the literature, Theorem 6 can serve as a bridge between models Γ^{A} and Γ^{N}. Recall that in general game theory, each SPE is also an NE. Our result does not contradict this well-known result because strategies have different meanings in Γ^{N} and Γ^{A}.

In addition, Theorem 6 gives an alternative answer to the question of how an NE could be possible. This question is quite challenging both in the general game theory and in the transportation study. While there are standard (but not completely satisfactory) answers, including pre-communication and rational expectation (Sheffi 1985), to defend the relevance of the NE notion, we provide an alterative answer via allowing the adaptiveness of agents. We argue that, when agents are able to make more flexible adaptive decisions than the usual rigid ones addressed in the previous literature, NE outcomes have more chances to be realized (by SPE).

6. Concluding remarks

In this paper, we have proposed a new network game model of atomic dynamic flows for both adaptive and nonadaptive agents. Our model is arguably promising thanks to its many desirable properties, including the equilibrium existence, equivalence between NEs and strong NEs, global FIFO, and computational efficiency for finding equilibria and best responses, which stand in stark contrast with existing related negative results on atomic dynamic flow games. In particular, the equivalence between NEs and strong NEs has rarely been seen in atomic routing games, even in a static setting, where rather restrictive conditions are often needed to guarantee their equivalence (Holzman and Law-Yone 1997).

We now briefly discuss the generality of the Unit Assumption. Our unit-capacity assumption goes inevitably together with the lane priorities, which constitute part of the input network. Gluing parallel lanes together, all the derived results without regard to computational complexity hold for networks with arbitrary capacities. When the input size of lane priorities is polynomial in that of the capacitated network, e.g., left lanes having higher priorities than the right ones, the running times of our algorithms are polynomial in the network's input size plus the number of agents. Our unit-length assumption is merely for the sake of easy description. When an edge does not have unit length, one can subdivide it into unit-length edges. An agent waits at the original edge if and only if he does so at the resulting unit-length edge that has the same tail vertex as the original edge, no queue being built up at any other resulting unit-length edges. Clearly, all our theoretical results (on equilibrium existence and properties) are still valid when the unit-length assumption is dropped. Moreover, as our algorithms only record nonempty queues in simulating the transit and queuing process (see the discussion following Theorem 4 in Section 4.4.2), the efficient computation can also be guaranteed for networks with arbitrary edge lengths.

Our results may also help us better understand the connections and differences between DQ-based games of atomic dynamic flows and those of nonatomic ones. It is known that NE flows, earliest arrival flows and global FIFO flows are all identical in related nonatomic models (Koch and Skutella 2011). In our atomic model, however, earliest arrival flows are NE flows, which are in turn global FIFO flows, but neither of the other ways around is generally valid. In addition, no overtaking is valid for nonatomic NE flows (Koch and Skutella 2011), which is not the case in our atomic model.

Our results provide some managerial insights and policy implications. First, back to the real world, the tie-breaking rule used in our model, which is based on edge priorities, plays a complementary role to the traffic-light system in coordinating the traffic. (A difference between our model, as well as almost all known related ones, and the real world is that the cross conflicts mediated by traffic lights are not considered.) Our theoretical results indicate that this kind of system possesses more nice properties, and is thus more reasonable than the system based on agent priorities. Second, our Braess-like paradox (Example 4) indicates that, in certain extreme scenarios, fewer vehicles may lead to a worse equilibrium. This raises, at least theoretically, a challenge to traffic restriction policies practiced in many cities all over the world in various ways.

This paper is our first attempt to understand games of atomic dynamic flows, especially with the introduction of agents' flexibility of online decision making. Many interesting problems are widely open. For example, is there an upper bound on the SPE (or NE) queue lengths for general single-origin single-destination networks when the inflow rate is no more than the minimum capacity of an origin-destination cut? Does a long-run steady state exist when the inflow is constant or seasonal? How efficient is this steady state if it does exist? What if agents are allowed to choose their departure times? The queue notion used in our model is also referred to as point-queue in the traffic community, i.e., queues have no physical lengths. Spillback models (Daganzo 1998, Bressan and Nguyen 2015, Sering and Koch 2019) that consider the physical lengths of queues are also important future directions.

One drawback of our nonadaptive model is that agents make their decisions simultaneously at the very beginning, which is before they enter the network. Investigating a more realistic model which rests between the adaptive and nonadaptive models in that agents make their route-choice decisions when they enter the network, as has been assumed in many nonatomic models, is also meaningful. Exploring such issues will undoubtedly help us better understand games of atomic dynamic flows. As a positive step in this spirit of pursuit, we consider a hybrid model, suggested by an anonymous reviewer, of agents who are between adaptive and nonadaptive, in the sense that an agent contemplates at every nonterminal vertex switching paths with a given probability. The SPE existence result for our model of adaptive agents is still valid for this hybrid model (see Section EC.13 of the Electronic Companion for details).

References

Anshelevich, E. and Ukkusuri, S. (2009). Equilibria in dynamic selfish routing. In *International Symposium on Algorithmic Game Theory*, pages 171–182. Springer.

Bressan, A. and Han, K. (2013). Existence of optima and equilibria for traffic flow on networks. *Networks and Heterogeneous Media*, 8(3):627–648.

Bressan, A. and Nguyen, K. T. (2015). Optima and equilibria for traffic flow on networks with backward propagating queues. *Networks & Heterogeneous Media*, 10(4):717–748.

Cao, Z., Chen, B., Chen, X., and Wang, C. (2017). A network game of dynamic traffic. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, pages 695–696. ACM.

Cominetti, R., Correa, J., and Larré, O. (2015). Dynamic equilibria in fluid queueing networks. *Operations Research*, 63(1):21–34.

Cominetti, R., Correa, J., and Olver, N. (2017). Long term behavior of dynamic equilibria in fluid queuing networks. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 161–172. Springer.

Correa, J., de Jong, J., De Keijzer, B., and Uetz, M. (2019). The inefficiency of Nash and subgame perfect equilibria for network routing. *Mathematics of Operations Research*, 44(4):1286–1303.

Correa, J. R. and Stier-Moses, N. E. (2010). Wardrop equilibria. *Wiley Encyclopedia of Operations Research and Management Science*.

Daganzo, C. F. (1998). Queue spillovers in transportation networks with a route choice. *Transportation Science*, 32(1):3–11.

Graf, L. and Harks, T. (2019). Dynamic flows with adaptive route choice. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 219–232. Springer.

Guo, R.-Y., Yang, H., and Huang, H.-J. (2018). Are we really solving the dynamic traffic equilibrium problem with a departure time choice? *Transportation Science*, 52(3):603–620.

Hamdouch, Y., Marcotte, P., and Nguyen, S. (2004). A strategic model for dynamic traffic assignment. *Networks and Spatial Economics*, 4(3):291–315.

Han, K., Friesz, T. L., and Yao, T. (2013). Existence of simultaneous route and departure choice dynamic user equilibrium. *Transportation Research Part B: Methodological*, 53:17–30.

Harks, T., Peis, B., Schmand, D., Tauer, B., and Koch, L. V. (2018). Competitive packet routing with priority lists. *ACM Transactions on Economics and Computation (TEAC)*, 6(1):4.

Hendrickson, C. and Kocur, G. (1981). Schedule delay and departure time decisions in a deterministic model. *Transportation Science*, 15(1):62–77.

Hoefer, M., Mirrokni, V. S., Röglin, H., and Teng, S.-H. (2009). Competitive routing over time. In *International Workshop on Internet and Network Economics*, pages 18–29. Springer.

Hoefer, M., Mirrokni, V. S., Röglin, H., and Teng, S.-H. (2011). Competitive routing over time. *Theoretical Computer Science*, 412(39):5420–5432.

Holzman, R. and Law-Yone, N. (1997). Strong equilibrium in congestion games. *Games and Economic Behavior*, 21(1-2):85–101.

Ismaili, A. (2017). Routing games over time with FIFO policy. In R. Devanur, N. and Lu, P., editors, *Web and Internet Economics*, pages 266–280, Cham. Springer International Publishing.

Koch, R. (2012). *Routing games over time.* Ph.D. thesis, Technische Universität Berline.

Koch, R. and Skutella, M. (2009). Nash equilibria and the price of anarchy for flows over time. In *International Symposium on Algorithmic Game Theory*, pages 323–334. Springer.

Koch, R. and Skutella, M. (2011). Nash equilibria and the price of anarchy for flows over time. *Theory of Computing Systems*, 49(1):71–97.

Kulkarni, J. and Mirrokni, V. (2015). Robust price of anarchy bounds via LP and Fenchel duality. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 1030–1049, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Long, J., Huang, H.-J., Gao, Z., and Szeto, W. Y. (2013). An intersection-movement-based dynamic user optimal route choice problem. *Operations Research*, 61(5):1134–1147.

Long, J. and Szeto, W. Y. (2019). Link-based system optimum dynamic traffic assignment problems in general networks. *Operations Research*, 67(1):167–182.

Macko, M., Larson, K., and Steskal, Ľ. (2013). Braess's paradox for flows over time. *Theory of Computing Systems*, 53(1):86–106.

Marcotte, P., Nguyen, S., and Schoeb, A. (2004). A strategic flow model of traffic assignment in static capacitated networks. *Operations Research*, 52(2):191–212.

Meunier, F. and Wagner, N. (2010). Equilibrium results for dynamic congestion games. *Transportation Science*, 44(4):524–536.

Peeta, S. and Ziliaskopoulos, A. K. (2001). Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1(3):233–265.

Roughgarden, T. (2007). Routing games. *Algorithmic Game Theory*, 18:459–484.

Roughgarden, T. and Tardos, É. (2002). How bad is selfish routing? *Journal of the ACM*, 49(2):236–259.

Scarsini, M., Schröder, M., and Tomala, T. (2018). Dynamic atomic congestion games with seasonal flows. *Operations Research*, 66(2):327–339.

Selten, R. (1965). Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit: Teil i: Bestimmung des dynamischen preisgleichgewichts. *Zeitschrift für die gesamte Staatswissenschaft/Journal of Institutional and Theoretical Economics*, (H. 2):301–324.

Sering, L. and Koch, L. V. (2019). *Nash Flows Over Time with Spillback*, pages 935–945.

Sheffi, Y. (1985). *Urban transportation networks*, volume 6. Prentice-Hall, Englewood Cliffs, NJ.

Vickrey, W. S. (1969). Congestion theory and transport investment. *The American Economic Review*, 59(2):251–260.

Wardrop, J. G. (1952). Road paper: Some theoretical aspects of road traffic research. In *ICE Proceedings: Engineering Divisions*, volume 1, pages 325–362. Thomas Telford.

Werth, T., Holzhauser, M., and Krumke, S. (2014). Atomic routing in a deterministic queuing model. *Operations Research Perspectives*, 1(1):18–41.

Yagar, S. (1971). Dynamic traffic assignment by individual path minimization and queuing. *Transportation Research*, 5(3):179–196.

Electronic Companion

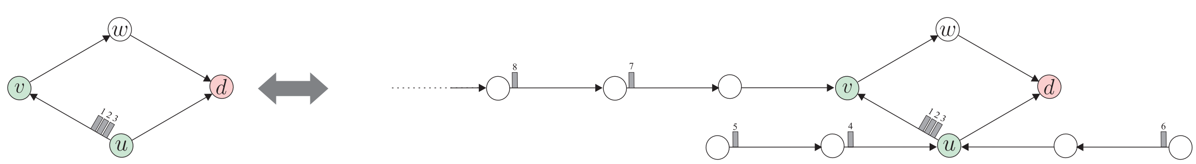
Throughout, for any directed multi-edge graph H with vertex set $\hat{V}$ and edge set $\hat{E}$, and any vertex $v\in\hat{V}$, we use $\hat{E}^+(v)$ and $\hat{E}^-(v)$ to denote the set of outgoing edges from v and the set of incoming edges to v in H, respectively. To avoid possible confusion in expressing parallel edges, for each edge $e\in\hat{E}$ with tail vertex u and head vertex v, we give u and v aliases u_e and v_e, respectively. When we write e as uv, we always mean $u=u_e$ and $v=v_e$.

For any strategy (path) profile $\boldsymbol{p}=(P_i)_{i\in\Delta}$ of game Γ^{N} and any agent subset S of Δ, the partial path profiles $(P_i)_{i\in S}$ and $(P_i)_{i\in\Delta\setminus S}$ are abbreviated to $\boldsymbol{p}_S$ and $\boldsymbol{p}_{-S}$, respectively. In particular, $\boldsymbol{p}_\emptyset$ is viewed as an empty profile.

EC.1. A technical transformation

In our model Γ^{N}, the superficial difference between the agents inside the initial queues and those outside makes our presentation cumbersome. To avoid awkward descriptions and also indicate more insights into agent interactions in model Γ^{N}, we introduce a new model, denoted by $\bar{\Gamma}^{\mathrm{N}}$, which we show is equivalent to Γ^{N}. The three seemly different characteristics of an agent in Γ^{N}, entry time, origin, and original rank, are unified to a single location feature in $\bar{\Gamma}^{\mathrm{N}}$. Studying this equivalent model not only significantly simplifies the five ranking criteria in edge-priority DQ rule (see Section 3) and many technical definitions (such as "agent preemption" in Section 4.4.1), but also substantially shortens our proofs (avoiding tedious case analyses to deal with different agent characteristics).

In $\bar{\Gamma}^{\mathrm{N}}$, all agents are located at the initial queues of the new input network $\bar{G}$, which is either a finite acyclic directed graph or some special infinite graph as illustrated in Figure EC.1: G is a subgraph of $\bar{G}$. Both games Γ^{N} and $\bar{\Gamma}^{\mathrm{N}}$ have the same set of agents $\Delta=\mathbb{Q}^0_{uv}\cup\Delta_{1,u}\cup\Delta_{2,u}\cup_{r\geq 1}\Delta_{r,v}$, and the same initial queue $\mathbb{Q}^0_{uv}=\{1,2,3\}$ in G; The sets of sequentially arriving agents in Γ^{N}, i.e., $\Delta_{1,u}=\{4\}$, $\Delta_{2,u}=\{5,6\}$ with agent 5 having a higher original rank than 6, $\Delta_{1,v}=\emptyset$, and $\Delta_{r,v}=\{r+5\}$ for every $r\geq 2$, correspond to initial queues outside G in $\bar{\Gamma}^{\mathrm{N}}$.



(a) Π on G at time $t=0$ (b) $\bar{\Pi}$ on $\bar{G}$ at time $t=0$

Figure EC.1 Game Γ^{N} on G vs. game $\bar{\Gamma}^{\mathrm{N}}$ on $\bar{G}$

Recall that the input of a game instance Γ^{N} consists of network $G=(V,E)$ with initial queues $(\mathbb{Q}_e^0)_{e\in E}$ and inflows of agents $\Delta_{r,v}$ ($r\geq 1$, $v\in V$) along with original ranks among them. Corresponding to this game instance, an instance of game $\bar{\Gamma}^{\mathrm{N}}$ is specified by the same set $\bar{\Delta}:=\Delta=(\cup_{e\in E}\mathbb{Q}_e^0)\cup(\cup_{r\geq 1,v\in V}\Delta_{r,v})$ of nonadaptive agents, but with a modified network input $\bar{G}=(\bar{V},\bar{E})$, which is obtained from G by adding a (typically infinite) number of pendant paths and specifying initial locations of the agents $\cup_{r\geq 1,v\in V}\Delta_{r,v}$ at the newly added pendant paths, as detailed below:

(T1) Network $\bar{G}$: For each vertex $v\in V$, we add a number $\max_{r\geq 1}|\Delta_{r,v}|$ (which is possibly zero) of paths $P_1^v,P_2^v,\ldots$ directed to v, each intersecting G only at v. Outside G, all added paths are mutually vertex-disjoint.

(T2) Priority preservation: For each $v\in V$, the decreasing priority ordering of the added incoming edges to v agrees with the (subscript) ordering of added paths containing them: the unique edge in P_1^v incoming to v has the highest priority, the one in P_2^v incoming to v has the second highest priority, and so on. This ordering is followed by the given ordering of incoming edges to v in G, which makes a complete priority order over all incoming edges to v in $\bar{G}$.

(T3) Rank preservation: Agent $i\in\Delta_{r,v}$ in game Γ^{N} with the hth highest original rank corresponds to agent $i\in\bar{\Delta}$ in game $\bar{\Gamma}^{\mathrm{N}}$, who is the only agent queuing at time 0 on path P_h^v at a distance r from v, where distance is measured by the number of edges. Particularly, agent i in $\bar{\Gamma}^{\mathrm{N}}$ will reach v at time r through the (added) incoming edge to v with the hth highest priority.

With the above transformation, it is easy to see that the agent set $\bar{\Delta}$ of game $\bar{\Gamma}^{\mathrm{N}}$ is simply the disjoint union of its initial queues, which we still denote as $\mathbb{Q}_e^0$, over all $e\in\bar{E}$. The game $\bar{\Gamma}^{\mathrm{N}}$ starts at time 0 with the input $\bar{G}$ and $(\mathbb{Q}_e^0)_{e\in\bar{E}}$. No entries of agents into the network $\bar{G}$ are involved throughout the game and all agents are in $\bar{G}$ from the very beginning. Therefore, no original ranks are needed to break ties. (Note that we have transformed the original ranks among $\Delta_{r,v}$ to the priorities of the added edges incoming to v.)

Due to the simpler form of the input for $\bar{\Gamma}^{\mathrm{N}}$, the edge-priority DQ rule (see Section 3) when applied to $\bar{\Gamma}^{\mathrm{N}}$ is simplified: we do not need rules (R3) and (R4) any more. Regarding any fixed edge e (with tail vertex u_e) and any pair of agents, the agent who enters e earlier has a higher queue rank at e. Ties are broken via (R2) only: higher priority is given to the agent who enters e through an incoming edge to u_e that has a higher priority according to $\prec_{u_e}$. Another simplification resulted from our transformation is that in $\bar{\Gamma}^{\mathrm{N}}$ all information about agent set is contained in the initial queues $(\mathbb{Q}_e^0)_{e\in\bar{E}}$. The chronological order of entrances into G in Γ^{N} is visualized by the lengths of paths of $\bar{G}$ in $\bar{\Gamma}^{\mathrm{N}}$, which makes the task of investigating agent interactions easier. For instance, in the example illustrated in Figure EC.1, game $\bar{\Gamma}^{\mathrm{N}}$ provides a faster way for one to find that agent 7 will preempt agent 3 at vertex v.

