



ON INTENTION-PROPAGATION-BASED PREDICTION IN AUTONOMOUSLY SELF-ADAPTING NAVIGATION *

LÁSZLÓ Z. VARGA[†]

Abstract. It is widely believed that road traffic as a whole self-adapts to the current situation to make travel times shorter by avoiding congestions, if the autonomously operating navigation devices exploit real-time traffic information. The classical theoretical models do not have definite answer if car navigation based on real-time data is able to self-adapt and produce better traffic, or not. The novel theoretical approach to study this belief is the online routing game model. Current commercial car navigation systems are modelled with the class of simple naive online routing games. It is already proved that simple naive online routing games may show undesirable phenomena. One of the approaches to improve car navigation is intention-propagation-based prediction, where agents share their intention, and can forecast future travel times. In this paper we prove that in spite of exploiting this type of prediction in online routing games, the phenomena studied in simple naive online routing games are still possible, although in a different way. With these theoretical results we point out hitherto hidden pivotal phenomena of intention-propagation-based prediction in collective adaptive systems.

Key words: online routing games, benefit of online data, prediction

AMS subject classifications. 68T42, 91A40, 93C40, 94A15

1. Introduction. Current road traffic navigation systems have access to real-time data, and they can be regarded as autonomously self-adapting navigation systems. They are *autonomous*, because everyday traffic is not coordinated by a centralized system, and the traffic participants make their own autonomous decisions based on their intentions and the information available for them locally. This means that instead of centralized decision making, we have a *collective* of autonomous distributed decision makers, i.e. a multi-agent system. In this aspect autonomy refers to the autonomous route planning by the navigation devices in the individual cars, instead of following the instructions of some centralized planner. Even if the traffic was coordinated by a centralized system, there would be the question whether the individual traffic participants would conform to its instructions. Current navigation systems are *self-adapting* as well, because the navigation devices in the cars can get up-to-date information on the current status of the traffic, like the current travel time on each road, indicating the current situation of the traffic that needs to be adapted to. The routing algorithms implemented in the navigation devices must be able to utilize this real-time data to self-heal the global traffic, for example if a road becomes congested then the navigation devices autonomously tell the individual cars how to adapt to the current traffic situation, and send the cars to less congested roads. Ultimately, road traffic guided by navigation devices that utilize real-time data is a *collective adaptive system*.

Although current navigation systems are already able to utilize real-time data for route planning, these systems were implemented without clearly understanding how real-time data impacts traffic as a whole, and how real-time data affects the above mentioned autonomous and self-adaptation aspects of navigation systems. Note that in this paper we focus on online-data-based self-adaptation. This is different from self-adaptation based on previous experiences, like, for example, in the case of route selection from home to work based on the experience of the previous day.

Autonomous agents of car navigation devices have access to online real-time data, and they create plans how to achieve their goals in an environment where they jointly utilize resources that become more costly as more agents use them. For example, the online data are the current travel times on roads, the goal is to get to a destination, the created plan is the travel route from the starting point to the destination, the jointly utilized resources are the roads, the cost of resource utilization is the travel time on the road, and the travel time on a road increases when more cars are on that road. Agents are dynamically arriving and departing. The plans are created by exploiting online data that describe the current status and the current cost of the resources. There

*Research supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013), and is part of TUD COST Action TU1102 Autonomic Road Transport Support Systems.

[†]ELTE Regional Knowledge Centre, Irányi Dániel u. 4., Székesfehérvár, H-8000 Hungary (<http://people.inf.elte.hu/lzvarga/>).

is uncertainty about the feasible decision of an agent, because the cost of the resources will change by the time the agent starts to use them: departing agents will release the resources, agents simultaneously creating their plans will influence each others' costs, and agents arriving later may also influence the costs of the resources used by agents already executing their plans.

We do not know much about the benefit of real time information, not to mention how to optimize route planning based on real-time data to make the traffic self-adapt to the current situation, and self-heal itself. The novel *online routing game* model [1] is a theoretical model to formally investigate if autonomously self-adapting car navigation produces better traffic, or not. There is need for such theoretical studies, because driverless autonomous cars are being designed, and online navigation devices based on the (possibly multiple objective [2]) shortest path search strategy are widely installed in many cars. Shortest path search is basically the same as the simple naive strategy that was discussed in [1], and it was proved that if car navigation devices use shortest path search, and exploit real-time data, then the traffic may show undesirable phenomena.

In order to improve the simple naive strategy, the approach of intention propagation was proposed in the anticipatory vehicle routing system using delegate multi-agent systems [3]. In this paper we take the online routing game model of [1], refine it, and then we extend this model to formally investigate and prove some of the properties of the usage of intention-propagation-based prediction in car navigation systems. These investigations are on the formal model level, so the results are valid for any online routing game using intention-propagation-based prediction, including real-time data based car navigation and traffic. The intention-propagation-based prediction can be implemented with or without bio-inspired design patterns [17] as long as they conform to the theoretical model. We do not deal with the implementation of intention-propagation-based prediction, therefore we are not investigating if its implementation can scale with the number of nodes, which may be hundreds or thousands in practice. We assume that intention-propagation-based prediction is somehow implemented, and we are investigating its benefits on the theoretical level.

In the next section we discuss related work and how we advance the state of the art, in Sect. 3 we shortly explain and refine the online routing game model, in Sect. 4 we summarise the intention-propagation-based prediction approach, in Sect. 5 we prove four properties of online routing games which exploit intention-propagation-based prediction, and finally in Sect. 6 we conclude our work.