Each agent $i \in \bar{\Delta}$ selects a path *starting from his initial edge* (the edge where he queues at time 0) and ending at the common destination vertex d. Such paths form his strategy set, which we write as $\bar{\mathscr{P}}_i$ to distinguish it from $\mathscr{P}_i$ in game Γ^{N}. We use o_i to denote the tail vertex of initial edge of agent $i \in \bar{\Delta}$ in $\bar{G}$. The following fact is obvious by our transformation from game Γ^{N} on G to game $\bar{\Gamma}^{\mathrm{N}}$ on $\bar{G}$.

Lemma EC.1. *Given game Γ^{N} with input (G, Δ), let game $\bar{\Gamma}^{\mathrm{N}}$ with input $(\bar{G}, \bar{\Delta})$ be constructed as in* (T1)–(T3)*. Then the game Γ^{N} is exactly the restriction of the game $\bar{\Gamma}^{\mathrm{N}}$ to G: the strategies and movements of agents together with their arrival times at vertices along their paths in Γ^{N} are identical with those in the restriction of $\bar{\Gamma}^{\mathrm{N}}$ to G.*

In view of agents' trivial movements outside G in game $\bar{\Gamma}^{\mathrm{N}}$, the above lemma enables us to turn our attention to $\bar{\Gamma}^{\mathrm{N}}$ when studying Γ^{N}. These two games are essentially identical. The notation and definitions introduced for game model Γ^{N} apply to game model $\bar{\Gamma}^{\mathrm{N}}$, as the latter is simply a special case of the former.

EC.2. Algorithm for finding an IDNE

In this section, we construct an IDNE for every game $\bar{\Gamma}^{\mathrm{N}}$ (see Definition 2). The result along with Lemma EC.1 directly yields an IDNE for every game Γ^{N}. Recall that, $[0] = \emptyset$, and for any positive integer k, $[k]$ denotes the set of all positive integers no more than k.

Algorithm. We are able to reindex the agents of $\bar{\Delta}$ as $1, 2, \ldots$ and find the associated path profile $\bar{\boldsymbol{p}} = (\bar{P}_1, \bar{P}_2, \ldots)$ such that, each agent $k \in \bar{\Delta}$ is a *dominator* in $\bar{\Delta} \backslash [k-1]$ and $\bar{P}_k$ is a *dominant path* in the following sense: under the assumption that agents in $[k-1]$ all follow $\bar{\boldsymbol{p}}_{[k-1]}$, as long as agent k takes $\bar{P}_k$, he will be among the first in $\bar{\Delta} \backslash [k-1]$ to reach every vertex of the path. Specifically, for any vertex v on $\bar{P}_k$, and partial path profile $\boldsymbol{q}_{-[k]}$ for agents in $\Delta \backslash [k]$, we have

$$t_k^v \left(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{-[k]}\right) = \min \left\{ t_j^v(\bar{\boldsymbol{p}}_{[k-1]}, \boldsymbol{r}_{-[k-1]}) : j \in \Delta \backslash [k-1], \right.$$
$$\left. \text{and } \boldsymbol{r}_{-[k-1]} \text{ is partial profile for } \Delta \backslash [k-1] \right\}. \tag{EC.1}$$

We call such a path profile *iteratively dominant*. As explained in Section 4.1, it is actually an NE, i.e., IDNE, of game $\bar{\Gamma}^{\mathrm{N}}$.

For completeness, we repeat the sketch of the algorithm in the context of $\bar{\Gamma}^{\mathrm{N}}$. Our algorithm runs roughly as follows. Initially, let agent subset $[0]$ of $\bar{\Delta}$ and partial routing $\bar{\boldsymbol{p}}_{[0]}$ of agents in $[0]$ be empty. Then recursively, assuming agents in $[k-1]$ go along their paths as specified in $\bar{\boldsymbol{p}}_{[k-1]}$, we enlarge $[k-1]$ with a new agent $k \in \bar{\Delta} \backslash [k-1]$ and enlarge $\bar{\boldsymbol{p}}_{[k-1]}$ with a path $\bar{P}_k \in \bar{\mathscr{P}}_k$ in the following way. For each agent $j \in \bar{\Delta} \backslash [k-1]$ and vertex $v \in \bar{V}$, we define

$$\tau_j^v = \min\{t_j^v(\bar{\boldsymbol{p}}_{[k-1]}, R_j) \,|\, R_j \in \bar{\mathscr{P}}_j\}$$

as the "ideal arrival time" of agent j at vertex v, where $t_j^v(\bar{\boldsymbol{p}}_{[k-1]}, R_j)$ is j's arrival time at v assuming that all agents in $\bar{G}$ are only those in $[k-1]\cup\{j\}$ and they follow $(\bar{\boldsymbol{p}}_{[k-1]}, R_j)$. We select the candidates j and $\bar{P}_j$ for agent k and his path $\bar{P}_k$ step by step. Initially, let u denote the destination vertex d. Let $j\in\bar{\Delta}\backslash[k-1]$ and $\bar{P}_j\in\bar{\mathscr{P}}_j$ be such that

(S1) $t_j^u(\bar{\boldsymbol{p}}_{[k-1]}, \bar{P}_j)$ equals $\min_{i\in\bar{\Delta}\backslash[k-1]}\tau_i^u$, the earliest ideal arrival time at u among all agents in $\bar{\Delta}\backslash[k-1]$;

(S2) If more than one candidate $(j,\bar{P}_j)$ satisfy (S1), then the choice of $(j,\bar{P}_j)$ from the candidates is made such that the incoming edge to u on $\bar{P}_j$ has the highest possible priority;

(S3) If still more than one candidate $(j,\bar{P}_j)$ satisfy (S1) and (S2), then the paths $\bar{P}_j$ involved must share the same incoming edge e to u (whose tail vertex is denoted as u_e), and at least one such $\bar{P}_j$ satisfies that $t_j^{u_e}(\bar{\boldsymbol{p}}_{[k-1]}, \bar{P}_j)$ is the earliest ideal arrival time at vertex u_e among all agents in $\bar{\Delta}\backslash[k-1]$; we update u with u_e and go back to (S1) for further selections from the *current* candidates (i.e., those satisfying all (S1), (S2), (S3) checked), unless e is the initial edge of all candidate agents j.

The above process is repeated until either only one candidate pair $(j,\bar{P}_j)$ is left or the edge $e\in\bar{E}$ in (S3) becomes the initial edge of all the remaining candidate agents. In the former case, we set $(k,\bar{P}_k)=(j,\bar{P}_j)$. In the latter case, all candidate paths must be identical, and we choose agent k to be the head of queue $\mathbb{Q}_e^0$ and set $\bar{P}_k$ to be the identical candidate path. In either case, we enlarge $[k-1]$ by k, augment $\bar{\boldsymbol{p}}_{[k-1]}$ with $\bar{P}_k$, obtaining a larger agent subset $[k]$ and the associated path profile $\bar{\boldsymbol{p}}_{[k]}$. We then iterate the above procedure based on $[k]$ and $\bar{\boldsymbol{p}}_{[k]}$. A formal description of the process is presented in Algorithm 2 on page ec5.

Proofs. Let the game instance $\bar{\Gamma}^{\mathrm{N}}$ on $(\bar{G},\bar{\Delta})$ be as specified in the input of Algorithm 2. To facilitate our discussions, we introduce some new notations. Given any path P in $\bar{G}$, a u-v subpath of P is often written as $P[u,v]$; furthermore, we write $P(u,v]=P[u,v]\backslash\{u\}$, $P[u,v)=P[u,v]\backslash\{v\}$ and $P(u,v)=P[u,v]\backslash\{u,v\}$.

Recall that o_i denotes the tail vertex of the initial edge of agent $i\in\bar{\Delta}$. Let agents $1,2,\dots$ of $\bar{\Delta}$ be indexed and path profile $\bar{\boldsymbol{p}}=(\bar{P}_k)_{k\in\bar{\Delta}}$ be computed as in Algorithm 2. Recall that $[0]=\emptyset$ and $\bar{\boldsymbol{p}}_{[0]}$ is the null profile. For any nonnegative integer k, any agent index j with $j>k$ and any vertex $v\in\bar{V}$, let

$$\ell_j^v[\![k]\!] := \min\left\{t_j^v(\bar{\boldsymbol{p}}_{[k]}, R_j)\,|\, R_j\in\bar{\mathscr{P}}_j\right\}$$

denote the value ℓ_j^v computed for agent j in Step 3 at the $(k+1)$st iteration of Algorithm 2, i.e., the earliest time for agent j to reach vertex v, based only on the partial routing $\bar{\boldsymbol{p}}_{[k]}$ of agents in $[k]$. In particular, by definition, $t_j^v(\bar{\boldsymbol{p}}_{[j]})=\ell_j^v[\![j-1]\!]$ for every agent j and every vertex $v\in\bar{P}_j$.

Algorithm 2 (Iteratively Dominant NE)

Input: game instance $\bar{\Gamma}^{\mathrm{N}}$: network $\bar{G}=(\bar{V},\bar{E})$ with initial queues $\mathbb{Q}_e^0, e\in\bar{E}$, where agent set $\bar{\Delta}=\cup_{e\in\bar{E}}\mathbb{Q}_e^0$.

Output: the special IDNE $\bar{\boldsymbol{p}}=(\bar{P}_k)_{k\in\bar{\Delta}}$ along with the corresponding agent indices 1, 2, ...

1. Initiate $\bar{\boldsymbol{p}}_{[0]}\leftarrow\emptyset$, $k\leftarrow 0$.
2. $k\leftarrow k+1$.

   (NB: Start to search for a new dominator k and his associated dominant path $\bar{P}_k$.)
3. **For** each agent $j\in\bar{\Delta}\backslash[k-1]$ and vertex $v\in\bar{V}$ **Do**
   - $\ell_j^v\leftarrow\min\left\{t_j^v(\bar{\boldsymbol{p}}_{[k-1]},R_j)\,|\,R_j\in\bar{\mathscr{P}}_j\right\}$;

     (NB: ℓ_j^v is the earliest time for j to reach vertex v, assuming that all other agents in $\bar{G}$ are those in $[k-1]$ and they go along their paths specified in $\bar{\boldsymbol{p}}_{[k-1]}$. Note that ℓ_j^v $(<\infty)$ is computable by the Dijkstra-like algorithm in Theorem EC.4 with partial path profile $\bar{\boldsymbol{p}}_{[k-1]}$ and agent j in place of $\boldsymbol{q}_{-i}$ and i over there, whose output τ^v is exactly ℓ_j^v.)
   - $\bar{\mathscr{P}}_j^v\leftarrow\left\{R_j[o_j,v]\,|\,R_j\in\bar{\mathscr{P}}_j \text{ and } t_j^v(\bar{\boldsymbol{p}}_{[k-1]},R_j)=\ell_j^v\right\}$.

     (NB: $\bar{\mathscr{P}}_j^v$ denotes the set of all paths starting with j's initial edge and ending at v along which j can reach v at time ℓ_j^v, under the above assumption. If there is no such a path in $\bar{G}$, then $\ell_j^v=\infty$ and $\bar{\mathscr{P}}_j^v=\emptyset$.)

   End-For
4. $C\leftarrow\bar{\Delta}\backslash[k-1]$, $P\leftarrow\emptyset$, $w\leftarrow d$.

   (NB: In the following while-loop, C is a set of candidates j for selecting k that will be pruned step by step; P is a subpath of $\bar{P}_j$ that will grow edge by edge starting from d; w is the latest vertex added to P; the value ℓ is strictly decreasing, which guarantees the termination of the while-loop.)
5. **While** $\ell:=\min_{j\in C}\ell_j^w\geq 1$ **Do**

   $C\leftarrow\{j\in C\,|\,\ell_j^w=\ell\}$;

   $uw\leftarrow$ the edge of the highest priority among all ending edges of paths in $\bigcup_{j\in C}\bar{\mathscr{P}}_j^w$;

   $P\leftarrow P\cup\{uw\}$;

   $w\leftarrow u$;

   End-While

   (NB: at the end of the while-loop the starting edge of P is the common initial edge of all agents in C.)
6. Let $k\in C$ be the agent who, at the very beginning, stands first (among all agents in C) on the starting edge of P.

   (NB: The agent k selected is called the *dominator of* $\bar{\Delta}\backslash[k-1]$.)
7. Let k be *associated* `with` $\bar{P}_k\leftarrow P$, $\bar{\boldsymbol{p}}_{[k]}\leftarrow(\bar{\boldsymbol{p}}_{[k-1]},\bar{P}_k)$.

   (NB: The algorithm outputs agent k and his associated dominant path $\bar{P}_k$.)
8. **If** $k<|\bar{\Delta}|$, **Then** go to Step 2.

Lemma EC.2. *Let agent indices j,k and agent subset S satisfy $j \in S \subseteq \bar{\Delta} \backslash [k-1]$. Then for every vertex $v \in \bar{P}_k$ and every path profile $\boldsymbol{q} = (Q_h)_{h \in \bar{\Delta}}$ of game $\bar{\Gamma}^{\mathrm{N}}$, it holds that*

$$t_k^v(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S \backslash \{k\}}) = \ell_k^v [\![k-1]\!] \le t_j^v(\bar{\boldsymbol{p}}_{[k-1]}, \boldsymbol{q}_S) \tag{EC.2}$$

This lemma exhibits some invariance and dominance properties possessed by the agent order 1, 2, ... and the path profile $\bar{\boldsymbol{p}}$ computed by Algorithm 2. Specifically, for every agent index k, as long as all agents in $[k]$ follow $\bar{\boldsymbol{p}}_{[k]}$, the following properties hold, no matter what paths other agents outside $[k]$ choose (even if some or all of them are missing):

- *Invariant arrival times of agents in* $[k]$: the arrival times at any vertex of their paths in $\bar{\boldsymbol{p}}_{[k]}$ can never be affected. (This is what the equality in (EC.2) says.)
- *Universal domination of agents in* $[k]$*:* no agent $j \ge k+1$ can overtake any agent $i \in [k]$ at any vertex of his route $\bar{P}_i$, which is what the inequality in (EC.2) says. In particular, it follows that if j queues before some agent, then this agent is outside $[k]$.
- *Invariant influence of agents in* $[k]$: due to their arrival-time invariance, the agents in $[k]$ who queue at an edge at some time depends only on the time and the edge under consideration (but not on the choices of agents outside $[k]$), which implies that the agents in $[k]$ exert *invariant influence* on the movements of other agents outside $[k]$.
- *Property of no speed-up for agents in* $\bar{\Delta} \backslash [k]$: due to the invariant influence of agents in $[k]$, no agent $j \in \bar{\Delta} \backslash [k]$ can be sped up by other agent(s) in $\bar{\Delta} \backslash [k]$. More specifically, assuming j follows a path $R_j \in \bar{\mathscr{P}}_j$ and agents in $[k]$ follow $\bar{\boldsymbol{p}}_{[k]}$, the earliest arrival time of j at each vertex of R_j is attained when no other agents are involved — involving some or all agents from $\bar{\Delta} \backslash ([k] \cup \{j\})$ in the routing cannot make j's arrival time earlier at any vertex of R_j. (Note that this seemingly quite natural property does not hold in general. See Example 4 for more discussions.)

Proof of Lemma EC.2. Let e denote the incoming edge to d that has the highest priority w.r.t. $\prec_d$. For convenience, we may assume w.l.o.g. that e is the initial edge of some agent. Otherwise, we could add a dummy agent to $\mathbb{Q}_e^0$, which does not exert any influence on the original agents, nor the output of Algorithm 2 with the dummy agent and his unique path e ignored. (Note that the dummy agent would be the first output by the algorithm.)

We prove (EC.2) by induction on k. For the base case $k=1$, under the above assumption, agent 1 is the head of $\mathbb{Q}_e^0$ and $\bar{P}_1 = e$. In any case, agent 1 reaches d at time 1, and (EC.2) is trivial. Suppose now $k \ge 2$ and (EC.2) is valid when k is smaller. This means that the invariant arrival times, universal domination and hence the *invariant influence* (resp. *no speed-up property*), as stated above, are true for agents in $[k-1]$ (resp. $\bar{\Delta} \backslash [k-1]$).

We claim $\ell_k^v[\![k-1]\!] \le \ell_j^v[\![k-1]\!]$. Otherwise, there would be a path $R_j \in \bar{\mathscr{P}}_j$ with $t_j^v(\bar{\boldsymbol{p}}_{[k-1]}, R_j) = \ell_j^v[\![k-1]\!]$ such that based on $\bar{\boldsymbol{p}}_{[k-1]}$, agent j would be able to use path $R_j[o_j, v] \cup \bar{P}_k[v, d] \in \bar{\mathscr{P}}_j$ to reach v earlier than k and subsequently reach all vertices of $\bar{P}_k(v, d]$, including d, no later than k, contradicting the choices of k and $\bar{P}_k$. The inequality part of (EC.2) thus follows from

$$\ell_k^v[\![k-1]\!] \le \ell_j^v[\![k-1]\!] \le t_j^v(\bar{\boldsymbol{p}}_{[k-1]}, Q_j) \le t_j^v(\bar{\boldsymbol{p}}_{[k-1]}, \boldsymbol{q}_S),$$

where the second inequality is by definition and the third is due to the no speed-up property of $\bar{\Delta}\backslash[k-1]$: agent j cannot be sped up by agents in $S\backslash\{j\}$.

Note that $t_k^v(\bar{\boldsymbol{p}}_{[k-1]}, \bar{P}_k) \le t_k^v(\bar{\boldsymbol{p}}_{[k-1]}, \bar{P}_k, \boldsymbol{q}_{S\backslash\{k\}})$, because k cannot be sped up by agents in $S\backslash\{k\}$. Thus, $t_k^v(\bar{\boldsymbol{p}}_{[k]}) = \ell_k^v[\![k-1]\!]$ implies

$$t_k^v(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}}) \ge \ell_k^v[\![k-1]\!].$$

So it remains to show that $t_k^v(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}}) \le \ell_k^v[\![k-1]\!]$, i.e., agents in $S\setminus\{k\}$ do not slow down agent k. Suppose on the contrary that $t_k^v(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}}) > \ell_k^v[\![k-1]\!] = t_k^v(\bar{\boldsymbol{p}}_{[k]})$ for some vertex $v \in \bar{P}_k$. Let v be the first such vertex encountered when traveling along $\bar{P}_k$, indicating that

(1) $t_k^w(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}}) = \ell_k^w[\![k-1]\!]$ for every vertex $w \in \bar{P}_k[o_k, v) = \bar{P}_k[o_k, v]\backslash\{v\}$.

In view of the invariant influence from agents in $[k-1]$ who follow $\bar{\boldsymbol{p}}_{[k-1]}$, there must exist some agent $i \in S\backslash\{k\}$ and an edge $xy \in \bar{P}_k[o_k, v]$ such that i slows down k on xy, or more precisely, under $(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}})$, agent i enters xy earlier than k, or enters xy at the same time as k and queues before k at xy. Let xy be the *first* such edge encountered when traveling along $\bar{P}_k[o_k, v]$. Observe that $x \in \bar{P}_k[o_k, v)$. By (1), we have

(2) under routing $(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}})$, agent i reaches vertex x and enters edge xy at time $t_i^x(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}}) \le t_k^x(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}}) = \ell_k^x[\![k-1]\!]$.

Construct a path $R_i := Q_i[o_i, x] \cup \bar{P}_k[x, d] \in \bar{\mathscr{P}}_i$ for agent i. Note that

(3) $t_i^x(\bar{\boldsymbol{p}}_{[k-1]}, R_i) = t_i^x(\bar{\boldsymbol{p}}_{[k-1]}, Q_i) \le t_i^x(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}})$,

where the equality follows from the definition of R_i and the inequality is due to the no speed-up property of $\bar{\Delta}\backslash[k-1]$: agent $i \notin [k]$ cannot be sped up by agents in $(S \cup \{k\})\backslash\{i\}$. In turn, we deduce from (3) and (2) that $t_i^x(\bar{\boldsymbol{p}}_{[k-1]}, R_i) \le \ell_k^x[\![k-1]\!] = t_k^x(\bar{\boldsymbol{p}}_{[k-1]}, \bar{P}_k)$. Consequently,

(4) $t_i^w(\bar{\boldsymbol{p}}_{[k-1]}, R_i) \le t_k^w(\bar{\boldsymbol{p}}_{[k]}) = \ell_k^w[\![k-1]\!]$ for each vertex $w \in R_i[x, d] = \bar{P}_k[x, d]$.

By the definition of agent k from Algorithm 2, we derive from (4) that

$$t_i^w(\bar{\boldsymbol{p}}_{[k-1]}, R_i) = \ell_k^w[\![k-1]\!] \text{ for each vertex } w \in R_i[x, d] = \bar{P}_k[x, d].$$

Consider $w = x$ in the above equation, we derive from (3) and (2) that

$$\ell_k^x[\![k-1]\!] = t_i^x(\bar{\boldsymbol{p}}_{[k-1]}, R_i) \le t_i^x(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}}) \le t_k^x(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\backslash\{k\}}) = \ell_k^x[\![k-1]\!].$$

The string of inequalities enforces that under $(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{S\setminus\{k\}})$, agents i and k enter xy at the same time, and therefore i queues before k at xy (recall that i slows down k on xy). So it must be the case that

- either (if R_i and $\bar{P}_k$ have different incoming edges to x) R_i has a higher priority incoming edge into x than $\bar{P}_k$ does,
- or (by the choice of edge xy, i.e., o_k's proximity to xy) $\{i,k\} \subseteq \mathbb{Q}^0_{xy}$, and agent i queues before agent k at their common initial edge xy.

However, the choice made at the kth iteration of Algorithm 2 excludes the possibilities of both cases. This completes the proof. Q.E.D.

By Lemma EC.1, profile $\bar{\boldsymbol{p}}$ is an IDNE of $\bar{\Gamma}^{\mathrm{N}}$ if and only if the restriction of $\bar{\boldsymbol{p}}$ to G is an IDNE of Γ^{N}. The following establishes Theorem 1.

THEOREM EC.2. *Algorithm 2 finds an IDNE of game $\bar{\Gamma}^{\mathrm{N}}$.*

Proof. For any agent $j \in \Delta \setminus [k-1]$, partial path profiles $\boldsymbol{q}_{-[k]}$ and $\boldsymbol{r}_{-[k-1]}$ for $\Delta\setminus[k]$ and $\Delta\setminus[k-1]$, it is instant from (EC.2) that $t^v_k(\bar{\boldsymbol{p}}_{[k]}, \boldsymbol{q}_{-[k]}) = \ell^v_k[\![k]\!] \le t^v_j(\bar{\boldsymbol{p}}_{[k-1]}, \boldsymbol{r}_{-[k-1]})$, which shows the validity of (EC.1) and thus that $\bar{\boldsymbol{p}}$ is an IDNE of $\bar{\Gamma}^{\mathrm{N}}$. Q.E.D.

EC.3. Generalized iterative dominance

As can be seen from the proof of Lemma EC.2, our induction hypothesis only involves the equation part of (EC.2), which guarantees the critical invariant influence property. This leads us to the following generalization of Algorithm 2 (see page ec9), which computes an iteratively dominant *partial* path profile *based* on a fixed routing of some special agents.

The verbatim adaption of the proof of Lemma EC.2 gives the following generalization for iterative dominance. It plays critical roles in proving the equilibrium properties presented in Sections 4.3 and 5.3.

LEMMA EC.3. *Regarding Algorithm 3, if $j \in S \subseteq \bar{\Delta}\setminus(U \cup [i-1])$, then for every vertex $v \in \bar{P}_i$ and path profile $\boldsymbol{q}$ of game $\bar{\Gamma}^{\mathrm{N}}$, it holds that*

$$t^v_i(\boldsymbol{b}, \bar{\boldsymbol{p}}_{[i]}, \boldsymbol{q}_{S\setminus[i]}) = \min_{R_i \in \bar{\mathscr{P}}_i} t^v_i(\boldsymbol{b}, \bar{\boldsymbol{p}}_{[i-1]}, R_i) \le t^v_j(\boldsymbol{b}, \bar{\boldsymbol{p}}_{[i-1]}, \boldsymbol{q}_S).$$

EC.4. Agent preemptions

This section elaborates on the notion of preemption (introduced in Section 4.4.1) for game $\bar{\Gamma}^{\mathrm{N}}$. Recall that, under some routing, if an agent does not reach a vertex, then we regard his arrival time at the vertex as infinity.

Algorithm 3 (Iteratively Dominant Partial Path Profile with a Base)

Input: game instance $\bar{\Gamma}^{\mathrm{N}}$: network $\bar{G}$ with agent set $\bar{\Delta}$, a partial path profile $\boldsymbol{b} = (B_h)_{h \in U}$ for a (possibly empty) finite subset $U \subseteq \bar{\Delta}$ that satisfies the following *arrival-time invariance*: for every agent $h \in U$ and every vertex $v \in B_h$, the arrival time $t_h^v(\boldsymbol{b}, \boldsymbol{q}_S)$ of h at v is an invariant against changing partial path profile $\boldsymbol{q}_S$, i.e., it is the same over all path profiles $\boldsymbol{q}$ of $\bar{\Gamma}^{\mathrm{N}}$ and agent subsets $S \subseteq \bar{\Delta} \backslash U$.

Output: the special iteratively dominant partial path profile (routing) $\bar{\boldsymbol{p}} = (\bar{P}_i)_{i \in \bar{\Delta} \backslash U}$ for $\bar{\Delta} \backslash U$ along with the corresponding agent indices 1, 2, ...

1. Initiate $\bar{\boldsymbol{p}}_{[0]} \leftarrow \emptyset$, $i \leftarrow 0$.
2. $i \leftarrow i + 1$

   (NB: Start to search for a new dominator i and his associated dominant path $\bar{P}_i$.)
3. **For** each agent $j \in \bar{\Delta} \backslash (U \cup [i-1])$ and vertex $v \in \bar{V}$ **Do**
   - $\ell_j^v \leftarrow \min\{t_j^v(\boldsymbol{b}, \bar{\boldsymbol{p}}_{[i-1]}, R_j) \,|\, R_j \in \bar{\mathscr{P}}_j\}$
   - $\bar{\mathscr{P}}_j^v \leftarrow \{R_j[o_j, v] \,|\, R_j \in \bar{\mathscr{P}}_j \text{ and } t_j^v(\boldsymbol{b}, \bar{\mathbf{p}}_{[i-1]}, R_j) = \ell_j^v\}$

   End-For
4. $C \leftarrow \bar{\Delta} \backslash (U \cup [i-1])$, $P \leftarrow \emptyset$, $w \leftarrow d$.
5. Run Steps 5 to 7 of Algorithm 2 to identify dominator i of $\bar{\Delta} \backslash (U \cup [i-1])$ and his associated dominant path $\bar{P}_i$.

   (NB: The algorithm returns agent i and his associated dominant path $\bar{P}_i$.)
6. Set $\bar{\boldsymbol{p}}_{[i]} \leftarrow (\bar{\boldsymbol{p}}_{[i-1]}, \bar{P}_i)$.
7. **If** $i < |\bar{\Delta}| - |U|$, **Then** go to Step 2.

Throughout this section, given game $\bar{\Gamma}^{\mathrm{N}}$ on network $\bar{G} = (\bar{V}, \bar{E})$ with agent set $\bar{\Delta}$, let i denote a fixed agent in $\bar{\Delta}$, and $\boldsymbol{q}_{-i} = (Q_j)_{j \in \bar{\Delta} \backslash \{i\}}$ denote a fixed partial path profile of all other agents. We consider the scenario where only agent i is allowed to change his path. For each vertex $v \in \bar{V}$, define

$$\tau^v := \min_{P_i \in \bar{\mathscr{P}}_i} \{t_i^v(P_i, \boldsymbol{q}_{-i})\}$$

as the earliest time at which agent i can reach vertex v by unilaterally changing his path (if $\bar{\mathscr{P}}_i$ contains no path through v, then we set $\tau^v := +\infty$). Analogously, for each agent $j \in \bar{\Delta} \backslash \{i\}$ and vertex $v \in Q_j$, define

$$\tau_j^v := \min_{P_i \in \bar{\mathscr{P}}_i} \{t_j^v(P_i, \boldsymbol{q}_{-i})\}$$

as the earliest time at which agent j can reach vertex v when agent i unilaterally changes his path. We emphasize that j keeps following his path Q_j (specified by $\boldsymbol{q}_{-i}$) in the definition of τ_j^v.

In the following, for any non-singleton path P in $\bar{G}$ and any non-starting vertex v of P, we use $e_v(P)$ to denote the incoming edge to v on P. By virtue of the technical transformation in Section EC.1, the preempt relation defined for game Γ^{N} in Section 4.4.1 translates to the following simplified definition for preemptions in $\bar{\Gamma}^{\mathrm{N}}$.

Definition EC.1 (Preemption). For every agent $j \in \bar{\Delta} \backslash \{i\}$ and vertex $v \in Q_j \backslash \{o_j\}$, we say that agent i *preempts* agent j at vertex v under $\boldsymbol{q}_{-i}$ if either $\tau^v < \tau^v_j$, or $\tau^v = \tau^v_j$ and $v \neq o_i$ is on some path $P_i \in \bar{\mathscr{P}}_i$ such that $t^v_i(P_i, \boldsymbol{q}_{-i}) = \tau^v$ and $e_v(P_i) \preceq_v e_v(Q_j)$.

Define vertex subset

$$\bar{Y} := \{v \in \bar{V} \mid \tau^v < \infty\}.$$

For each $v \in \bar{Y}$, let $O^v_i \in \bar{\mathscr{P}}_i$ denote the path achieving $\tau^v = t^v_i(O^v_i, \boldsymbol{q}_{-i})$ such that the priority of $e_v(O^v_i)$ w.r.t. $\prec_v$ is as high as possible. It is clear that

$$\begin{array}{l}\text{If } i \text{ preempts } j \text{ at } v\text{, then either } t^v_i(O^v_i, \boldsymbol{q}_{-i}) < \tau^v_j,\\ \text{or } t^v_i(O^v_i, \boldsymbol{q}_{-i}) = \tau^v_j \text{ and } e_v(O^v_i) \preceq_v e_v(Q_j).\end{array} \tag{EC.3}$$

For each vertex $v \in \bar{V}$, we denote $\mathcal{A}_v \subset \bar{\Delta}$ as the set of agents j other than i whose arrival times at v can be *affected by* i (with his unilateral path change), i.e., there exist $P_i, P'_i \in \bar{\mathscr{P}}_i$ such that $t^v_j(P_i, \boldsymbol{q}_{-i}) < t^v_j(P'_i, \boldsymbol{q}_{-i})$.

Lemma EC.4. *If $\mathcal{A}_v \neq \emptyset$, then v is on some path in $\bar{\mathscr{P}}_i$, i.e., $\tau^v < \infty$.*

Proof. Suppose $j \in \mathcal{A}_v$ and agent j's arrival time at v can be influenced. Let $e_v(Q_j) = uv$ be the incoming edge to v on Q_j. If $\mathcal{A}_u = \emptyset$ and uv is not contained in any path in $\bar{\mathscr{P}}_i$, then no matter which path agent i switches to, the arrival times at u of all agents in $\bar{\Delta} \backslash \{i\}$ and hence j's queuing time at edge uv remain the same as those under $\boldsymbol{q}$, which shows a contradiction to $j \in \mathcal{A}_v$. Therefore, either uv and hence v are contained in some path in $\bar{\mathscr{P}}_i$, in which case we are done, or $\mathcal{A}_u \neq \emptyset$, to which we can apply backward induction (as $\bar{G}$ is acyclic) to derive a path $P \in \bar{\mathscr{P}}_i$ that contains u, giving $v \in P[o_i, u] \cup \{uv\} \cup Q_j[v, d] \in \bar{\mathscr{P}}_i$, as desired. Q.E.D.

Lemma EC.5. *For any agent $j \in \bar{\Delta} \backslash \{i\}$ and vertex $v \in Q_j$, if there exist paths $P_i, P'_i \in \bar{\mathscr{P}}_i$ such that $t^v_j(P_i, \boldsymbol{q}_{-i}) \neq t^v_j(P'_i, \boldsymbol{q}_{-i})$, then $v \in Q_j \backslash \{o_j\}$ and i preempts j at v under $\boldsymbol{q}_{-i}$.*

Proof. Recall from the Unit Assumption that all edges of network $\bar{G}$ have a unit capacity and a unit length. Apparently, if $j \in \mathcal{A}_v$, then it must be the case that $v \in Q_j(o_j, d]$. The lemma can be restated as: agent i preempts all agents of $\mathcal{A}_v$ at vertex v. Notice from Lemma EC.4 that $\{v \mid \mathcal{A}_v \neq \emptyset\} \subseteq \bar{Y}$. To prove the lemma, it suffices to prove that

$$\text{For any vertex } v \in \bar{Y}\text{, agent } i \text{ preempts every agent } j \in \mathcal{A}_v \text{ at vertex } v. \tag{EC.4}$$

Since $\bar{G}$ is acyclic, there exists a complete order on the vertices in $\bar{Y}$ which is *acyclic* in that for each edge with both end-vertices in $\bar{Y}$, its tail vertex has an order smaller than its head vertex. We will verify (EC.4) by induction on the order of the vertices in $\bar{Y}$.

Suppose that $o_i a$ is the initial edge of agent i, which is contained in every path in $\bar{\mathscr{P}}_i$. Therefore, $a \in \bar{Y}$. Apparently, the order of vertex a is the smallest, and the base case where $v = a$ is trivial because of $\mathcal{A}_a = \emptyset$. To proceed inductively, assume that (EC.4) is true for all vertices in $\bar{Y}$ with orders smaller than v.

Since the case $\mathcal{A}_v = \emptyset$ is trivial, we suppose now $\mathcal{A}_v \neq \emptyset$ and consider an arbitrary agent $j \in \mathcal{A}_v$ with $e_v(Q_j) = uv$. In the following, we prove first that agent i preempts agent j at vertex u, then show the preemption at vertex v.

If $j \in \mathcal{A}_u$, since u has a smaller order than v, then by induction hypothesis, agent i preempts agent j at vertex u. If $j \notin \mathcal{A}_u$, then no matter how i changes his path, agent j's arrival time at u cannot be influenced by i. On the other hand, since $j \in \mathcal{A}_v$, there exist $P_i, P'_i \in \bar{\mathscr{P}}_i$ such that $t^v_j(P_i, \boldsymbol{q}_{-i}) < t^v_j(P'_i, \boldsymbol{q}_{-i})$. Then, combining $j \notin \mathcal{A}_u$ and $j \in \mathcal{A}_v$, we deduce that one of the two following cases must happen:

(a) There exists agent $h \in \mathcal{A}_u$ with $uv \in Q_h \cap Q_j$ such that $t^u_h(P'_i, \boldsymbol{q}_{-i}) < t^u_j(P'_i, \boldsymbol{q}_{-i})$, or $t^u_h(P'_i, \boldsymbol{q}_{-i}) = t^u_j(P'_i, \boldsymbol{q}_{-i})$ and $e_u(Q_h) \prec_u e_u(Q_j)$, i.e., j queues at uv under $(P'_i, \boldsymbol{q}_{-i})$ for a longer time than he does under $(P_i, \boldsymbol{q}_{-i})$ due to h's presence (resp. absence) at uv at the time j reaches u under $(P'_i, \boldsymbol{q}_{-i})$ (resp. $(P_i, \boldsymbol{q}_{-i})$).