2. Related Work. Several authors investigated with simulation tools how the traffic would self-adapt if the majority of vehicles used online traffic information for route planning, and they concluded that online data has to be used carefully. A simple scenario consisting of two parallel routes was investigated in [4], and the simulations showed that online information often leads to oscillations in the number of cars on the routes, the velocity, and travel times, which lead to worse overall performance. In the discussion, the authors conclude that one of the reasons for the oscillations is that the real-time information reflects the state of the network some time ago. Another reason for the oscillation is that the agents do not coordinate their actions. In order to improve these, the authors advise the usage of anticipatory traffic forecast based on the broadcast route choice of the agents, which basically means that the agents share or propagate their intentions.

Anticipation [14] is essential to multi-agent systems in order to be able to behave proactively, and adapt to changing situations. Several design patterns were studied [15] to help the systematic design of self-organising emergent systems that show anticipatory behaviour. Many of these design patterns are inspired by biological systems [16], which are well suited to design pervasive service ecosystems like crowd steering [18]. Among the bio-inspired design patterns [17], the digital pheromone pattern and the evaporation pattern are the two most relevant to implement intention propagation. The online routing game model of this paper abstracts away from these design patterns, and models the mechanisms of these design patterns with the ability of the agents of the self-organising system to spread their truthful intentions, and the ability of the infrastructure layer to compute the future status of the system as a result of the intended actions of the agents (see Sect. 4 and the beginning of Sect. 5).

An empirical intention propagation system using such design patterns was elaborated in the anticipatory vehicle routing simulation system using delegate multi-agent systems [3]. This simulated distributed delegate multi-agent system is able to forecast future traffic conditions based on the broadcast intention of the vehicle agents. This forecast information is then used by the vehicle agent to make routing decisions. The approach was evaluated in a simulated network of Leuven, using a statistical approach, and the authors concluded that the

use of forecast data results in shorter trip durations than the use of real-time data from Traffic Message Channel (TMC). The statistical approach took into account the trip durations from the anticipatory route planning and the mean values of a perceived distribution. The extreme values of trip durations were not included in the evaluation. The authors claim that the forecast data not only allow drivers to avoid existing congestions, but prevents them from forming congestion as well. The authors admit that providing efficient and stable routes to vehicles will be challenging, even with the use of intention propagation.

Simulations are good for initial evaluations of the approaches, but we can expect a complete evaluation only from theoretical models. The most relevant model is algorithmic game theory, which studies networks with source routing (Sect. 18 in [5]), where end users simultaneously choose full routes for their traffic, and the traffic is routed in a congestion sensitive manner. The algorithmic game theory investigations of the routing game problem revealed important basic properties, like the conditions for the existence of equilibrium, or the limit on the "price of anarchy". However the algorithmic game theory model contains assumptions which do not hold in traffic routing using real-time data. These assumptions are: a) the throughput characteristic of the network does not change with time, and the agents know the characteristic functions; b) the agents simultaneously decide their optimal route; and c) the outcome travel time for a given agent depends on the choice of all the agents and the characteristic of the network, but not on the trip schedule of the other agents.

The issue of traffic dynamism is studied in the field of dynamic traffic assignment [6], but there they investigate the time-varying properties of traffic flow, whereas here we assume that the traffic flow is constant, and only the cost functions may change. In our investigations the critical issue is the sequential decision making of the autonomous agents, which self-adapt to the observed real-time status of the world. This is partly handled by online mechanism design [9].

The novel online routing game model [1] has less restrictive assumptions than the routing game model, and the online routing game approach more realistically models traffic routing based on real-time data by combining the algorithmic game theory approach with the approaches of online algorithms [10] and online mechanism design. Online routing games are different from resource allocation or minority games [11], which are simultaneous one shot or repeated simultaneous games where there might be some coordination among some of the agents. In contrast, online routing games are continuous and non-cooperative games exploiting real-time data. Online routing games are classified [1] into different classes depending on the decision strategies used by the agents. Commercial car navigation software use shortest path search in the road network, possibly taking into account real time traffic data and modifying the static data with the online information about the actual traffic delays. This decision strategy is the simple naive strategy studied theoretically in [1]. It was proved that stability is not guaranteed in the class of simple naive online routing games, and it may happen in some networks that a single flow enters the network, and on some edge inside the network the flow is bigger than the one that entered the network. It was proved that the worst case benefit of online data in some networks may be bigger than one, which means that the worst case travel time with online data may be more than the travel time without online data.

The online routing game model allows the agents to exploit the services of an "advisory" like the intention-propagation-based forecast of the delegate multi-agent system in [3] or the central planner in the PLANETS system [12]. In this paper we advance the state of the art by formally investigating the usage of intention-propagation-based prediction in online routing games, and proving four properties of the class of simple naive intention propagation online routing games. Specifically we prove that in spite of using intention-propagation-based prediction, in some online routing games a single flow that entered the network may become bigger on some edge inside the network, the traffic may fluctuate, the worst case benefit of online data may be bigger than one, i.e. it may be a "price", and this price may be arbitrarily large. By these theoretical results, we pinpoint the front lines of future research on self-adapting autonomous traffic systems.