(b) Edge $uv \in P'_i \cap Q_j$ and $t^u_i(P'_i, \boldsymbol{q}_{-i}) < t^u_j(P'_i, \boldsymbol{q}_{-i})$, or $t^u_i(P'_i, \boldsymbol{q}_{-i}) = t^u_j(P'_i, \boldsymbol{q}_{-i})$ and $e_u(P'_i) \prec_u e_u(Q_j)$, i.e., the role of h in the above case is played by i here.

In case (a), by the induction hypothesis, i preempts all agents in $\mathcal{A}_u$ and in particular h at vertex u. Thus by (EC.3), we have $\tau^u = t^u_i(O^u_i, \boldsymbol{q}_{-i}) \le \tau^u_h \le t^u_h(P'_i, \boldsymbol{q}_{-i}) \le t^u_j(P'_i, \boldsymbol{q}_{-i})$ and the inequalities hold with equalities only if $e_u(O^u_i) \prec_u e_u(Q_h) \prec_u e_u(Q_j)$. Since $j \notin \mathcal{A}_u$, it follows that $t^u_j(P'_i, \boldsymbol{q}_{-i}) = \tau^u_j$, and further that agent i preempts agent j at vertex u.

In case (b), $\tau^u = t^u_i(O^u_i, \boldsymbol{q}_{-i}) \le t^u_i(P'_i, \boldsymbol{q}_{-i}) \le t^u_j(P'_i, \boldsymbol{q}_{-i}) = \tau^u_j$ and the inequalities hold with equalities only if $e_u(O^u_i) \preceq_u e_u(P'_i) \prec_u e_u(Q_j)$, which shows that i preempts agent j at vertex u. Hence, no matter whether j belongs to $\mathcal{A}_u$ or not, agent i always preempts agent j at vertex u.

Next we prove i preempts j at vertex v. Suppose that path $R_i \in \bar{\mathscr{P}}_i$ satisfies $\tau^v_j = t^v_j(R_i, \boldsymbol{q}_{-i})$. Notice that $\breve{O}_i := O^u_i[o_i, u] \cup Q_j[u, d] \in \bar{\mathscr{P}}_i$. Under the path profile $(\breve{O}_i, \boldsymbol{q}_{-i})$, consider first the case where i moves along edge uv immediately after he reaches u, i.e., there is no queue before him over there. In this case, $t^v_i(\breve{O}_i, \boldsymbol{q}_{-i}) = \tau^u + 1 \le \tau^u_j + 1 \le t^u_j(R_i, \boldsymbol{q}_{-i}) + 1 \le t^v_j(R_i, \boldsymbol{q}_{-i}) = \tau^v_j$. Combining this with the facts that $\tau^v \le t^v_i(\breve{O}_i, \boldsymbol{q}_{-i})$ and $e_v(\breve{O}_i) = uv = e_v(Q_j)$, we can deduce that i preempts j at v. Now we are left with the case where under $(\breve{O}_i, \boldsymbol{q}_{-i})$ agent i spends at least one time unit queuing at uv, i.e., there is a nonempty queue before him at the time he reaches u. Let $\mathcal{B}$ be the set of agents in this queue and those who pass through uv earlier than that queue. Let $h\in\mathcal{B}$ be the last agent in that queue, i.e., he queues at uv right before i: $t_i^v(\breve{O}_i,\boldsymbol{q}_{-i})=t_h^v(\breve{O}_i,\boldsymbol{q}_{-i})+1$. Since $t_i^u(\breve{O}_i,\boldsymbol{q}_{-i})=\tau^u$ (by the definition of $\breve{O}_i$), it follows from Definition EC.1 that i cannot preempt any agent in $\mathcal{B}$ at vertex u. Now as i preempts all agents in $\mathcal{A}_u$ at u by the inductive hypothesis, we see that $\mathcal{B}\cap\mathcal{A}_u=\emptyset$ and further that, no matter how i changes his path, every agent in $\mathcal{B}$ travels along uv at the same time and his arrival time at v is not affected, which gives $\mathcal{B}\cap\mathcal{A}_v=\emptyset$. Thus, $t_h^v(R_i,\boldsymbol{q}_{-i})=\tau_h^v=t_h^v(\breve{O}_i,\boldsymbol{q}_{-i})$. Recall that i preempts j at vertex u and $uv\in Q_h\cap Q_j$. Therefore, no matter how i chooses his path, agent j will always arrive at vertex v at least one time unit later than h. So, by the definition of path R_i, we have $\tau_j^v=t_j^v(R_i,\boldsymbol{q}_{-i})\ge t_h^v(R_i,\boldsymbol{q}_{-i})+1=t_h^v(\breve{O}_i,\boldsymbol{q}_{-i})+1=t_i^v(\breve{O}_i,\boldsymbol{q}_{-i})$. This along with the facts that $\tau^v\le t_i^v(\breve{O}_i,\boldsymbol{q}_{-i})$ and $e_v(\breve{O}_i)=e_v(Q_j)$ implies that agent i preempts agent j at vertex v, as desired. Q.E.D.

Note that what the last paragraph of the above proof does is to derive agent i's preemption over agent j at vertex v from his preemption at vertex u, where uv is an edge of Q_j. This particularly gives the following stronger result.

COROLLARY EC.1. *Given i and $\boldsymbol{q}_{-i}$, if agent i preempts agent $j\in\bar{\Delta}\backslash\{i\}$ at vertex $v\in Q_j$, then i preempts j at all vertices on the subpath $Q_j[v,d]$.*

REMARK EC.1. Edge priorities play an important role in defining the preemption and validating Lemma EC.5 (equivalently, Lemma 1 in Section 4.4.1) and several results that follow from it. The properties implied by Lemma 1 might be invalid if global priorities were placed on agents (as in Scarsini et al. 2018). For example, consider a modification of the game presented in Example 3, where the edge y_2d is subdivided by a newly added vertex. Suppose that the path profile $(P_h,\boldsymbol{q}_{-h})$ is such that agents g and i both choose their upper paths and agent h chooses his lower path. Under this path profile, agent g reaches destination d at time 5, one time unit after agent i. Note that agent h is able to affect g's arrival time at vertex d (decrease it to 4) by switching to his upper path P'_h. However, fixing $\boldsymbol{q}_{-h}$ (i.e., the upper path choices of g and i), agent h is unable to reach d at time 4 or earlier in any case.

EC.5. Computation of EE best-responses

By virtue of Lemma EC.5 established for agent preemptions, we prove in this section the correctness of the Dijkstra-like algorithm presented in Section 4.4.2 for computing EE best-responses.

Given an arbitrarily fixed agent $i\in\bar{\Delta}$ and an arbitrarily fixed partial path profile $\boldsymbol{q}_{-i}$ for other agents in game $\bar{\Gamma}^{\mathrm{N}}$, the EE best-response of agent i to $\boldsymbol{q}_{-i}$ is defined as in Definition 5 with $\bar{\mathscr{P}}_i$ in place of $\mathscr{P}_i$. The agent sets $\mathcal{Q}_e^r$ and $\mathcal{Q}_{e,e'}^r$ given in Definition 6 are now defined w.r.t. $(\bar{G},\bar{\Delta})$ instead of (G,Δ). We have denoted, for each vertex $v\in\bar{V}$, agent i's earliest achievable arrival time at v as $\tau^v := \min\{t_i^v(P_i, \boldsymbol{q}_{-i}) \mid P_i \in \bar{\mathscr{P}}_i\}$. As in Section EC.4, there exists an acyclic complete order on the vertices of $\bar{Y} = \{v \in \bar{V} \mid \tau^v < \infty\}$ such that for each edge of $\bar{E}$ with both end-vertices in $\bar{Y}$, its tail vertex has an order smaller than its head vertex. Recalling the transformation in Section EC.1, it is apparent that the vertices in $\bar{Y}\backslash V$ have smaller orders (if any) than those in $\bar{Y} \cap V = Y = \{v \in V \mid \tau^v < \infty\}$, which is defined in Section 4.4.2. When $\bar{Y}\backslash V \neq \emptyset$, there is only one edge between $\bar{Y}\backslash V$ and Y, i.e., the one incoming to i's origin vertex at G. So, it is clear from Lemma EC.1 that the correctness of the Dijkstra-like algorithm for $\bar{\Gamma}^{\mathrm{N}}$ implies directly its correctness for Γ^{N}.

Since all agents are inside $\bar{G}$ at time 0, the initial setting of our dynamic program is now simplified: If $e = uv$ is the initial edge of agent i, then trivially $\tau^u = 0$, and we initially use the symbol $\mathcal{Q}^0_{e,\hat{e}_u}$ to denote the set of agents in $\mathbb{Q}^0_e$ who queue after i. The following result shows the correctness of the Dijkstra-like algorithm (Algorithm 1) for computing EE best-responses.

Theorem EC.4. *Let E' denote the set of edges on paths in $\bar{\mathscr{P}}_i$. For any vertex $v \in \bar{Y}$ that is not i's starting vertex, it holds that*

$$\tau^v = \min_{u:uv\in E'} \left\{\tau^u + \left|\mathcal{Q}^{\tau^u}_{uv} \backslash \mathcal{Q}^{\tau^u}_{uv,\hat{e}_u}\right| + 1\right\}, \tag{EC.5}$$

where, when u is not i's starting vertex, $\hat{e}_u$ is the edge wu in $\operatorname{argmin}_{wu\in E'} \left\{\tau^w + 1 + \left|\mathcal{Q}^{\tau^w}_{wu} \backslash \mathcal{Q}^{\tau^w}_{wu,\hat{e}_w}\right|\right\}$ *that has the highest priority (w.r.t. $\prec_u$).*

Proof. We prove (EC.5) by induction on the order of those vertices in $\bar{Y}$. The base case where v is the head of i's initial edge is trivial. Let us consider the case where v is not the head of i's initial edge, and suppose (EC.5) is true for vertices $u \in Y$ with orders smaller than v.

We claim that, for every edge $uv \in E'$, no matter how i chooses his path, the arrival times of agents in $\mathcal{Q}^{\tau^u}_{uv} \setminus \mathcal{Q}^{\tau^u}_{uv,\hat{e}_u}$ at vertex u will never be influenced. Suppose the contrary. Then, by Lemma EC.5, agent i preempts at least one agent $j \in \mathcal{Q}^{\tau^u}_{uv} \setminus \mathcal{Q}^{\tau^u}_{uv,\hat{e}_u}$ at vertex u under $\boldsymbol{q}_{-i}$. Note first from the definition of $\mathcal{Q}^{\tau^u}_{uv}$ that $\tau^u_j \leq \tau^u$, where τ^u_j is the earliest time j can reach u when i changes his path. By Definition EC.1, it can only be the case that $\tau^u_j = \tau^u$ and i is able to arrive at u at time τ^u via an edge e' that has a priority no lower than the one taken by j. By induction hypothesis, $\tau^u = \min_{wu\in E'} \left\{\tau^w + 1 + \left|\mathcal{Q}^{\tau^w}_{wu} \backslash \mathcal{Q}^{\tau^w}_{wu,\hat{e}_w}\right|\right\}$; in turn the definition of $\hat{e}_u$ implies that the priority of $\hat{e}_u$ is not lower than that of e', and hence not lower than that of the edge taken by j. However, this is impossible because $j \notin \mathcal{Q}^{\tau^u}_{uv,\hat{e}_u}$. Hence the claim is valid. Therefore, regardless of i's choice, all agents in $\mathcal{Q}^{\tau^u}_{uv} \setminus \mathcal{Q}^{\tau^u}_{uv,\hat{e}_u}$ arrive at u no later than τ_u and those arriving at time τ^u (if any) use incoming edges to u with priorities higher than edge $\hat{e}_u$. It follows from the definition of $\tau^u = \min\{t_i^u(P_i, \boldsymbol{q}_{-i}) \mid P_i \in \bar{\mathscr{P}}_i\}$, induction hypothesis on u and definition of $\hat{e}_u$ that i cannot move along uv until all agents in $\mathcal{Q}^{\tau^u}_{uv} \setminus \mathcal{Q}^{\tau^u}_{uv,\hat{e}_u}$ exit uv.

Consequently, if agent i uses edge $uv \in E'$ to reach v, his arrival time at v is at least $\tau_u + |\mathcal{Q}_{uv}^{\tau^u} \setminus \mathcal{Q}_{uv,\hat{e}_u}^{\tau^u}| + 1$. On the other hand, by the induction hypothesis, this value is obtainable by reaching u at time τ^u via $\hat{e}_u$. It follows that the earliest time i can reach v via edge uv is exactly $\tau_u + |\mathcal{Q}_{uv}^{\tau^u} \setminus \mathcal{Q}_{uv,\hat{e}_u}^{\tau^u}| + 1$. Since i must use one edge $uv \in E'$ to reach v, the correctness of (EC.5) is established. Q.E.D.

The subgraph of $\bar{G}$ spanned by all edges $\hat{e}_v$, $v \in \bar{Y} \setminus \{o_i\}$ defined in Theorem EC.4 contains a unique o_i-d path. By Definition 5, it is the EE best-response of agent i to $\boldsymbol{q}_{-i}$.

EC.6. Characterization of NEs

In this section, we first make some observations on agent interactions, then establish the iterative batch-dominance characterization of all NEs of game $\bar{\Gamma}^{\mathrm{N}}$ and hence Γ^{N}.

EC.6.1. Agent precedence

We investigate the precedence relations between agents under the same (partial) routing of $\bar{\Gamma}^{\mathrm{N}}$. These relations are much more direct and visible than the preemption relations (see Definition EC.1), which generally involve two different routings.

Given game $\bar{\Gamma}^{\mathrm{N}}$ on $(\bar{G}, \bar{\Delta})$, every (partial) path profile $\boldsymbol{q}_S = (Q_i)_{i \in S}$ of $\bar{\Gamma}^{\mathrm{N}}$ for agents in $S \subseteq \bar{\Delta}$ is often considered as a routing for the game restricted to agents in S, where each agent $i \in S$ follows Q_i. For any agent $i \in S$ and vertex $v \in \bar{G}$, we use $t_i^v(\boldsymbol{q}_S)$ to denote agent i's arrival time at v under routing $\boldsymbol{q}_S$.

Definition EC.2 (Precedence). Given a (partial) path profile $\boldsymbol{q}_S$ of game $\bar{\Gamma}^{\mathrm{N}}$, and agents $i, j \in S$, we say that agent i *strongly precedes* agent j through vertex v under $\boldsymbol{q}_S$ at time $t_i^v(\boldsymbol{q}_S)$ if under routing $\boldsymbol{q}_S$ they both pass v and i reaches v earlier than j. We say that i *precedes* j through vertex v under $\boldsymbol{q}_S$ at time $t_i^v(\boldsymbol{q}_S)$ if either i strongly precedes j through vertex v, or i and j reach v at the same time but i comes from an edge (incoming to v) with a higher priority than the edge from which agent j comes.

Observe from the above definition that if agent i precedes agent j through a vertex u and both i and j choose to enter the same edge uv, then i strongly precedes j through vertex v. It is possible that agent i strongly precedes agent j through some vertex and j strongly precedes i through another vertex, even under NEs (see Example EC.2 in Section EC.12). We emphasize again that while the notion of preemption (Definition EC.1) compares the arrival times of two agents at the same vertex under possibly *different* path profiles, precedence compares two arrival times under the *same* (partial) path profile. Unlike the Braess-like paradox presented in Example 4, as far as precedence is concerned, the following lemma accords with the intuition that fewer agents lead to faster travel.

LEMMA EC.6. *Let S and T be agent subsets with $\emptyset \neq S \subset T \subseteq \bar{\Delta}$, and $\boldsymbol{q}_T$ be a partial path profile for agents in T. If under $\boldsymbol{q}_T$ some agent in $T \backslash S$ precedes an agent in S at some time τ, then there exists agent $i \in T \backslash S$ such that under $\boldsymbol{q}_{S\cup\{i\}}$ agent i precedes some agent in S no later than τ.*

Proof. Suppose that under $\boldsymbol{q}_T$, agent $i \in T\backslash S$ precedes agent $j \in S$ through some vertex v, and further that $t_i^v(\boldsymbol{q}_T)$ is as small as possible. The minimality implies that $t_i^v(\boldsymbol{q}_T) \leq \tau$, and under $\boldsymbol{q}_T$ no agent in $T\backslash(S\cup\{i\})$ precedes any agent in S before time $t_i^v(\boldsymbol{q}_T)$. So removing $\boldsymbol{q}_{T\backslash(S\cup\{i\})}$ (i.e., removing agents of $T\backslash(S\cup\{i\})$ and their paths) from routing $\boldsymbol{q}_T$ can only possibly reduce i's queuing time before the time when he reaches v, accelerating his arrival time at v, which implies $t_i^v(\boldsymbol{q}_{S\cup\{i\}}) \leq t_i^v(\boldsymbol{q}_T) \leq \tau$. Moreover, since under $\boldsymbol{q}_T$ before time $t_i^v(\boldsymbol{q}_{S\cup\{i\}}) \leq t_i^v(\boldsymbol{q}_T)$, all agents of $T\backslash(S\cup\{i\})$ run after or reach no common vertices with all agents of S, we see that removing $\boldsymbol{q}_{T\backslash(S\cup\{i\})}$ does not change the routing status of agents in S before time $t_i^v(\boldsymbol{q}_{S\cup\{i\}})$. Therefore, i precedes j through v under $\boldsymbol{q}_{S\cup\{i\}}$ at time $t_i^v(\boldsymbol{q}_{S\cup\{i\}}) \leq \tau$. Q.E.D.

EC.6.2. Characterization of iterative batch-dominance

Building on the lemmas (established in Sections EC.4 and EC.6.1) for agent preemption and precedence, we prove the NE characterization in this subsection. The notation and definitions presented in Section 4.3 apply directly to $\bar{\Gamma}^{\mathrm{N}}$, with the only symbolic replacement of Δ by $\bar{\Delta}$ to indicate that we are in the setting of $\bar{\Gamma}^{\mathrm{N}}$. For example, the kth batch of a routing $\boldsymbol{q}$ for $\bar{\Gamma}^{\mathrm{N}}$ is written as $\bar{\Delta}(\boldsymbol{q},k)$.

LEMMA EC.7. *Let $\boldsymbol{p} = (P_h)_{h\in\bar{\Delta}}$ be an NE of game $\bar{\Gamma}^{\mathrm{N}}$. For every $k \geq 1$ and every agent $j \in \bar{\Delta} \setminus \bar{\Delta}(\boldsymbol{p},[k])$, agent j cannot preempt any agent $i \in \bar{\Delta}(\boldsymbol{p},[k])$ at any vertex of path P_i under $\boldsymbol{p}_{-j}$.*

Proof. Suppose on the contrary that agent $j \in \bar{\Delta}\backslash\bar{\Delta}(\boldsymbol{p},[k])$ preempts agent $i \in \bar{\Delta}(\boldsymbol{p},[k])$ at some vertex of P_i under $\boldsymbol{p}_{-j}$. Then from Corollary EC.1 (with i and j switching their roles over there), we deduce that under $\boldsymbol{p}_{-j}$ agent j also preempts agent i at vertex d. This means that there exists a path $P_j^* \in \bar{\mathscr{P}}_j$ such that

$$t_j^d(P_j^*, \boldsymbol{p}_{-j}) \leq \min_{R_j\in\bar{\mathscr{P}}_j}\{t_i^d(R_j,\boldsymbol{p}_{-j})\} \leq t_i^d(\boldsymbol{p}) \leq \tau(\boldsymbol{p},k).$$

However, $t_j^d(\boldsymbol{p}) > \tau(\boldsymbol{p},k)$ due to $j \in \bar{\Delta}\backslash\bar{\Delta}(\boldsymbol{p},[k])$, indicating that j has an incentive to switch to P_j^*, which violates the fact that $\boldsymbol{p}$ is an NE. Q.E.D.

Given a partial path profile $\boldsymbol{q}_S = (Q_i)_{i\in S}$ of $\bar{\Gamma}^{\mathrm{N}}$ on agent set $S \subseteq \bar{\Delta}$, for every agent $j \in S$ and vertex $v \in Q_j$, we consider $(Q_j[o_j,v], \boldsymbol{q}_{S\backslash\{j\}})$ as the (incomplete) routing in which j follows $Q_j[o_j,v]$ and agents in $S\backslash\{j\}$ follow $\boldsymbol{q}_{S\backslash\{j\}} = (Q_i)_{i\in S\backslash\{j\}}$. It is clear that for every vertex $u \in Q_j[o_j,v]$, the arrival time of agent j at u under $(Q_j[o_j,v], \boldsymbol{q}_{S\backslash\{j\}})$, denoted as $t_j^u(Q_j[o_j,v], \boldsymbol{q}_{S\backslash\{j\}})$, is the same as that under $\boldsymbol{q}_S$, i.e., $t_j^u(\boldsymbol{q}_S)$.

LEMMA EC.8. *Let $\boldsymbol{p} = (P_h)_{h\in\bar{\Delta}}$ be an NE of game $\bar{\Gamma}^{\mathrm{N}}$. For any batch index $k \geq 1$, agent $i \in \Omega := \bar{\Delta}(\boldsymbol{p}, [k])$, vertex $v \in P_i$, agent $j \in \bar{\Delta}\backslash\Omega$, and partial path profile $\boldsymbol{q}_{-\Omega}$ for agents in $\bar{\Delta}\backslash\Omega$, the following hold:*

$$t_i^v(\boldsymbol{p}) = t_i^v(\boldsymbol{p}_\Omega) = t_i^v(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}) \leq t_j^v(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}), \tag{EC.6}$$

$$t_j^d(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}) \geq \tau(\boldsymbol{p}, k+1) > t_i^d(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}). \tag{EC.7}$$

Proof. For each agent $j \in \bar{\Delta}\backslash\Omega$, define r_j as the earliest time when j can precede (recall Definition EC.2) some agent of Ω under (partial) path profile $(\boldsymbol{p}_\Omega, R_j)$ among all paths $R_j \in \bar{\mathscr{P}}_j$. If for any $R_j \in \bar{\mathscr{P}}_j$, under $(\boldsymbol{p}_\Omega, R_j)$ agent j can never precede any agent in Ω, we set $r_j := \infty$. Define

$$r_* := \min\{r_j \mid j \in \bar{\Delta}\backslash\Omega\}.$$

It follows from Lemma EC.6 that for any agent subset $S \subseteq \bar{\Delta}\backslash\Omega$ and any partial path profile $\boldsymbol{x}_S$ of agents in S,

$$\text{Under } (\boldsymbol{p}_\Omega, \boldsymbol{x}_S) \text{ no agent of } S \text{ can precede any agent of } \Omega \text{ before time } r_*. \tag{EC.8}$$

Validity of (EC.6) is implied by $r_* = \infty$. Indeed, if $r_* = \infty$, then applying (EC.8) with $S = \bar{\Delta}\backslash\Omega$ and $\boldsymbol{x}_S = \boldsymbol{q}_{-\Omega}$, Definition EC.2 directly gives the inequality in (EC.6). The equalities in (EC.6) will also be valid, because as long as agents in Ω follow $\boldsymbol{p}_\Omega$, they are not affected by the remaining agents, none of whom can precede agents in Ω.

Suppose on the contrary that $r_* < \infty$. By the definition of r_*, there exist agent $i \in \Omega$, agent $j_* \in \bar{\Delta}\backslash\Omega$, path $R_{j_*} \in \bar{\mathscr{P}}_{j_*}$ and vertex $v \in P_i \cap R_{j_*}$ such that under $(\boldsymbol{p}_\Omega, R_{j_*})$ agent j_* precedes agent i through vertex v at time

$$t_j^v(\boldsymbol{p}_\Omega, R_{j_*}) = r_*.$$

Therefore, there exists vertex $u \in P_i[o_i, v]$ such that under $(\boldsymbol{p}_\Omega, R_{j_*})$ agent i reaches u at time $t_i^u(\boldsymbol{p}_\Omega, R_{j_*}) = r_*$. Moreover, applying (EC.8) with $\boldsymbol{x}_S = R_{j_*}$ and $\boldsymbol{x}_S = \boldsymbol{q}_{-\Omega}$, respectively, we derive

$$t_i^u(\boldsymbol{p}_\Omega, R_{j_*}) = r_* = t_i^u(\boldsymbol{p}_\Omega) \quad \text{and} \quad t_i^u(\boldsymbol{p}_\Omega) = r_* = t_i^u(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}).$$

The trivial relation $t_i^v(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}) \geq t_i^u(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega})$ (as $u \in P_i[o_i, v]$) and the precedence of j_* over i through v give the following:

$$\begin{array}{l} t_i^v(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}) \geq r_* \text{ for any partial path profile } \boldsymbol{q}_{-\Omega} \text{ of agents in } \bar{\Delta}\backslash\Omega, \text{ and} \\ e_v(R_{j_*}) \prec e_v(P_i) \text{ if } u = v. \end{array} \tag{EC.9}$$

Moreover, notice from (EC.8) that as long as agents in Ω follow $\boldsymbol{p}_\Omega$, from time 0 till time r_*, the arrival times of all agents in Ω at the corresponding vertices are invariant against route changes of agents outside Ω. These invariant arrival times lead to invariant influence of agents in Ω on agents in

$\bar{\Delta}\backslash\Omega$ till time r_*. Therefore, we may define $j \in \bar{\Delta}\backslash\Omega$, using an adaptation of Algorithm 3 with vertex v (resp. Ω and $\boldsymbol{p}_\Omega$) in place of destination d (resp. U and $\boldsymbol{b}$) over there, as the "dominator" agent of $\bar{\Delta}\backslash\Omega$ (the first agent output by the adaptation) who is associated with an o_j-v path $\bar{Q}$ starting with the initial edge of j. Recalling that $t^v_{j*}(\boldsymbol{p}_\Omega, R_{j*}) = r_*$, the dominance of j gives $t^v_j(\boldsymbol{p}_\Omega, \bar{Q}) \le r_*$. In turn, the minimality of r_* enforces $t^v_j(\boldsymbol{p}_\Omega, \bar{Q}) = r_*$, which along with the dominance of j implies

$$e_v(\bar{Q}) \preceq e_v(R_{j*}). \tag{EC.10}$$

Combining $t^v_j(\boldsymbol{p}_\Omega, \bar{Q}) = r_*$ and Ω's invariant influence on $\bar{\Delta}\backslash\Omega$ till time r_*, we deduce as in Lemma EC.3 that, assuming $\boldsymbol{p}_\Omega$, the "dominator" agent j is not preceded by any agent in $\bar{\Delta}\backslash(\Omega \cup \{j\})$ when he travels along $\bar{Q}$, regardless of the choices of agents in $\bar{\Delta}\backslash(\Omega \cup \{j\})$. In particular, we have $t^v_j(\bar{Q}, \boldsymbol{p}_{-j}) = r_*$. Define path $Q_j := \bar{Q} \cup P_i[v,d] \in \bar{\mathscr{P}}_j$. Then $t^v_j(Q_j, \boldsymbol{p}_{-j}) = r_*$, and it follows from (EC.9) and (EC.10) that under $(Q_j, \boldsymbol{p}_{-j})$ agent j precedes agent i through vertex v at time r_*. Thus j arrives at d no later than i under routing profile $(Q_j, \boldsymbol{p}_{-j})$, i.e., $t^d_j(Q_j, \boldsymbol{p}_{-j}) \le t^d_i(Q_j, \boldsymbol{p}_{-j})$, because of $Q_j[v,d] = P_i[v,d]$. (Note equation $t^d_j(Q_j, \boldsymbol{p}_{-j}) = t^d_i(Q_j, \boldsymbol{p}_{-j})$ holds only when $v = d$.)

Now we turn our attention from precedence (Definition EC.2) to preemption (Definition EC.1). If $t^w_i(Q_j, \boldsymbol{p}_{-j}) \neq t^w_i(\boldsymbol{p})$ for some vertex $w \in P_i$, then by Lemma EC.5 agent j preempts i at w under $\boldsymbol{p}_{-j}$, which is a contradiction to Lemma EC.7. We are left with the case where $t^w_i(Q_j, \boldsymbol{p}_{-j}) = t^w_i(\boldsymbol{p})$ holds for all vertices $w \in P_i$. It follows that $t^d_j(Q_j, \boldsymbol{p}_{-j}) \le t^d_i(Q_j, \boldsymbol{p}_{-j}) = t^d_i(\boldsymbol{p}) = \tau(\boldsymbol{p}, k) < t^d_j(\boldsymbol{p})$, where the last inequality follows from $j \notin \Omega$. However, $t^d_j(Q_j, \boldsymbol{p}_{-j}) < t^d_j(\boldsymbol{p})$ contradicts the fact that $\boldsymbol{p}$ is an NE. This proves the correctness of (EC.6).

Now let us prove (EC.7). Once the agents in Ω have chosen their paths as specified by $\boldsymbol{p}_\Omega$, thanks to (EC.6) about the invariant influence of Ω on $\bar{\Delta}\backslash\Omega$, we can apply Algorithm 3 with $U := \Omega$ and $\boldsymbol{b} := \boldsymbol{p}_\Omega$, which provides us a dominator $f \in \bar{\Delta}\backslash\Omega$ and his associated path $\bar{P}_f \in \bar{\mathscr{P}}_f$ such that (by Lemma EC.3) $t^d_f(\boldsymbol{p}_{-f}, \bar{P}_f) = t^d_f(\boldsymbol{p}_\Omega, \bar{P}_f) \le t^d_f(\boldsymbol{p})$ and $t^d_j(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}) \ge t^d_f(\boldsymbol{p}_\Omega, \bar{P}_f)$ for any $j \in \bar{\Delta}\backslash\Omega$ and partial path profile $\boldsymbol{q}_{-\Omega}$ of $\bar{\Delta}\backslash\Omega$.

Since $\boldsymbol{p}$ is an NE of $\bar{\Gamma}^{\mathrm{N}}$, we have $t^d_f(\boldsymbol{p}_{-f}, \bar{P}_f) \ge t^d_f(\boldsymbol{p})$, and hence $t^d_f(\boldsymbol{p}_\Omega, \bar{P}_f) = t^d_f(\boldsymbol{p}) \ge \tau(\boldsymbol{p}, k+1)$, where the last inequality is due to $f \notin \Omega$. On the other hand, $t^d_f(\boldsymbol{p}_\Omega, \bar{P}_f) \le \min\{t^d_h(\boldsymbol{p}) \mid h \in \bar{\Delta}\backslash\Omega\} = \tau(\boldsymbol{p}, k+1)$, from which we deduce that $t^d_j(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}) \ge t^d_f(\boldsymbol{p}_\Omega, \bar{P}_f) = \tau(\boldsymbol{p}, k+1)$, yielding the first inequality in (EC.7). The second inequality in (EC.7) follows from $t^d_i(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega}) \le \tau(\boldsymbol{p}, k)$, which is guaranteed by the equalities in (EC.6). Q.E.D.

We are ready to prove Theorem 2 in the language of game $\bar{\Gamma}^{\mathrm{N}}$ (recalling Lemma EC.1).

Theorem EC.5. *A path profile is an NE for $\bar{\Gamma}^{\mathrm{N}}$ if and only if it is iteratively batch-dominant.*

Proof. By Lemma EC.8, it suffices to prove the "if" part. Suppose $\boldsymbol{q}$ is an iteratively batch-dominant path profile of $\bar{\Gamma}^{\mathrm{N}}$ as specified in Definition 3. Consider an arbitrary agent $j \in \bar{\Delta}$ and suppose he belongs to the kth batch $\bar{\Delta}(\boldsymbol{q}, k)$, i.e., $t_j^d(\boldsymbol{q}) = \tau(\boldsymbol{q}, k)$. For any $R_j \in \bar{\mathscr{P}}_j$, it follows from Definition 3 that $t_j^d(\boldsymbol{q}_{-j}, R_j) \geq \tau(\boldsymbol{q}, k) = t_j^d(\boldsymbol{q})$, which states that $\boldsymbol{q}$ is indeed an NE of $\bar{\Gamma}^{\mathrm{N}}$. Q.E.D.

EC.7. More NE properties

In this section, we first verify that every NE of game $\bar{\Gamma}^{\mathrm{N}}$ (equivalently game Γ^{N}) possesses the properties that have been mentioned in Section 4.3. Then we discuss more NE properties implied by the EE best-response and global FIFO.

THEOREM EC.6. *Let $\boldsymbol{p}$ be an NE of game $\bar{\Gamma}^{\mathrm{N}}$. The following properties are satisfied.*

(i) Hierarchical independence. *If agents in a batch and those in earlier batches all follow their equilibrium strategies as in $\boldsymbol{p}$, then their arrival times at any vertex are independent of other agents' strategies.*

(ii) Hierarchal optimality. *The arrival time of each agent in the first batch $\bar{\Delta}(\boldsymbol{p}, 1)$ is the smallest among the arrival times of all agents under any routing of $\bar{\Gamma}^{\mathrm{N}}$. In general, for all $k \geq 2$, the arrival time of each agent in the kth batch $\bar{\Delta}(\boldsymbol{p}, k)$ is the smallest among the arrival times of all agents outside the first $k-1$ batches (i.e., those in $\bar{\Delta} \setminus \bar{\Delta}(\boldsymbol{p}, [k-1])$) under any routing of $\bar{\Gamma}^{\mathrm{N}}$ in which agents in the first $k-1$ batches $\bar{\Delta}(\boldsymbol{p}, [k-1])$ follow their routes specified by $\boldsymbol{p}$.*

(iii) General FIFO. *Under $\boldsymbol{p}$, if agent i precedes agent j through some vertex (see Definition EC.2), then i reaches the destination d no later than j. (Apparently, the property of general FIFO includes the global FIFO as a special case.)*

(iv) Strong NE. *Profile $\boldsymbol{p}$ is a strong NE of game $\bar{\Gamma}^{\mathrm{N}}$, and thus it is weakly Pareto optimal.*

Proof. (i) The hierarchical independence is simply an interpretation of $t_i^v(\boldsymbol{p}) = t_i^v(\boldsymbol{p}_\Omega, \boldsymbol{q}_{-\Omega})$ with $\Omega = \bar{\Delta}(\boldsymbol{p}, [k])$ for each $k \geq 1$ in Lemma EC.8.

(ii) For each $k \geq 1$, let $\Omega := \bar{\Delta}(\boldsymbol{p}, [k-1])$, and let r^* denote the *earliest* time an agent in $\bar{\Delta}\backslash\Omega$ reaches d among all routings of $\bar{\Gamma}^{\mathrm{N}}$ in which agents in Ω take their routes as in $\boldsymbol{p}_\Omega$. We need to verify the hierarchical optimality that $\tau(\boldsymbol{p}, k) = r^*$. Clearly,

$$r^* \leq \tau(\boldsymbol{p}, k).$$

The equalities in (EC.6) (i.e., Ω's invariant influences on $\bar{\Delta}\backslash\Omega$) enable us to apply Algorithm 3 and Lemma EC.3, which provides us a dominator agent $i \in \bar{\Delta}\backslash\Omega$, who is associated with a path $\bar{P}_i \in \bar{\mathscr{P}}_i$, provided the agents in Ω follow their routes as in $\boldsymbol{p}_\Omega$. It follows from Lemma EC.3 that $t_i^d(\boldsymbol{p}_{-i}, \bar{P}_i) = \min\{t_i^v(\boldsymbol{p}_\Omega, R_i) \mid R_i \in \bar{\mathscr{P}}_i\} \leq r^*$. On the other hand, since i cannot be better off by switching to $\bar{P}_i$, we have $t_i^d(\boldsymbol{p}_{-i}, \bar{P}_i) \geq t_i^d(\boldsymbol{p}) \geq r^*$. Therefore, $t_i^d(\boldsymbol{p}) = r^*$, which along with $r^* \leq \tau(p, k) \leq t_i^d(\boldsymbol{p})$ enforces $\tau(\boldsymbol{p}, k) = r^*$ as desired.