3. Formal Analysis with Online Routing Games. The online routing game model of [1] is a combination of the algorithmic game theory model of the routing problem and the online mechanisms. The online routing game model resembles the algorithmic game theory routing game model in the concepts of flow, cost, and resources, and it resembles the model of online mechanisms in the sequences of time periods and decisions. A T time unit is introduced, because the costs of the resources depend on the rate of their utilization. The online routing game model of [1] needs some additional refinements, which were not relevant in [1], but are

needed here, so we refine the model of [1] in the following. The online routing game model is the sextuple (t, T, G, c, r, k) , where

- $t = \{1, 2, \dots\}$ is a sequence of equal time periods.
- T is a natural number, and T time periods give one time unit (e.g. one minute).
- G is a directed multi-graph¹ $G = (V, E)$ with vertex set V and edge set E where each $e \in E$ is characterized by a cost function c_e and a time gap gap_e .
- c is the cost function of G with $c_e : R^+ \rightarrow R^+$ for each edge e of G , and it maps the flow² $f_e(\tau)$ (that enters the edge e at time τ) to the travel time on the edge. c_e for the agent entering the edge e is never less than the remaining cost of any other agent already utilizing that edge increased with a time gap³ gap_e , which is specific to edge e . The value of the flow $f_e(\tau)$ is the number of agents that entered the edge e between $\tau - T$ (inclusive) and τ (non-inclusive). If two agents enter an edge exactly at the same time τ , then one of them (randomly selected) suffers a delay gap_e , which is part of its cost on edge e , and its remaining cost is determined at the delayed time, so its cost on edge e will be $gap_e + c_e(f_e(\tau + gap_e))$. The cost functions are nonnegative, continuous, and nondecreasing. The cost functions have a constant part which does not depend on the flow on the edge, and a variable part which depends on the flow on the edge. The variable part is not known to any of the actors of the model until an agent exits an edge, and reports it.
- r is the total flow given by a vector of flows with r_i denoting the flow aiming for a trip P_i from a source vertex s_i of G to a target vertex t_i of G . The flow r_i is given by $T \div n_i$ where $n_i \in [1, \dots, T]$ is a natural number constant, meaning that the following distance of the agents of the flow r_i are n_i time periods. Thus the value of r_i is between 1 and T . (Note that T , and the time unit to be modelled by T , have to be selected according to the flow values to be modelled.)
- $k = (k^1, k^2, \dots)$ is a sequence of decision vectors with decision vector $k^t = (k_1^t, k_2^t, \dots)$ made in time period t . k_i^t is the decision made by the agent of the flow r_i in time period t . The decision k_i^t is how the trip P_i is routed on a single path of the paths leading from s_i to t_i . The actual cost of a path (e_1, e_2, e_3, \dots) for a flow starting at time τ is $c_{e_1}(f_{e_1}(\tau)) + c_{e_2}(f_{e_2}(\tau + c_{e_1}(f_{e_1}(\tau)))) + c_{e_3}(f_{e_3}(\tau + c_{e_1}(f_{e_1}(\tau)) + c_{e_2}(f_{e_2}(\tau + c_{e_1}(f_{e_1}(\tau))))) + \dots$, because the actual cost of an edge is determined at the time when the flow enters the edge.

The actual cost of the edges become known for the agents in the system only when an agent reports its actual cost. Because agents do not report cost values in each time step, the agents interested in the cost values must decrease the last reported value by taking into account the time elapsed since the last reporting event (it is similar to the pheromone evaporation in [13]).

With the online routing game model, we can investigate formally if the agents are better off by making decisions based on online real-time data, or not. In the algorithmic game theory model, if there is equilibrium then the price of anarchy is the ratio between the equilibrium and the optimum. It was proved in [1] that equilibrium is not guaranteed in simple naive online routing games, therefore the equilibrium cannot be used to define the benefit of real-time data in online routing games. In order to be able to compare the costs of the agent system using online data with the costs of the agent system not using online data, three kinds of benefit of online data were defined [1] for the worst, average, and best cases.

DEFINITION 3.1. *The worst(/average/best) case benefit of online real time data at a given flow is the ratio between the cost of the maximum(/average/minimum, correspondingly) cost of the flow and the cost of the same flow with an oracle using the same decision making strategy with only the constant part of the cost functions.*

The agents are happy with online-data if the benefit values are below one. If the worst case benefit of online data is above 1, then the agents may be sometimes worse off by utilizing online data than without online data. If the best case benefit of online data is above 1, then the agents are always worse off by utilizing online data than without online data.

It is important to note here that most of the reported empirical evaluation of the different navigation systems based on real-time data focused on the average travel times, with an analysis of the distribution (for

¹In this model the graph G may contain parallel edges.

²The flow consists of units, called agents, that enter the network in one of the time periods.

³In this model cars cannot overtake the cars already on the road and there is a time gap, i.e. minimum "following distance".

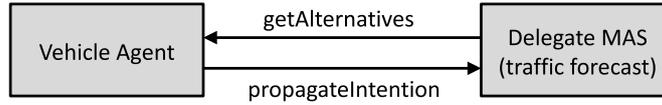


FIG. 4.1. *Intention propagation: the vehicle agent asks from the delegate MAS the travel time forecast on different routes, selects the best route, and sends back its intended route to the delegate MAS.*

example [3]), but did not include the determination of the maximum travel time. In our view this approach is good from the statistical point of view, but the users of a navigation system based on real-time data would be interested in the extreme values as well. In a practical setting, if a user experiences at least once that the travel time based on real-time data is, let's say, ten times longer than the travel time without real-time data, then, in our view, this user would never use the navigation system based on real-time data again. Therefore the evaluation of the navigation systems should include the worst case benefit defined in [1]. As we mentioned earlier, the formal analysis of the class of simple naive online routing games proved that the users of the typical commercial navigation systems in some networks may be sometimes worse off by using real-time data than by not using real-time data.