(iii) If under $\boldsymbol{p}=(P_h)_{h\in\bar{\Delta}}$ agent i reaches d later than agent j, i.e., $i\notin\bar{\Delta}(\boldsymbol{p},k)\ni j$ for some k, then by Lemma EC.8 we claim that

$$\text{For every } v\in P_j \text{ and every } Q_i\in\bar{\mathscr{P}}_i, \text{ it holds that } t_j^v(\boldsymbol{p})=t_j^v(\boldsymbol{p}_{-i},Q_i)\le t_i^v(Q_i,\boldsymbol{p}_{-i}).$$

As i precedes j at some vertex u, it must be the case that $t_i^u(\boldsymbol{p})=t_j^u(\boldsymbol{p})$ and $e_u(P_i)\prec_u e_u(P_j)$. As $t_i^d(\boldsymbol{p})>t_j^d(\boldsymbol{p})$, we see that $u\ne d$ and suppose that uw is the outgoing edge from u on P_j. If $uw\in P_i$, then at time $t_i^u(\boldsymbol{p})$ agent i queues before agent j at edge uw, yielding $t_j^w(\boldsymbol{p})\ge t_i^w(\boldsymbol{p})+1$, a contradiction to the above claim. So we are left with the case of $uw\notin P_i$. Considering $Q_i:=P_i[o_i,u]\cup P_j[u,d]\in\bar{\mathscr{P}}_i$ with $w\in Q_i$, we have $t_j^w(Q_i,\boldsymbol{p}_{-i})=t_i^w(Q_i,\boldsymbol{p}_{-i})+1$ (because the capacity of edge uw is 1, and i comes from the edge $e_u(Q_i)=e_u(P_i)$ with a higher priority than edge $e_u(P_j)$), which is a contradiction to the above claim. So we have proved the general FIFO property.

(iv) Suppose on the contrary that there exists a set $S\subseteq\bar{\Delta}$ of agents who can be strictly better off through collectively deviating from an NE $\boldsymbol{p}$ of game $\bar{\Gamma}^{\mathrm{N}}$. Let k be the smallest batch index such that $S\cap\bar{\Delta}(\boldsymbol{p},k)\ne\emptyset$. Due to the hierarchical optimality stated in (ii), all agents in $\bar{\Delta}(\boldsymbol{p},k)$ obtain their earliest arrival times since no agent in $\bar{\Delta}(\boldsymbol{p},[k-1])$ deviates from $\boldsymbol{p}$, a contradiction. Q.E.D.

According to Theorem EC.4, in game $\bar{\Gamma}^{\mathrm{N}}$ every agent possesses a best response that is a path for the earliest arrival. This implies the following NE property.

Corollary EC.2 **(Weak earliest arrival)**. *Any NE for a new game building on $\bar{\Gamma}^{\mathrm{N}}$ with the additional restriction that all agents take earliest-arrival paths is still an NE of the game $\bar{\Gamma}^{\mathrm{N}}$ that does not have this restriction.*

For ease of exposition, in the next corollary, we restrict our attention to game Γ^{N} on network $G=(V,E)$ with a single origin. Recalling the original ranks defined in Section 3, let agents in Δ be indexed as $1,2,\ldots$ according to their entry times into G and their original ranks (smaller indices correspond to earlier entry times and higher ranks in the case of equal entry time). We have the following straightforward corollary of the global FIFO property stated in Theorem 3 or in Theorem EC.6(iii).

Corollary EC.3. *If $\boldsymbol{p}$ is an NE of Γ^{N} with a single origin o, then the following properties are satisfied:*

(i) Consecutive exiting. *The indices of agents within the same batch under $\boldsymbol{p}$ are consecutive. That is, if $i,j\in\Delta(\boldsymbol{p},k)$ with $i<j$, then $h\in\Delta(\boldsymbol{p},k)$ for all $i\le h\le j$.*

(ii) Temporal overtaking. *If under $\boldsymbol{p}$ agent j strongly precedes agent i $(<j)$ at some vertex $v\in V\setminus\{o\}$, i.e., j reaches v earlier than i, then under $\boldsymbol{p}$ they reach the destination d at the same time.*

When focusing on agents originating from the same origin, the above two properties can be extended to networks with multiple origins.

EC.8. Actions and consecutive configurations in game Γ^{A}

Given configuration $\mathsf{c}_r = (\mathbb{Q}_e^r)_{e\in E}$, the *action set* of agent $i \in \Delta(\mathsf{c}_r) = (\cup_{e\in E}\mathbb{Q}_e^r) \cup (\cup_{v\in V}\Delta_{r+1,v})$, denoted by $E(i,\mathsf{c}_r)$, is defined as follows. If $i \in \Delta_{r+1,v}$, then $E(i,\mathsf{c}_r) = E^+(v)$. Suppose $i \in \mathbb{Q}_e^r$, where $e = uv$.

- If $v = d$ and i queues first in $\mathbb{Q}_e^r$, then $E(i,\mathsf{c}_r) := \emptyset$, i.e., i simply exits G at time $r+1$ (from d).
- If $v \neq d$ and i queues first in $\mathbb{Q}_e^r$, then $E(i,\mathsf{c}_r) := E^+(v)$, i.e., agent i selects the next edge that is available at v.
- Otherwise (i.e., i is not the head of $\mathbb{Q}_e^r$), agent i has to stay at e with $E(i,\mathsf{c}_r) := \{e\}$.

Given a configuration c_r and an action profile $\boldsymbol{a} = (a_i)_{i\in\Delta(\mathsf{c}_r)}$ with $a_i \in E(i,\mathsf{c}_r)$, the edge-priority DQ rule leads to a new configuration $\mathsf{c}_{r+1} = (\mathbb{Q}_e^{r+1})_{e\in E}$ at time $r+1$, referred to as a *consecutive configuration* of c_r:

- As a set, $\mathbb{Q}_e^{r+1} = \{i \in \Delta(\mathsf{c}_r) \,|\, a_i = e\}$ consists of agents choosing e in action profile $\boldsymbol{a}$.
- As a sequence, $\mathbb{Q}_e^{r+1}$ is obtained from $\mathbb{Q}_e^r$ by removing its head and making its tail followed by agents in $\mathbb{Q}_e^{r+1}\backslash\mathbb{Q}_e^r$ whose positions are determined according to the priority order $\prec_u$ at the tail vertex u of edge $e = uv$.

EC.9. Construction of a special SPE

This section is devoted to proving the SPE existence in game Γ^{A}, which has been discussed in Section 5.2. We call the normal-form game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$ introduced in Section 5.2 the *intermediary game of* Γ^{N} *starting from* c_r. For any time point $r \geq 0$, let C_r denote the set of all possible configurations at time r; in particular $\mathsf{C}_0 = \{\mathsf{c}_0\}$ consists of the unique initial configuration given by initial queues in G at time 0.

Proof of Theorem 5. Given any history $\mathsf{h}_r = (\mathsf{c}_0, \ldots, \mathsf{c}_r) \in \mathsf{H}_r$ for any time point $r \geq 0$, recall that $\mathscr{D}(\mathsf{c}_r) = \Delta(\mathsf{c}_r) \cup (\cup_{s\geq r+2, v\in V}\Delta_{s,v})$ is the agent set of game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$. According to Lemma EC.1, let $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$ denote the game instance of model $\bar{\Gamma}^{\mathrm{N}}$ transformed from game $\Gamma^{\mathrm{N}}(c_r)$ using (T1)–(T3). So the agent set of $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$ is $\mathscr{D}(\mathsf{c}_r)$, and the restriction of $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$ to G is $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$. Suppose that the agents in $\mathscr{D}(\mathsf{c}_r)$ are named as $1_r, 2_r, \ldots$ such that agent i_r is the ith agent added to D in Step 13 of Algorithm 2 with input being the game instance $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$. For each agent $i_r \in \mathscr{D}(\mathsf{c}_r)$, let $\bar{P}_{i_r}^{\mathsf{c}_r}$ denote the dominant path associated to i_r in Algorithm 2.

Consider any agent $i_r \in \Delta(\mathsf{c}_r) = (\cup_{e\in E}\mathbb{Q}_e^r) \cup (\cup_{v\in V}\Delta_{r+1,v})$. Note that at the beginning of $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$, agent i_r queues at the first edge of $\bar{P}_{i_r}^{\mathsf{c}_r}$. If $i_r \in \cup_{e\in E}\mathbb{Q}_e^r$, then $\bar{P}_{i_r}^{\mathsf{c}_r}$ is a path in G; otherwise, the first edge of $\bar{P}_{i_r}^{\mathsf{c}_r}$ is the only edge of $\bar{P}_{i_r}^{\mathsf{c}_r}$ that is outside G and i_r is the only agent queuing, at the beginning of $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$, at that edge (i.e., he will enter G at the next time point). A configuration in C_{r+1} will result from c_r according to action profile $\boldsymbol{a}^{\mathsf{c}_r}$ defined as follows:

$$\text{The action of } i_r = \begin{cases} \text{the first edge of } \bar{P}_{i_r}^{\mathsf{c}_r}, & \text{if } i \text{ queues after another agent;} \\ \emptyset, & \text{if } i \text{ will exit } G \text{ from } d \text{ at the next time point;} \\ \text{the second edge of } \bar{P}_{i_r}^{\mathsf{c}_r}, & \text{otherwise;} \end{cases}$$

where the second condition is equivalent to i queuing first at the last edge of $\bar{P}_{i_r}^{\mathsf{c}_r}$. Observe that in any case the action defined above (if not $\emptyset$) is an edge of graph G. The set $\cup_{r\geq 0}\cup_{\mathsf{c}_r\in\mathsf{C}_r}\boldsymbol{a}^{\mathsf{c}_r}$ of action profiles defines a strategy profile $\sigma^*=(\sigma_i^*)_{i\in\Delta}$ of Γ^{A}. We will prove that σ^* is an SPE of game Γ^{A}.

Let $(\mathsf{c}_r,\mathsf{c}_{r+1},\ldots)$ be the list of configurations and $(P_i^*)_{i\in\mathscr{D}(\mathsf{c}_r)}$ be the path profile induced by h_r and σ^*. It can be deduced from Lemma EC.2 and Algorithm 2 that

- For any $s\geq r+1$, agent sequence $(1_s,2_s,\ldots)$ is a subsequence of $(1_{s-1},2_{s-1},\ldots)$ such that $\mathscr{D}(\mathsf{c}_{s-1})\backslash\mathscr{D}(\mathsf{c}_s)$ consists of the first $|\mathscr{D}(\mathsf{c}_{s-1})\backslash\mathscr{D}(\mathsf{c}_s)|$ agents of $1_{s-1},2_{s-1},\ldots$;
- For any $s\geq r+1$ and $i\in\Delta(\mathsf{c}_s)\backslash\cup_{v\in V}\Delta_{s+1,v}$, $\bar{P}_i^{\mathsf{c}_s}$ is a subpath of $\bar{P}_i^{\mathsf{c}_{s-1}}$ ($\bar{P}_i^{\mathsf{c}_s}$ is either $\bar{P}_i^{c_{s-1}}$ or $\bar{P}_i^{c_{s-1}}$ with its first vertex and edge removed).

Therefore, the path P_i^* formed by the actions of each agent $i\in\mathscr{D}(\mathsf{c}_r)$ is exactly the restriction of $\bar{P}_i^{\mathsf{c}_r}$ to G. According to Lemma EC.2 (the equation in (EC.2)), we have

$$t_{i_r}(\sigma^*|\mathsf{h}_r)=r+\min\left\{t_{i_r}^d(\bar{P}_{1_r}^{\mathsf{c}_r},\ldots,\bar{P}_{(i-1)_r}^{\mathsf{c}_r},R_{i_r})\,|\,R_{i_r}\in\bar{\mathscr{P}}_{i_r}\right\}\text{ for every } i\geq 1.$$

Moreover, for any $j\geq 1$ and any strategy profile σ' of Γ^{A} with $\sigma'_{i_r}=\sigma^*_{i_r}$ for all $i\in[j]$, considering the path profile $(P'_i)_{i\in\mathscr{D}(\mathsf{c}_r)}$ induced by h_r and σ', we can deduce from an inductive argument that for each $i=1,\ldots,j$, P'_{i_r} is exactly $P^*_{i_r}$, i.e., the restriction of $\bar{P}_{i_r}^{\mathsf{c}_r}$ to G.

Now given any $k\geq 1$ and any $\sigma'_{k_r}\in\Sigma_{k_r}$, we consider strategy profile $\sigma'=(\sigma'_{k_r},\sigma^*_{-k_r})$ and the path profile $\boldsymbol{p}'=(P'_i)_{i\in\mathscr{D}(\mathsf{c}_r)}$ induced by h_r and σ'. We have $P'_{i_r}=P^*_{i_r}$ for all $i\in[k-1]$, and

$$t_{k_r}(\sigma'_{k_r},\sigma^*_{-k_r}|\mathsf{h}_r)=r+t_{k_r}^d(\boldsymbol{p}')=r+t_{k_r}^d(P^*_{1_r},\ldots,P^*_{(k-1)_r},\boldsymbol{p}'_{-\{1_r,\ldots,(k-1)_r\}}).$$

It follows from Lemma EC.2 (the inequality in (EC.2)) that

$$t_{k_r}(\sigma'_{k_r},\sigma^*_{-k_r}|\mathsf{h}_r)\geq r+\min\left\{t_{k_r}^d(\bar{P}_{1_r}^{\mathsf{c}_r},\ldots,\bar{P}_{(k-1)_r}^{\mathsf{c}_r},R_{k_r})\,|\,R_{k_r}\in\bar{\mathscr{P}}_{k_r}\right\}=t_{k_r}(\sigma^*|\mathsf{h}_r).$$

The arbitrary choices of k and σ'_{k_r} imply that σ^* is an SPE of game Γ^{A}. Q.E.D.

[1] That is, i queues first at the last edge of $\bar{P}_{i_r}^{\mathsf{c}_r}$.

EC.10. Realization of NEs from SPEs

In this section, we establish by construction that with the same input, each NE outcome of game Γ^{N} is a certain SPE outcome of game Γ^{A}.

Recall from Section 5.2 that each configuration c_r of the extensive-form game Γ^{A} corresponds to a normal-form game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$ on network $G=(V,E)$ with agent set $\mathscr{D}(\mathsf{c}_r)$, i.e., the intermediary game of Γ^{N} starting from c_r at time r. For every agent $i \in \mathscr{D}(\mathsf{c}_r)$, let $\mathscr{P}_i^{\mathsf{c}_r}$ denote his strategy set in $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$, i.e., the set of paths in G along which i could travel (during a time period no earlier than r) given his position specified by c_r and $\cup_{s\geq r+1, v\in V}\Delta_{s,v}$.

Given any path profile $\boldsymbol{q}=(Q_i)_{i\in\mathscr{D}(\mathsf{c}_r)}$ of game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$, agent $j\in\mathscr{D}(\mathsf{c}_r)$ and vertex $v\in Q_j$, we use $t_j^v(\boldsymbol{q})_{\mathsf{c}_r}$ to denote the time when j reaches v under $\boldsymbol{q}$.

EC.10.1. Outline

Given any NE profile $\boldsymbol{p}$ of game Γ^{N}, we construct, for every history $\mathsf{h}_r=(\mathsf{c}_0,\ldots,\mathsf{c}_r)$ of game Γ^{A}, an NE $\boldsymbol{p}(\mathsf{h}_r)$ of game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$, i.e., the intermediary game of Γ^{N} starting from c_r with agent set $\mathscr{D}(\mathsf{c}_r)$. In particular, we set $\boldsymbol{p}(h_0):=\boldsymbol{p}$. Then, we construct an SPE of Γ^{A} by assembling these NEs such that starting from any history h_r $(r\geq 0)$ the outcome of the SPE is exactly the NE $\boldsymbol{p}(\mathsf{h}_r)$. Note that the reference of each NE constructed is a history instead of a configuration. Since different histories may have the same ending configuration c_r, we may construct multiple NEs for the same intermediary game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$.

Such an NE-based assembling is more complicated than the one discussed in Sections 5.2 and EC.9, which aims at producing nothing more than an SPE. What is more complicated here is that we are unable to design a Markovian SPE. In particular, the natural idea of constructing the NEs $\boldsymbol{p}(\mathsf{h}_r)$, $r\geq 1$, directly using Algorithm 2 does not work anymore. For example, an agent outside the first batch under $\boldsymbol{p}$ may have an incentive to deviate at the game tree root of Γ^{A} to another child node for which the special IDNE computed by Algorithm 2 chooses different routes (with unchanged arrival times) for agents in earlier batches, which creates room for the agent to minimize his own arrival time.

EC.10.2. Inductive construction of history-based NEs

Our (inductive) construction of the NEs $\boldsymbol{p}(\mathsf{h}_r)$ is done iteratively on the game tree of Γ^{A} starting from the root $\mathsf{h}_0=(\mathsf{c}_0)$. Initially, the constructed NE $\boldsymbol{p}(\mathsf{h}_0)$ for h_0 is simply the given NE $\boldsymbol{p}$. For each $r\geq 1$, suppose inductively that for a history $\mathsf{h}_{r-1}=(\mathsf{c}_0,\ldots,\mathsf{c}_{r-1})\in\mathsf{H}_{r-1}$, the NE $\boldsymbol{p}(\mathsf{h}_{r-1})$ of game $\Gamma^{\mathrm{N}}(\mathsf{c}_{r-1})$, written for convenience as $\boldsymbol{\alpha}=(A_i)_{i\in\mathscr{D}(\mathsf{c}_{r-1})}$, has been constructed. We construct in two steps the NE $\boldsymbol{p}(\mathsf{h}_r)$, denoted $\boldsymbol{\beta}=(B_i)_{i\in\mathscr{D}(\mathsf{c}_r)}$, for each child history $\mathsf{h}_r=(\mathsf{c}_0,\ldots,\mathsf{c}_{r-1},\mathsf{c}_r)$ of h_{r-1}. In the first step, we identify a subset U of $\mathscr{D}(\mathsf{c}_r)$ and let B_i, for each $i\in U$, be the subpath of A_i that i has not visited until time r under $\boldsymbol{\alpha}$. In the second step, based on $\boldsymbol{\beta}_U$ determined, we find an iteratively dominant partial path profile $\boldsymbol{\beta}_{\mathscr{D}(\mathsf{c}_r)\setminus U}$ for the remaining agents, who can by no means affect the agents in U provided the latter follow $\boldsymbol{\beta}_U$.