4. Intention-Propagation-Based Prediction. As it was noted in [3] and [4], one of the disadvantages of the class of simple naive online routing games is that they react upon traffic congestions only after the congestions have occurred, and they do not try to prevent them. In order to alleviate this problem, anticipatory [14] vehicle routing is implemented in the simulation environment of [3] which uses the individual planned routes of the agents to forecast future traffic density. Every vehicle is represented by a vehicle agent running on a smart device inside the vehicle. Vehicle agents communicate with the delegate multi-agent system (delegate MAS). The delegate MAS is implemented with the usage of different bio-inspired design patterns [17]. The delegate MAS represents the traffic environment, and it is able to make forecast of future traffic density based on the current traffic situation and the planned routes of the vehicles. The delegate MAS provides the traffic forecast back to the vehicle agents, which use this information to plan their trip. Fig. 4.1 shows how the delegate MAS and the vehicle agents interact in this intention propagation process. We are using this abstraction in our model, because it is independent from the actual implementation of the traffic forecaster as long as the traffic forecaster computes the forecasts in accordance with the planned routes of the vehicles.

The delegate MAS of [3] can predict future travel times based on the intention notifications that it has received from all vehicle agents. The delegate MAS has a parametrised model that describes the relationship between the travel time and the intention notifications. The parameters are continuously updated, based on both historical and real-time data, so basically the delegate MAS computes the cost functions of the online routing game model with the ability to handle varying cost functions.

If the predicted future travel times show that a new travel route is preferable, then the vehicle agent is free to change its route plan. If the vehicle agent changes its route plan, then it notifies the delegate MAS of its change of intention. The old intention is then invalidated, and the new intention is registered in the delegate MAS.

Although the vehicle agent could use several strategies to revise its intention, the decision strategy is not detailed in [3], so we assume that vehicle agents always select the shortest travel time. This route selection strategy is called simple naive decision strategy in the online routing game model.

5. Intention-Propagation-Based Prediction in Online Routing Games. Now we are going to use the above anticipatory vehicle routing system to define and formally analyse the class of online routing games that use intention-propagation-based prediction in their decision mechanism. We call this class of online routing games simple naive intention propagation online routing games.

DEFINITION 5.1. *Simple naive intention propagation online routing games (SNIP online routing games) are online routing games where*

- *the decision making agents of the flows are the vehicle agents of the anticipatory vehicle routing system,*
- *the delegate MAS of the anticipatory vehicle routing system predicts the travel times for each path of the trip P_i , based on the planned routes of the agents currently in the system,*

- and the decision k_i^t is to select the path with the shortest travel time among the predicted travel times on the different paths of the trip P_i .

The vehicle agent notifies the delegate MAS of its selected path, and the delegate MAS remembers this selection while the vehicle agent is in the road network, and the delegate MAS invalidates it when the vehicle agent exits the network.

Note that SNIP online routing games are a little bit different from the anticipatory vehicle routing system of [3], because the SNIP vehicle agents select their route when they enter the road network, and, in accordance with the online routing game model, they do not revise it during their trip. In the future, we may develop a dynamic version of the online routing game model to be able to model dynamic re-routing. The reasoning of the SNIP online routing game agent is thus the following:

Reasoning of the vehicle agent in the SNIP online routing game

```

1: // Get the predicted travel times
2: routes := getAlternatives(source,target)
3: // Select the shortest path
4: selectedRoute := choose_shortest(routes)
5: intention := selectedRoute
6: // Propagate the current intention
7: propagateIntention(intention)
8: // Instruct the driver to destination
9: instructDriver(intention)

```

We assume that agents signal their intentions truthfully. We do not know if truthfulness is individually rational or not, but in the future, the SNIP online routing game model may be used to investigate this question as well. At this time, we want to investigate if the additional information about the cognitive state of the agents already in the network helps, or not. When the agents enter the network of the SNIP online routing game, they base their decisions not only on the current status of the network (the travel times on roads), but also on the current cognitive status of the agents (the intentions of the agents) already in the network.

We investigate the class of SNIP online routing games with formal methods to see if the properties of the SNIP online routing games are different from the properties of the simple naive online routing games, or not. In SNIP online routing games the agents receive a prediction of future traffic, so we would expect that this additional information can be used to improve the properties of simple naive online routing games. Unfortunately intention-propagation-based prediction does not eliminate fully the "single flow intensification" problem, as the next Theorem shows.

THEOREM 5.1. *There are simple naive intention propagation online routing games, where the total traffic flow has only one incoming flow, i.e. $r = (r_1)$, however the flow on some of the edges sometimes may be more than r_1 .*

Proof. We are going to show how this happens in the SNIP online routing game $SN_1 = (t, T, G, r, c, k)$ which is described as follows. $T = 10$, modelling one time unit⁴. The network G is given as a three node directed graph with $V = \{v_1, v_2, v_3\}$ and $E = \{e_1, e_2, e_3\}$ as shown in Fig. 5.1. Note that the edges e_1 and e_2 are parallel, but have different cost functions. The cost function c of the road network is $c_{e_1} = 10 + x$, $c_{e_2} = 10.5 + x$, and $c_{e_3} = 1 + x$, where x is the total incoming flow on the edge. The total traffic flow is $r = (r_1)$ with $r_1 = 1$ on the trip P_1 from the source vertex v_1 to the target vertex v_3 of G . The trip P_1 has the set of paths $\{p_1, p_2\}$, where $p_1 = (e_1, e_3)$ and $p_2 = (e_2, e_3)$.

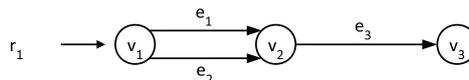


FIG. 5.1. The network of the SNIP online routing game SN_1 .

⁴In this case, the time unit consists of 10 time steps, so time 1 is at time step 10.

TABLE 5.1
Predictions, path selections and actual travel times in SN_1 .

Start at v_1	Path	Predicted at v_2	Predicted at v_3	Selected path	Actual at v_3	Travel time
0	(e_1, e_3)	10	11	(e_1, e_3)	11	11
0	(e_2, e_3)	10.5	11.5			
1	(e_1, e_3)	12	13			
1	(e_2, e_3)	11.5	12.5	(e_2, e_3)	12.5	11.5
2	(e_1, e_3)	12	14	(e_1, e_3)	14	12
2	(e_2, e_3)	13.5	14.5			
3	(e_1, e_3)	14	15			
3	(e_2, e_3)	13.5	14.5	(e_2, e_3)	14.5	11.5
4	(e_1, e_3)	14	16	(e_1, e_3)	16	12
4	(e_2, e_3)	15.5	16.5			
...

Table 5.1 shows the predictions, the path selections and the actual travel times for the first five agents of the flow r_1 . The first column of the table is the start time of the given agent of r_1 . Because $r_1 = 1$, one agent starts at the beginning of each time unit. The second column is the path for which the predictions are shown in the third and fourth column. The third column shows the predicted arrival time at vertex v_2 and the fourth column shows the predicted arrival time at vertex v_3 . The fifth column shows the path selected by the given agent of r_1 based on the prediction. The sixth column shows the actual arrival time at vertex v_3 , and the last column shows the actual travel time of the given agent. The boldface figures show the actual arrival times. The actual arrival time at vertex v_2 must always be the same as one of the predicted arrival times, because vertex v_2 is always the end vertex of the first edge of the path. This is why the actual arrival time at vertex v_2 is boldfaced in the third column.

The predictions for the agent starting at time 0 are based on the fixed parts of the cost functions, because the edges are empty. So the predicted travel time on e_1 is 10, on e_2 it is 10.5, and on e_3 it is 1. Thus the predicted arrival times at vertex v_2 are 10 on path (e_1, e_3) and 10.5 on path (e_2, e_3) . The predicted arrival times at vertex v_3 are 11 on path (e_1, e_3) , and 11.5 on path (e_2, e_3) . The predicted fastest path is (e_1, e_3) , so this agent of the flow selects this path. The next agent of the flow starts at time 1, and it cannot enter e_3 before time 11, so the agents starting at time 1 and later cannot interfere with the agent starting at time 0, and the actual arrival times of the agent starting at time 0 are the same as the predicted arrival times on path (e_1, e_3) .

The predictions for the agent starting at time 1 are more complex, because the previous agent influences the travel time on one of the roads. The predicted travel time on e_1 is 11, because the agent starting at time 0 entered this road 1 time unit before. The predicted travel time on e_2 is 10.5, because this road is empty. Thus the predicted arrival times at vertex v_2 are time 12 on path (e_1, e_3) and time 11.5 on path (e_2, e_3) . The predicted travel time on e_3 on path (e_1, e_3) is 1, because the agent starting at time 1 is predicted to enter e_3 on this path at time 12, which is more than 1 time unit later than the previous agent. Thus the predicted arrival time at vertex v_3 on path (e_1, e_3) is 13. The predicted travel time on e_3 on path (e_2, e_3) is 1 too, because the agent starting at time 1 is predicted to enter e_3 at time 11.5 which is more than 1 time unit later than the previous agent. Thus the predicted arrival time at vertex v_3 on path (e_2, e_3) is 12.5. The predicted arrival times at vertex v_3 are 13 on path (e_1, e_3) and 12.5 on path (e_2, e_3) . The predicted fastest path is (e_2, e_3) , so this agent of the flow selects this path.

The predicted travel time for the agent starting at time 2 on e_1 is 10, because the last agent entered this road more than 1 time unit before. The predicted travel time on e_2 is 11.5, because the agent starting at time 1 entered this road 1 time unit before. Thus the predicted arrival times at vertex v_2 are time 12 on path (e_1, e_3) and time 13.5 on path (e_2, e_3) . The predicted travel time on e_3 on path (e_1, e_3) is 2, because the agent starting at time 2 is predicted to enter e_3 on this path at time 12, and one agent entered e_3 within 1 time unit. Thus the predicted arrival time at vertex v_3 on path (e_1, e_3) is 14. The predicted travel time on e_3 on path (e_2, e_3) is 1, because the agent starting at time 2 is predicted to enter e_3 at time 13.5 which is more than 1 time unit later than the previous agent. Thus the predicted arrival time at vertex v_3 on path (e_2, e_3) is 14.5. The predicted

fastest path is (e_1, e_3) , so this agent of the flow selects this path.

The agent starting at time 2 enters e_3 at time 12, and it does not interfere with the agent starting at time 1, so the actual arrival times of the agent starting at time 1 are the same as its predicted arrival times on path (e_2, e_3) .