The first step. Let $(a_i)_{i\in\Delta(\mathsf{c}_{r-1})}$ be the action profile at game tree node h_{r-1} determined by $\boldsymbol{\alpha}$, i.e, no action in the profile deviates from $\boldsymbol{\alpha}$. More specifically,

- a_i is the first edge of A_i if under $\boldsymbol{\alpha}$ agent i does not move during time period $[r-1,r]$;
- a_i is the second edge of A_i if under $\boldsymbol{\alpha}$ agent i queues at the second edge of A_i at time r;
- a_i is the null action ϕ if under $\boldsymbol{\alpha}$ agent i exits G at time r.

Let $(b_i)_{i\in\Delta(\mathsf{c}_{r-1})}$ be the action profile that leads history h_{r-1} to its child history h_r (or equivalently leads c_{r-1} to c_r).

Recall that $\Delta(\boldsymbol{\alpha},[k])$ denotes the first k batches of agents reaching d under routing $\boldsymbol{\alpha}$, where $k\ge 0$. Define $\Bbbk\ge 0$ to be the maximum nonnegative integer k such that the action of each agent of $\Delta(\boldsymbol{\alpha},[k])\cap\Delta(\mathsf{c}_{r-1})$ under $(a_i)_{i\in\Delta(\mathsf{c}_{r-1})}$ is the same as that under $(b_i)_{i\in\Delta(\mathsf{c}_{r-1})}$, i.e., we set

$$\Bbbk := \sup\{k \mid a_i = b_i \text{ for all } i\in\Delta(\boldsymbol{\alpha},[k])\cap\Delta(\mathsf{c}_{r-1})\}. \tag{EC.11}$$

It is possible that $\Bbbk=0$ with $\Delta(\boldsymbol{\alpha},[0])=\emptyset$ or $\Bbbk=\infty$ with $\Delta(\boldsymbol{\alpha},[\infty])=\mathscr{D}(\mathsf{c}_{r-1})$. Define

$$\Omega := \Delta(\boldsymbol{\alpha},[\Bbbk]) \text{ and } U := \Omega\cap\mathscr{D}(\mathsf{c}_r). \tag{EC.12}$$

The set U consists of agents who under $\boldsymbol{\alpha}$ are in the first $\Bbbk$ batches and will not exit G from d by time r.

In the following construction of B_i for each $i\in U$, we let i "keep" his path under $\boldsymbol{\alpha}$, which yields an invariance of arrival times as specified below in Lemma EC.9. For each agent $i\in\Delta(\mathsf{c}_{r-1}) = (\cup_{e\in E}\mathbb{Q}_e^r)\cup(\cup_{v\in V}\Delta_{r+1,v})$, let e_i denote the first edge of A_i.

Construction I: (Construction of $\boldsymbol{\beta}_U$ with Invariant Arrival Times)

For each agent $i\in U$, set

$$B_i := \begin{cases} A_i\backslash\{e_i\}, & \text{if } a_i=b_i \text{ is the second edge of } A_i \text{ (which implies } i\in\cup_{e\in E}\mathbb{Q}_e^r\subseteq\Delta(\mathsf{c}_{r-1}));\\ A_i, & \text{otherwise.}\end{cases}$$

(NB: The if-condition in the above construction is equivalent to stating that when configuration c_{r-1} changes to configuration c_r, from time $r-1$ to time r, agent $i\in U$ travels along the edge e_i in G whose tail vertex is not the destination d, i.e., at time r agent i queues at the second edge of A_i. When the condition is satisfied, we set B_i to be $A_i\backslash\{e_i\}$, which is the path obtained from A_i by deleting its starting vertex and first edge e_i.)

The NE paths $B_i \in \{A_i, A_i\backslash\{e_i\}\}$ kept for agents i in $U = \Delta(\boldsymbol{\alpha}, [\mathbb{k}]) \cap \mathscr{D}(\mathsf{c}_r)$ particularly guarantee invariant arrival times at any vertex for these agents regardless of other agents' choices. To be specific, with the hierarchical independence of $\boldsymbol{\alpha}$ (as an NE of game $\Gamma^{\mathrm{N}}(\mathsf{c}_{r-1})$) stated in Section 4.3 and Theorem EC.6(i), we see that, as long as the chosen paths of agents in $\Omega = \Delta(\boldsymbol{\alpha}, [\mathbb{k}])$ remain as in $\boldsymbol{\alpha}_\Omega$, no matter what paths the agents in $\mathscr{D}(\mathsf{c}_{r-1})\backslash\Omega$ choose, the latter agents have no impact on the arrival times of the former agents at any vertex. This along with Construction I above implies the following lemma, which is the base of our construction of $\boldsymbol{\beta}_{\mathscr{D}(\mathsf{c}_r)\backslash U}$ in the second step.

For notational convenience, for all $i \in \mathscr{D}(\mathsf{c}_{r-1})\backslash\Delta(\mathsf{c}_{r-1})$ (i.e., agents who enter G at times later than r), we set e_i to be the null element ϕ.

LEMMA EC.9 **(Invariant Arrival Times)**. *For any agent $i \in U$, any vertex $v \in B_i \subseteq A_i$, and any partial path profile $\boldsymbol{q}_{\mathscr{D}(\mathsf{c}_r)\backslash U} = (Q_j)_{j\in\mathscr{D}(\mathsf{c}_r)\backslash U}$ in game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$, where $Q_j \in \mathscr{P}_j^{\mathsf{c}_r}$ for every $j \in \mathscr{D}(\mathsf{c}_r)\setminus U$, it holds that*

$$t_i^v(\boldsymbol{\alpha})_{\mathsf{c}_{r-1}} = t_i^v(\boldsymbol{\alpha}_\Omega, (\{e_j\} \cup Q_j)_{j\in\mathscr{D}(\mathsf{c}_r)\backslash U})_{\mathsf{c}_{r-1}} = t_i^v((B_j)_{j\in U}, \boldsymbol{q}_{\mathscr{D}(\mathsf{c}_r)\backslash U})_{\mathsf{c}_r}.$$

Before proving the lemma, we make some observations. For any agent $j \in \mathscr{D}(\mathsf{c}_r)$ and any path $Q_j \in \mathscr{P}_j^{\mathsf{c}_r}$, it is clear that $\{e_j\} \cup Q_j \in \mathscr{P}_j^{\mathsf{c}_{r-1}}$. Observe that either $\mathscr{D}(\mathsf{c}_{r-1}) = \mathscr{D}(\mathsf{c}_r)$, or $\mathscr{D}(\mathsf{c}_{r-1})\backslash \mathscr{D}(\mathsf{c}_r) \neq \emptyset$ and each agent in $\mathscr{D}(\mathsf{c}_{r-1})\backslash\mathscr{D}(\mathsf{c}_r)$ exits G at time r, giving $\mathscr{D}(\mathsf{c}_{r-1})\backslash\mathscr{D}(\mathsf{c}_r) = \Delta(\boldsymbol{\alpha}, 1) \subseteq \Delta(\boldsymbol{\alpha}, [\mathbb{k}])$. In any case we have

$$\mathscr{D}(\mathsf{c}_{r-1})\backslash\mathscr{D}(\mathsf{c}_r) \subseteq \Omega = \Delta(\boldsymbol{\alpha}, [\mathbb{k}]) \text{ and } \mathscr{D}(\mathsf{c}_r)\backslash U = \mathscr{D}(\mathsf{c}_r)\backslash\Omega = \mathscr{D}(\mathsf{c}_{r-1})\backslash\Omega.$$

Therefore, $(\boldsymbol{\alpha}_\Omega, (\{e_j\} \cup Q_j)_{j\in\mathscr{D}(\mathsf{c}_r)\backslash U})$ in Lemma EC.9 is simply $(\boldsymbol{\alpha}_\Omega, (\{e_j\} \cup Q_j)_{j\in\mathscr{D}(\mathsf{c}_{r-1})\backslash\Omega})$, a strategy profile of game $\Gamma^{\mathrm{N}}(\mathsf{c}_{r-1})$, in which the agents, including i, of the first $\mathbb{k}$ batches (defined w.r.t. $\boldsymbol{\alpha}$) follow their paths as in $\boldsymbol{\alpha}$.

Proof of Lemma EC.9. The first equality of the conclusion follows from the hierarchical independence in Theorem EC.6(i). The second equality is straightforward from Construction I and the fact that each agent in $\Omega\backslash U = \Omega\backslash\mathscr{D}(\mathsf{c}_r) \subseteq \mathscr{D}(\mathsf{c}_{r-1})\backslash\mathscr{D}(\mathsf{c}_r)$ (if any) exits G at time r, and he only has the null action under c_{r-1}, which has no effect on other agents. Q.E.D.

The second step. Based on the partial path profile $\boldsymbol{\beta}_U$ constructed (i.e., inherited from α_U) in the first step, we call Algorithm 3 to find an iteratively dominant path profile $(B_i)_{i\in\mathscr{D}(\mathsf{c}_r)\backslash U}$ for the remaining agents.

Recalling Lemma EC.1, let $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$ be the game on $\bar{G}$ whose restriction to G is the game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$. The partial path profile $(B_i)_{i\in U}$ constructed in Construction I naturally extends to a partial path profile $(\bar{B}_i)_{i\in U}$ of $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$, where the restriction of each $\bar{B}_i$ to G is B_i.

Construction II: (CONSTRUCTION OF ITERATIVELY DOMINANT $\boldsymbol{\beta}_{\mathscr{D}(\mathsf{c}_r)\backslash U}$)

1. Run Algorithm 3 with input $\bar{\Gamma}^{\mathrm{N}}(\mathsf{c}_r)$ and $\boldsymbol{b}=(\bar{B}_i)_{i\in U}$, which outputs $(\bar{P}_i)_{i\in\mathscr{D}(\mathsf{c}_r)\setminus\Omega}$.
2. For each agent $i\in\mathscr{D}(\mathsf{c}_r)\setminus U$, set B_i to be the restriction of $\bar{P}_i$ to G.

For easy expression of the null actions B_j of agents in $j\in\mathscr{D}(\mathsf{c}_{r-1})\setminus\mathscr{D}(\mathsf{c}_r)=\Delta(\mathsf{c}_{r-1})\setminus\Delta(\mathsf{c}_r)$, we reserve symbol $\boldsymbol{\phi}$ for the profile $(B_j)_{j\in\mathscr{D}(\mathsf{c}_{r-1})\setminus\mathscr{D}(\mathsf{c}_r)}$ of the null actions.

LEMMA EC.10. *Profile $\boldsymbol{\beta}$ is an NE of game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$.*

Proof. We need to prove that $t_i^d(\boldsymbol{\beta})_{\mathsf{c}_r}\le t_i^d(B_i',\boldsymbol{\beta}_{\mathscr{D}(\mathsf{c}_r)\setminus\{i\}})_{\mathsf{c}_r}$ for every agent $i\in\mathscr{D}(\mathsf{c}_r)$ and every path $B_i'\in\mathscr{P}_i^{\mathsf{c}_r}$.

Case 1: $i\in U\subseteq\Omega$. Suppose $i\in\Delta(\boldsymbol{\alpha},k)$ for some $k\le\Bbbk$. Then for any path profile $\boldsymbol{q}=(Q_j)_{j\in\mathscr{D}(\mathsf{c}_r)}$ of $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$, with $\mho:=\Delta(\boldsymbol{\alpha},[k-1])\subset\Omega$, we have

$$t_i^d(\boldsymbol{\beta})_{\mathsf{c}_r}=t_i^d(\boldsymbol{\alpha})_{\mathsf{c}_{r-1}}\le t_i^d(\boldsymbol{\alpha}_{\mho},(\{e_j\}\cup Q_j)_{j\in\mathscr{D}(\mathsf{c}_r)\setminus\mho},\boldsymbol{\phi}_{\mathscr{D}(\mathsf{c}_{r-1})\setminus\mathscr{D}(\mathsf{c}_r)\setminus\mho})_{\mathsf{c}_{r-1}}=t_i^d(\boldsymbol{\beta}_{\mathscr{D}(\mathsf{c}_r)\cap\mho},\boldsymbol{q}_{\mathscr{D}(\mathsf{c}_r)\setminus\mho})_{\mathsf{c}_r},$$

where the first equality is by Lemma EC.9, the inequality is from hierarchical optimality in Theorem EC.6(ii), and the last equality is due to Construction I. In particular, when taking $Q_i=B_i'$ (noting $i\notin\mho$) and $Q_j=B_j$ for every $j\in\mathscr{D}(\mathsf{c}_r)\setminus\mho\setminus\{i\}$, we obtain $t_i^d(\boldsymbol{\beta})_{\mathsf{c}_r}\le t_i^d(B_i',\boldsymbol{\beta}_{\mathscr{D}(\mathsf{c}_r)\setminus\{i\}})_{\mathsf{c}_r}$ as desired.

Case 2: $i\in\mathscr{D}(\mathsf{c}_r)\setminus U=\mathscr{D}(\mathsf{c}_r)\setminus\Omega$. By Construction II, we deduce from Lemma EC.3 that the path B_i is i's best response to other agents' choices, giving $t_i^d(\boldsymbol{\beta})_{\mathsf{c}_r}\le t_i^d(B_i',\boldsymbol{\beta}_{\mathscr{D}(\mathsf{c}_r)\setminus\{i\}})_{\mathsf{c}_r}$. Q.E.D.

With Lemma EC.10, we complete our inductive constructions of history-based NEs $\boldsymbol{p}(\mathsf{h}_r)$ for all histories h_r of game Γ^{A}.

EC.10.3. Assembling an SPE from NEs

The partial hierarchical independence and iterative dominance guaranteed by Constructions I and II enable us to accomplish our task of assembling all the NEs $\boldsymbol{p}(\mathsf{h}_r)$, $\mathsf{h}_r\in\mathsf{H}_r$, $r\ge 0$, constructed in Section EC.10.2 into an SPE of Γ^{A}.

Let $\sigma=(\sigma_i)_{i\in\Delta}$ be a strategy profile of Γ^{A} defined as follows: at each history $\mathsf{h}_r=(\mathsf{c}_0,\ldots,\mathsf{c}_r)$, agents in $\mathscr{D}(\mathsf{c}_r)$ take actions as specified by the NE $\boldsymbol{p}(\mathsf{h}_r)$ constructed in Section EC.10.2 for h_r, where $\boldsymbol{p}(\mathsf{c}_0)$ is the given NE $\boldsymbol{p}$ of game Γ^{N}.

THEOREM EC.7. *The strategy profile σ is an SPE of game Γ^{A} such that the path profile induced by the initial history h_0 and σ is exactly $\boldsymbol{p}$.*

Proof. Similar to the proof of Theorem 5 (see Section EC.9), it can be deduced from Constructions I and II (and Lemma EC.3) that, for each history h_r, the path profile induced by h_r and σ is exactly $\boldsymbol{p}(\mathsf{h}_r)$.

To see that σ is an SPE of Γ^{A}, we fix an arbitrary $r \geq 0$ and an arbitrary history $\mathsf{h}_r = (\mathsf{c}_0, \ldots, \mathsf{c}_r) \in \mathsf{H}_r$. Let $\boldsymbol{\beta} = (B_i)_{i \in \mathscr{D}(\mathsf{c}_r)}$ denote the NE $\boldsymbol{p}(\mathsf{h}_r)$ of $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$ we have constructed for h_r. In the case of $r = 0$, we set $\boldsymbol{\beta} := \boldsymbol{p}$. Moreover, we consider any $i \in \mathscr{D}(\mathsf{c}_r)$, any $\sigma'_i \in \Sigma_i$, and the path profile $\boldsymbol{q} = (Q_j)_{j \in \mathscr{D}(\mathsf{c}_r)}$ induced by h_r and $\sigma' := (\sigma'_i, \sigma_{-i})$. We need to verify that $t_i(\sigma|\mathsf{h}_r) \leq t_i(\sigma'|\mathsf{h}_r)$.

If $r = 0$, then we suppose that $i \in \Delta(\boldsymbol{p}, k)$ and write $\mho = \Delta(\boldsymbol{p}, [k-1])$. By the hierarchical independence of $\boldsymbol{p}$ (Theorem EC.6(i)), no action change of agent i can alter the batch index of any agent in $\mho$. Therefore, using an inductive argument, we deduce from Construction I that at each history node $\mathsf{h}_s = (\mathsf{c}_0, \ldots, \mathsf{c}_s)$ on the path (in the game tree of Γ^{A}) induced by σ', all agents of $\mathscr{D}(\mathsf{c}_s) \cap \mho$ belong to the set Ω defined w.r.t. $\boldsymbol{p}(\mathsf{h}_s)$ (cf. (EC.12), where Ω is defined w.r.t. $\boldsymbol{\alpha}$). It follows that $Q_j = P_j = B_j$ for all $j \in \mho$. In turn, $\boldsymbol{p}$'s hierarchical optimality (Theorem EC.6(ii)) states that $t_i(\sigma|\mathsf{h}_0) = t_i^d(\boldsymbol{p}) \leq t_i^d(\boldsymbol{p}_{\mho}, \boldsymbol{q}_{\Delta \setminus \mho})_{\mathsf{c}_0} = t_i^d(\boldsymbol{q})_{\mathsf{c}_0} = t_i(\sigma'|\mathsf{h}_0)$.

So we assume now $r \geq 1$. Then h_r is a child history of some (unique) history $\mathsf{h}_{r-1} = (\mathsf{c}_0, \ldots, \mathsf{c}_{r-1}) \in \mathsf{H}_{r-1}$. Let $\boldsymbol{\alpha}$ denote the NE $\boldsymbol{p}(\mathsf{h}_{r-1})$ of $\Gamma^{\mathrm{N}}(\mathsf{c}_{r-1})$, and let $\Bbbk$ and $\Omega = \Delta(\boldsymbol{\alpha}, [\Bbbk])$ be defined as in (EC.11) and (EC.12).