The predicted travel time for the agent starting at time 3 on e_1 is 11, because the last agent entered this road 1 time unit before. The predicted travel time on e_2 is 10.5, because the agent starting at time 2 entered this road more than 1 time unit before. Thus the predicted arrival times at vertex v_2 are time 14 on path (e_1, e_3) and time 13.5 on path (e_2, e_3) . The predicted travel time on e_3 on both paths is 1, because the agent starting at time 3 is predicted to enter e_3 on this path at time 13.5 or later, which is more than 1 time unit later than the previous agent that started at time 1 (and is predicted to enter e_3 at time 12). Thus the predicted arrival time at vertex v_3 on path (e_1, e_3) is 15, and the predicted arrival time at vertex v_3 on path (e_2, e_3) is 14.5. The predicted fastest path is (e_2, e_3) , so this agent of the flow selects this path.

The agent starting at time 3 enters e_3 at time 13.5 and does not interfere with the agent starting at time 2, so the actual arrival times of the agent starting at time 2 are the same as its predicted arrival times on path (e_1, e_3) .

The rows for the agent starting at time 4 can be filled out with similar reasoning as the rows for the agent starting at time 2.

As Table 5.1 shows, the agent starting at time 1 and the agent starting at time 2 arrive at vertex v_2 and enter edge e_3 at time 11.5 and 12 correspondingly. The agent starting at time 3 and the agent starting at time 4 enter edge e_3 within one time unit as well. This means that, from time to time, two agents enter edge e_3 within one time unit, therefore the flow on e_3 will be 2 for a while, which is more than $r_1 = 1$. \square

The above Theorem 5.1 shows that "single flow intensification" may happen in some SNIP online routing games, but it does not happen the same way as in simple naive online routing games. Prediction helps to share the resources, and the flow alternates between the parallel roads e_1 and e_2 . Because of this, we cannot prove the same way as it is done in [1] that the worst case benefit of online data in SNIP online routing games may be more than 1. We are giving below an alternative proof in Theorem 5.2 for this.

THEOREM 5.2. *There are simple naive intention propagation online routing games where the worst case benefit of online real-time data is greater than 1.*

Proof. We are going to show how this is possible in the SNIP online routing game $SN_2 = (t, T, G, r, c, k)$ which is described as follows. $T = 10$, modelling one time unit. The network G is given as a four node directed graph with $V = \{v_0, v_1, v_2, v_3\}$ and $E = \{e_1, e_2, e_3, e_4\}$. G and its edges are shown in Fig. 5.2. Note that the edges e_3 and e_4 are parallel, but have different cost functions. The cost function c of the road network is $c_{e_1} = 1$, $c_{e_2} = 1$, $c_{e_3} = 10 + x$, and $c_{e_4} = 10.5 + 10 \times x$, where x is the total incoming flow on the edge. The total traffic flow is $r = (r_1, r_2)$ with $r_1 = 1$ on the trip P_1 from source vertex v_0 to target vertex v_3 of G and $r_2 = 1$ on the trip P_2 from source vertex v_1 to target vertex v_3 of G . The trip P_1 has the set of paths $\{p_1, p_2\}$, where $p_1 = (e_1, e_3)$ and $p_2 = (e_1, e_4)$. The trip P_2 has the set of paths $\{p_3, p_4\}$, where $p_3 = (e_2, e_3)$ and $p_4 = (e_2, e_4)$.

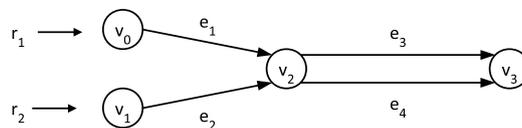


FIG. 5.2. The network of the SNIP online routing game SN_2 .

In order to compute the benefit of online data, we first determine the non-online cost of the flow r with an oracle using the SNIP decision making strategy but using only the constant part of the cost functions. In this case the predicted cost is the constant part of the cost functions, and the agents always select the shortest path based on the constant part of the cost functions. Thus flow r_1 always goes on the path $p_1 = (e_1, e_3)$, and flow r_2 always goes on the path $p_3 = (e_2, e_3)$. The flow on edge e_3 will be 2, because one agent from r_1 and one agent from r_2 arrive at v_2 in every time unit, and one of them enters e_3 immediately, and the other one with gap_{e_3} delay. We define SN_2 with $gap_{e_3} < T$. So the cost of path $p_1 = (e_1, e_3)$ is $c_{p_1}^{oracle} = 1 + 10 + 2 = 13$ and the cost

of path $p_3 = (e_2, e_3)$ is $c_{p3oracle} = 1 + 10 + 2 = 13$. The total cost of the flow r is $c_{roracle} = 1 \times 13 + 1 \times 13 = 26$ with the non-online oracle.

Now we are going to prove that the cost of the flow r may be above 26 in the SNIP online routing game SN_2 . When the flow r starts to flow at time step $t = 0$, the traffic forecaster will predict 10 for the cost of e_3 and 10.5 for the cost of e_4 , so both r_1 and r_2 will select the edge e_3 . The flows propagate their intentions, so the traffic forecaster will know that a flow of 2 agents will enter edge e_3 from time step $t = 10$ (time step $t = 10$ is time 1, and one agent enters e_3 at time 1, and one agent at time $1 + gap_{e_3}$), and 2 agents will be on e_3 until at least time step $t = 110$ (time 11). Both r_1 and r_2 are 1, so they make their next decision at time step $t = 10$ (time 1). At this time the traffic forecaster tells both r_1 and r_2 that this agent of their flow will arrive at v_2 at the time step $t = 20$ (time 2) facing e_3 with cost 12 (because two agents entered e_3 in the preceding time unit) and e_4 with cost 10.5. Note that the traffic forecaster is only aware of the intention propagations before the current time step, and the traffic forecaster does not forecast the decisions at the current time step. Because the predicted cost of e_4 is smaller than the predicted cost of e_3 , both r_1 and r_2 will select e_4 . Both will arrive at v_2 at time step $t = 20$ (time 2), and one of them (let's say r_1) will enter the edge e_4 at time 2 with a cost of 10.5, and the other one will enter the edge e_4 at time $2 + gap_{e_4}$ with a cost of $10.5 + 10 \times 1 = 20.5$. We define SN_2 with $gap_{e_4} < T$. So the flow r at the time step $t = 10$ will decide to go on the paths $p_2 = (e_1, e_4)$ and $p_4 = (e_2, e_4)$. The cost of the path p_2 will be $c_{p2online}(1) = 1 + 10.5 = 11.5$, and the cost of the path p_4 will be $c_{p4online}(1) = 1 + 20.5 = 21.5$. The total cost of the flow r will be $c_{ronline}(1) = 1 \times 11.5 + 1 \times 21.5 = 33$ with the SNIP online strategy.

From the above calculations $c_{ronline}(1) = 33 > c_{roracle} = 26$, so the worst case benefit of online data in the SNIP online routing game SN_2 is greater than 1. \square

The above theorem points out that worst case benefit of online data below 1 is not guaranteed in SNIP online routing games. There is even stronger statement, because the worst case benefit of online data can be arbitrarily large, making the agents very unhappy, as the next theorem shows. Note, that this is not specific to intention propagation, just this property was not investigated for simple naive online routing games.

THEOREM 5.3. *Given any arbitrarily large number α , there are simple naive intention propagation online routing games with cost functions of the form $a \times x + b$, where the worst case benefit of online real-time data is bigger than α .*

Proof. Let us take the SNIP online routing game $SN_2 = (t, T, G, r, c, k)$ from the above Theorem 5.2, and formulate its cost function c_{e_4} in a parametrized way by replacing the 10 coefficient with a $26 \times \alpha$ parameter: $c_{e_4} = 10.5 + 26 \times \alpha \times x$. Repeating the reasoning as in Theorem 5.2, we get that $c_{p4online}(1) = 1 + 10.5 + 26 \times \alpha \times 1 = 11.5 + 26 \times \alpha$, and $c_{ronline}(1) = 1 \times 11.5 + 1 \times (11.5 + 26 \times \alpha) = 23 + 26 \times \alpha$. The worst case benefit of online data is at least $c_{ronline}(1) \div c_{roracle} = (23 + 26 \times \alpha) \div 26 = 23 \div 26 + \alpha$ which is bigger than α . \square

The third problem investigated in [1] is the question of equilibrium, and it is shown that equilibrium is not guaranteed in simple naive online routing games, because they may "fluctuate" at some flow values in some games. Now the question is whether intention propagation can help to avoid "fluctuation"?

THEOREM 5.4. *There are simple naive intention propagation online routing games which do not have equilibrium at certain flow values.*

Proof. The SNIP online routing game SN_2 of Theorem 5.2 is an example which does not have equilibrium and shows fluctuation. We are going to continue the scenario of Theorem 5.2, and we show what will be the decisions of the flows after time step $t = 10$. The flow r at time step $t = 10$ decided to go on the paths $p_2 = (e_1, e_4)$ and $p_4 = (e_2, e_4)$, and there will be a flow on the edge e_4 from time step $t = 20$ (time 2) with cost 21.5. Because overtaking is not allowed on the edges, the cost of the edge e_4 will be 21.5 from time step $t = 20$, and this cost will linearly decrease as time goes by, reaching the value 12 at time step $t = 115$ (i.e. 95 time steps later), if no other flow enters the edge e_4 . If the full flow r repeatedly goes on e_3 , then the cost of e_3 is 12 (see the reasoning in Theorem 5.2), so the traffic forecaster will predict maximum 12 for the cost of e_3 and more than 12 for the cost of e_4 between time steps $t = 20$ and $t = 115$. This means that the flow r will select the edge e_3 until time step $t = 110$. Time step $t = 110$ will be the next decision point when the traffic forecaster will predict value below 12 for the cost of e_4 , because it will tell both r_1 and r_2 that the current agent of their flow will arrive at v_2 at time step $t = 120$ (time 12) facing e_3 with cost 12 and e_4 with cost 11.5. Because the predicted cost of e_4 is smaller than the predicted cost of e_3 , both r_1 and r_2 will select e_4 . Both will arrive at v_2

at time step $t = 120$ (time 12), and one of them (let's say r_1) will enter the edge e_4 with a cost of 11.5, the other one will enter the edge e_4 with a cost of $10.5 + 10 \times 1 = 20.5$ (see the reasoning in Theorem 5.2). So the flow r at the time step $t = 110$ will decide to go on the paths $p_2 = (e_1, e_4)$ and $p_4 = (e_2, e_4)$. The cost of the path p_2 will be $c_{p_2online}(11) = 1 + 11.5 = 12.5$, and the cost of the path p_4 will be $c_{p_4online}(11) = 1 + 10.5 + 10 \times 1 = 21.5$. The total cost of the flow r will be $c_{ronline}(11) = 1 \times 12.5 + 1 \times 21.5 = 34$ with the SNIP online strategy. After time step $t = 110$, the cycle will start again: the flow r will return to edge e_3 , then after a while there will be a short switch to e_4 , then e_3 again, etc. The flow r will not have an equilibrium, and its cost will fluctuate between 13 and 34. \square

Now we have analysed SNIP online routing games, and we have proved that the three properties proved for simple naive online routing games in [1] are true for SNIP online routing games as well. However we can observe that SNIP online routing games manifest these properties in a different way, which we could describe informally as "not so intensive", because prediction helps to detect the problems even before they actually occur. So intention-propagation-based prediction helps to smooth down the problems, but cannot always prevent them.