If $i \in \Delta(\boldsymbol{\alpha}, k) \subseteq \Omega$ for some $k \leq \Bbbk$, then Construction I implies that $Q_j = B_j$ for all $j \in \mathscr{D}(\mathsf{c}_r) \cap \mho$, where $\mho := \Delta(\boldsymbol{\alpha}, [k-1])$. As in Case 1 of the proof of Lemma EC.10, we deduce that $t_i(\sigma|\mathsf{h}_r) = t_i^d(\boldsymbol{\beta})_{\mathsf{c}_r} \leq t_i^d(\boldsymbol{\beta}_{\mathscr{D}(\mathsf{c}_r) \cap \mho}, \boldsymbol{q}_{\mathscr{D}(\mathsf{c}_r) \setminus \mho})_{\mathsf{c}_r} = t_i^d(\boldsymbol{q})_{\mathsf{c}_r} = t_i(\sigma'|\mathsf{h}_r)$.

It remains to consider the case of $i \in \mathscr{D}(\mathsf{c}_r) \setminus \Omega = \mathscr{D}(\mathsf{c}_r) \setminus U$. Assume w.l.o.g. that i is exactly the ith agent in the ordering $1, 2, \ldots$ of agents in $\mathscr{D}(\mathsf{c}_r) \setminus \Omega$ associated with the iteratively dominant path profile constructed in Construction II. Again Construction I guarantees $\boldsymbol{q}_U = \boldsymbol{\beta}_U$. It follows from Lemma EC.3 (i.e., the iterative dominance) that $\boldsymbol{q}_{[i-1]} = \boldsymbol{\beta}_{[i-1]}$, and $t_i(\sigma|\mathsf{h}_r) = t_i^d(\boldsymbol{\beta})_{\mathsf{c}_r} \leq t_i^d(\boldsymbol{\beta}_U, \boldsymbol{\beta}_{[i-1]}, \boldsymbol{q}_{\mathscr{D}(\mathsf{c}_r) \setminus U \setminus [i-1]})_{\mathsf{c}_r} = t_i^d(\boldsymbol{q})_{\mathsf{c}_r} = t_i(\sigma'|\mathsf{h}_r)$, which completes the proof. Q.E.D.

EC.11. NE existence: edge priorities vs. agent priorities

We have proved that our game Γ^{N} admits an NE, where edge priorities play a crucial role. In contrast, as Example 3 shows, an NE may not exist in the multi-origin case under the model of Scarsini et al. (2018), where priorities are placed on agents. On the other hand, in the case of single origin, their model does guarantee the NE existence. In this section, we explain why the NE existence result on single-origin networks extends to the multi-origin case in our model, but not in the model of Scarsini et al. (2018).

The critical reason lies in whether we are able to order all agents in some way such that former agents in this order have absolute advantages over latter ones, using their heterogeneities, such as initial priorities, entering times, and different origins, etc. This is possible in the single-origin case of Scarsini et al. (2018), because a proper combination of the agents' entry times into the network and their initial priorities works. In this combination, entry times play a dominant role over initial priorities and hence the two factors are actually combined in a lexicographical way. To be more specific, since there is a single origin, agents entering the network earlier are always ordered before later ones; for agents entering the network at the same time, priorities associated with them can be used to break ties. Along with the local FIFO principle, we have seen that this ordering, a lexicographical combination of entry times and initial priorities, is decisive in that as long as an agent has some advantage over another at the origin, he will have advantages at all subsequent vertices. This idea is the essence of almost all related NE existence results in atomic dynamic routing games.

As Example 3 demonstrates, the above idea does not extend to the multi-origin case for the model of Scarsini et al. (2018), because Rock-Paper-Scissor relationships may occur. When agents enter the network from different origins, the same two factors, entry times and agent priorities, are still important. But they cannot be reconciled so well as in the single-origin case. The power of entry times is significantly weakened: when two agents come into the network from different origins, their entry times might not so important, while the locations of their entry points matter. However, the original locations and agent priorities cannot work together in a lexicographical way to determine a decisive ordering: sometimes original locations are more powerful and some other times agent priorities are more powerful, and this may lead to cyclic phenomenon as demonstrated in Example 3, making the existence of an NE impossible. To be more specific, we have shown in Example 3 that the first prioritized agent g may be blocked by the last prioritized agent i in every possible path for him (due to i's original location advantage); the last prioritized agent i may be blocked by the second prioritized agent h (due to h's priority over i), and the second prioritized agent h may be blocked by the first prioritized agent g (due to g's priority over h). The three agents form a Rock-Paper-Scissor cycle, destroying the existence of NE.

One advantage of our model is that we introduce edge priorities, which may be viewed as a tool of space, to help us untangle the complicated relationships among all agents. (Note that this kind of space information is ignored in the model of Scarsini et al. (2018).) We have seen that the combination of time and space plays a decisive role in the routing from a new perspective: as long as an agent is able to reach the destination earlier than another, he is able to do so for any intermediary vertex. To be more specific, the location of an agent's origin and fixed edge priorities of the network under our model can induce a space advantage for the agent, while the entry time of an agent can be viewed as his time advantage. The agents can be linearly ordered according to a kind of "combination" of their space and time advantages so that an agent with a higher order can find a path from his origin to the destination such that he dominates all agents with lower orders all the way along the path. Intuitively, the agent priorities (though consistent with the time advantages) in the model of Scarsini et al. (2018) may not reconcile with the space advantages, while the edge priories in our model, which define parts of space advantages, make possible the reconciliation with time and space advantages.

EC.12. Supplementary examples

In this section, we present several supplementary examples under our game model Γ^{N}, which demonstrate a Braess-like paradox (involving route changes due to routing environment improvement or deterioration), absence of the earliest arrival, and presence of overtaking. (Recall from Section 4.1 that IDNEs are earliest arrival and no overtaking.)

A paradox involving route changes. We illustrate the counter-intuitive phenomenon that the *route changes* resulting from removing initial queues (or removing agents or shortening path lengths) in a series-parallel network may slow the system performance. This kind of paradox was discovered by Scarsini et al. (2018) under their model. Example 3 presented in Scarsini et al. (2018) is an extension-parallel network adjusted from the one in Macko et al. (2013) for showing a classical Braess's paradox in nonatomic dynamic flow games. Our example below is a direct adaptation of the example in Scarsini et al. (2018).

EXAMPLE EC.1. Consider a game instance Γ^{N} on the series-parallel network illustrated in Figure EC.2, where o is the single origin, d is the single destination, e_1 has a higher priority than e_2, and at e_3 there is an initial queue of three agents. At each time point $r \geq 1$, three agents of $\Delta_{r,o} = \{1_r, 2_r, 3_r\}$ enter the network from origin o. Regarding the original ranks, 1_r's rank is higher than 2_r's, and 2_r's is higher than 3_r's. The agents in $\cup_{r\geq 1}\Delta_{r,o}$ may choose one of the five o-d paths $R_1 := ou_1u_2d$, $R_2 := ou_1u_2u_3d$, $R_3 := ovu_2d$, $R_4 := ovu_2u_3d$ and $R_5 := ow_1w_2w_3d$.

(E1) It is easy to verify that, with the presence of the initial queue at e_3, every NE of the game Γ^{N} incurs a travel cost 4 to each agent outside the initial queue. For example, that agents $1_r, 2_r, 3_r$ (for all $r \geq 1$) follow R_1, R_4, R_5 respectively gives an NE.

(E2) Removal of the initial queue (i.e., the three agents) at e_3 may lead the system to a less efficient NE. While agent 1_1 still follows R_1, which incurs him the smallest travel cost 3, agent 2_1 (resp. 3_1) may *change* his route to R_1 (resp. R_4) along which he pays the smallest possible travel cost 4 (given the choice of 1_1). Building on the best choices R_1, R_1, R_3 of agents $1_1, 2_1, 3_1$, it is routine to verify that for every $r = 2, 3, \ldots$, the sequential route changes of agents $1_r, 2_r, 3_r$ to paths R_3, R_5, R_2 incur them sequentially smallest possible costs 4, 4, 5. These paths indeed form an NE.

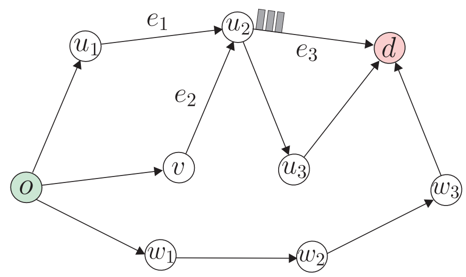


Figure EC.2 Removal of an initial queue may slow down the system performance

It is worth noting that the role of the initial queue in the above example can be played by some agents who enter the network earlier or by decreasing the length of a certain u_2-d path.

An NE that is not earliest arrival. The NE specified in (E2) of Example EC.1 is not earliest arrival, since, given other agents' choices, the earliest time agent 2_1 could reach vertex u_2 is 3, one time unit earlier than his arrival time at u_2 under the NE.

An NE that is temporally overtaking. The following example shows that an NE of game Γ^{N} is not necessarily no-overtaking.

Example EC.2. Consider a game Γ^{N} on the single-origin single-destination network in Figure EC.3, where at edge wx (resp. wy) there is an initial queue of three agents. In addition to the six agents, there are two agents, 1 and 2, entering the network from origin o at times 1 and 2, respectively. If agents 1 and 2 go through paths $ouvwxd$ and $owyd$, respectively, then they both reach destination d at the earliest possible time 6, yielding an NE of the game. Under this NE, agent 2 overtakes agent 1 at vertex w.

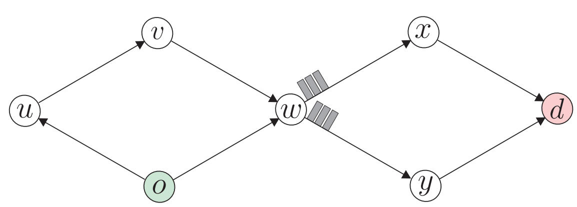


Figure EC.3 A temporal overtaking NE

EC.13. The hybrid game model

In this section, we consider "hybrid" agents, whose behaviors lie between adaptive and nonadaptive. An *agent* used without specification is meant a hybrid agent in this section. The corresponding game model is referred to as hybrid.

EC.13.1. Model description

For every agent i and every vertex v that is neither i's origin nor the destination d, we are given a probability $\theta_{i,v}$ that agent i contemplates switching to other paths at v. Let $\boldsymbol{\theta}$ denote the vector of these probabilities. We use $\Gamma^\sharp(\boldsymbol{\theta})$ to denote the hybrid game with parameter vector $\boldsymbol{\theta}$.

While adaptive agents make routing decisions at every nonterminal vertex they reach as to which *edge* to take next, hybrid agents make decisions at every nonterminal vertex as to which *path* to take in the future if they are given the chances (by Nature) to reconsider their plans, and just follow their previous plans otherwise. Intuitively, each agent always holds a plan (a path from his current edge to the destination) and may update it with a new one when chances are given. A precise definition of a strategy is presented as follows.

Definition EC.3 (Strategy). A *strategy* of agent $i \in \Delta$ is a mapping $\sigma_i^\sharp$ that maps each history $\mathsf{h}_r = (\mathsf{c}_0, \ldots, \mathsf{c}_r)$ till time r with $i \in \Delta(\mathsf{c}_r)$ to $\sigma_i^\sharp(\mathsf{h}_r)$ such that, based on c_r and the edge-priority DQ rule, either $\sigma_i^\sharp(\mathsf{h}_r)$ is a path from the current edge where i stays to the destination d, or $\sigma_i^\sharp(\mathsf{h}_r)$ is a null element when under c_r agent i will exit G at time $r+1$.

The *strategy set* of agent i is denoted as $\Sigma_i^\sharp$. A vector $\sigma^\sharp = (\sigma_i^\sharp)_{i\in\Delta}$ is called a *strategy profile* of the hybrid game $\Gamma^\sharp(\boldsymbol{\theta})$. Note that this game is typically a stochastic model. We use $\mathbb{E}[t_i(\sigma^\sharp|\mathsf{h}_r)]$ to denote the expected arrival time of agent i at the destination under strategy profile $\sigma^\sharp$ starting from history h_r.

Definition EC.4 (SPE in the hybrid game). A strategy profile $\sigma^\sharp = (\sigma_i)_{i\in\Delta}$ is a *subgame perfect equilibrium* (SPE) of $\Gamma^\sharp(\boldsymbol{\theta})$ if for any time $r \geq 0$ and any history $\mathsf{h}_r \in \mathsf{H}_r$, $\mathbb{E}[t_i(\sigma^\sharp|\mathsf{h}_r)] \leq \mathbb{E}[t_i(\sigma_i^{\sharp\prime}, \sigma_{-i}^\sharp|\mathsf{h}_r)]$ holds for all $i \in \Delta(\mathsf{c}_r)$ and all $\sigma_i^{\sharp\prime} \in \Sigma_i^\sharp$ such that $(\sigma_i^{\sharp\prime}, \sigma_{-i}^\sharp)$ still leads to history h_r, where $\sigma_{-i}^\sharp$ is the partial strategy profile of $\sigma^\sharp$ for agents in $\Delta\backslash\{i\}$.

EC.13.2. Results

As intuitively expected, we have the following observation.

Lemma EC.11. *For the hybrid model $\Gamma^\sharp(\boldsymbol{\theta})$, the case $\boldsymbol{\theta} = \mathbf{0}$ corresponds to the nonadaptive model Γ^{N} and the case $\boldsymbol{\theta} = \mathbf{1}$ corresponds to the adaptive model Γ^{A}.*

Proof. In fact, when $\theta = 0$, all the plans at the non-origin vertices will never be used and hence a strategy for a hybrid agent reduces to a strategy of a nonadaptive agent. On the other hand, when $\theta = 1$, all the plans at the non-origin vertices will always be given the chances to realize and hence only the immediate next edges are meaningful for the plans and the set of these immediate next edges is equivalent to a strategy of the adaptive agent. Q.E.D.

Suppose that we are given an SPE σ for game Γ^{A} that is constructed from an NE $\boldsymbol{p}$ of game Γ^{N}, as discussed in Sections 5.3 and EC.10. We construct a strategy profile $\sigma^{\sharp}$ for the hybrid model $\Gamma^{\sharp}(\boldsymbol{\theta})$ as follows. For each history $\mathsf{h}_r = (\mathsf{c}_0, \ldots, \mathsf{c}_r)$, if all players carry out their strategies in σ, then for each player i, a path from his current edge to the destination will be determined. We set $\sigma_i^{\sharp}(\mathsf{h}_r)$ as this path. This defines a strategy profile $\sigma^{\sharp}$ for the hybrid model $\Gamma^{\sharp}(\boldsymbol{\theta})$.

Theorem EC.8. *The strategy profile $\sigma^{\sharp}$ constructed above is an SPE for the hybrid game $\Gamma^{\sharp}(\boldsymbol{\theta})$.*

Proof. By definition, it suffices to prove that, for any time $r \geq 0$ and any history $\mathsf{h}_r \in \mathsf{H}_r$, $\mathbb{E}[t_i(\sigma^{\sharp}|\mathsf{h}_r)] \leq \mathbb{E}[t_i(\sigma_i^{\sharp\prime}, \sigma_{-i}^{\sharp}|\mathsf{h}_r)]$ holds for all $i \in \Delta(\mathsf{c}_r)$ and all $\sigma_i^{\sharp\prime} \in \Sigma_i^{\sharp}$ such that $(\sigma_i^{\sharp\prime}, \sigma_{-i}^{\sharp})$ still leads to history h_r, where $\sigma_{-i}^{\sharp}$ is the partial strategy profile of $\sigma^{\sharp}$ for agents in $\Delta \backslash \{i\}$.

Consider the subgame starting from history h_r. At the starting time r, all agents i at their initial positions in the subgame hold $\sigma_i^{\sharp}(\mathsf{h}_r)$ their initial plans. Then all agents i act during time $[r, r+1]$ according to $\sigma_i^{\sharp}(\mathsf{h}_r)$, which leads to a history h_{r+1}. By our construction presented in Section EC.10, agent i's new plan $\sigma_i^{\sharp}(\mathsf{h}_{r+1})$ at time $r+1$ is consistent with his old plan $\sigma_i^{\sharp}(\mathsf{h}_r)$ at time r, i.e., he does not switch his path even if he is given the chance to do so. Inductively, we see that the realized path profiles of the two strategy profiles $\sigma^{\sharp}$ (in game $\Gamma^{\sharp}(\boldsymbol{\theta})$) and σ (in game Γ^{A}) are the same. Therefore, the arrival time $t_i(\sigma^{\sharp}|\mathsf{h}_r)$ of i at the destination is also deterministic, and equals $t_i(\sigma|\mathsf{h}_r)$.

Suppose that i is in the kth batch in the routings determined by $\sigma^{\sharp}$ and h_r. Consider the single-deviation of agent i. By the construction of the SPE σ, all agents in the first $k-1$ batches will keep their plans unchanged in the histories following h_r. In other words, regardless of the chances given by Nature, all agents in the first $k-1$ batches will always follow their paths in the corresponding NE $\boldsymbol{p}(\mathsf{h}_r)$ of game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$ (see Section EC.10). Recalling the hierarchically optimality property for any NE of game $\Gamma^{\mathrm{N}}(\mathsf{c}_r)$, we see that $t_i(\sigma^{\sharp}|\mathsf{h}_r) = t_i(\sigma|\mathsf{h}_r)$, the exit time of agent i, is the smallest among all the exit times of all agents outside the first $k-1$ batches under any routing in which agents in the first $k-1$ batches follow their NE routes (see Sections 4.3 and EC.7). This proves that i cannot be better off by a unilateral deviation in game $\Gamma^{\sharp}(\boldsymbol{\theta})$ and hence the constructed $\sigma^{\sharp}$ is an SPE. Q.E.D.