6. Conclusions. In this paper we have investigated whether intention-propagation-based prediction solves some problems of collective adaptive systems composed of autonomously self-adapting navigation devices. The formal investigation was done on the SNIP online routing game model, which is a somewhat simplified model of the anticipatory vehicle routing system using delegate multi-agent systems described in [3], because the SNIP online routing game model does not include dynamic re-routing.

Online routing games are a novel formal model of autonomously self-adapting vehicle navigation systems using real-time data. The class of simple naive online routing games are the model of current commercial vehicle navigation systems, because current commercial navigation systems use the simple naive decision strategy, which searches the (possibly multiple objective [2]) shortest path, with taking into account the real-time information available at decision time. It is proved [1] that in simple naive online routing games equilibrium is not guaranteed, "single flow intensification" is possible and the benefit of online data is not guaranteed to be below 1. In practical terms these mean that if the vehicles use simple naive strategy, and they exploit real-time data, then the traffic may sometimes fluctuate: if there is congestion on some road, then many vehicles try to adapt to the situation, they select other route, and they cause congestion on some other road. Congestion may be created by the "single flow intensification" phenomenon as well, because vehicles entering the road network later may select alternative faster routes, and they may catch up with the vehicles already on the road, and this way they cause congestion. The result of all these is that sometimes the traffic may be worse off by exploiting real-time information than without exploiting real-time information. One of the causes of this strange behaviour is that vehicles become aware of congestions with delay, only after the congestion is formed. In order to improve this situation, anticipative adaptation techniques [17], among them the techniques of intention propagation and traffic prediction [3], was proposed.

In this paper we have proved that online routing games with intention-propagation-based prediction may also have "single flow intensification" (although in a somewhat "lighter" way), the worst case benefit of online data may go above 1 (in fact it may be arbitrarily large), and the traffic may fluctuate. In practical terms these mean that if vehicles use simple naive strategy, and they exploit intention-propagation-based prediction, then in some networks and in some cases the traffic may be worse off by exploiting real-time information than without exploiting real-time information.

We have pointed out that one of the causes of this surprising result is that the traffic forecaster predicts future traffic conditions based on the intentions of the vehicles already on the road, but it does not predict the intentions of the vehicles currently making decisions. If several vehicles make decisions at the same time, then they may make the same decision to take the same alternative route to avoid the already predicted congestion, and they cause congestion on the alternative route. Obviously, intention propagation helps the vehicles to detect the possibility of congestion formation before the congestion is actually formed, and thus there is smaller "time window" to make the same "wrong" decision to head towards the newly forming congestion. The technique of intention-propagation-based traffic forecast is therefore an important improvement to the simple naive online strategy.

Although intention-propagation-based traffic forecast is an important improvement, it does not fully take us closer to the goal that was set out in [1]: "we expect that the application of these new strategies will

be individually rational choice, and therefore the decision strategies can be implemented in the navigation devices themselves instead of the centralized planning approaches like those of Google Maps and Waze, because some users are reluctant to provide private data for the centralized approach". The implementation of intention propagation and traffic prediction involves a delegate MAS which is similar to GoogleMaps and Waze. Although the delegate MAS is distributed, it plays a kind of "central actor" role, because the delegate MAS may collect data about the agents similarly as Google Maps and Waze.

The results of this paper point out future research directions. One direction is to apply other "social" information sharing patterns [17], like gossiping, among vehicle agents to eliminate the "central actor" role of the delegate MAS, so that the vehicle agents have an integrated view of the intentions of all of the vehicle agents, but none of the agents in the system is able to discover the concrete intention of any other agent. Another research direction is to develop new decision strategies that can help to avoid that many vehicles make the same decision at the same time to cause congestion on the newly selected route. We hope that if such strategy is developed, then it may help us to keep the worst case benefit of online data below 1, and thus we can ensure that the users of navigation systems exploiting real-time data will never experience worse performance with online data than without online data.

Because we based our investigations on the online routing game model, our results can be transferred to other application fields than road traffic navigation. In the road traffic navigation field the goals are trips, the shared resources are roads, and the costs are travel times. For example, in cloud computing the goals are jobs to be done, the resources are computing resources, and costs are execution times; in intelligent manufacturing, the goals are workflows to be executed, the resources are manufacturing phases, and the costs are manufacturing times or real financial costs. These other fields can be modelled with online routing games or with online joint resource utilization games as defined in [1].

Acknowledgments. The author thanks the anonymous reviewers of the previous version of this paper which was presented at the FoCAS workshop of the eighth edition of the IEEE International Conferences on Self-Adaptive and Self-Organizing Systems, and the reviewers of the current version. The comments of the reviewers and the participants of the workshop helped to prepare this improved and extended version.

REFERENCES

- [1] LÁSZLÓ Z. VARGA, *Online Routing Games and the Benefit of Online Data*, in Proc. Eighth International Workshop on Agents in Traffic and Transportation, at 13th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-6, 2014, Paris, France, pp. 88-95.
- [2] VICTOR J. BLUE, JEFFREY L. ADLER, GEORGE F. LIST, *Real-Time Multiple-Objective Path Search for In-Vehicle Route Guidance Systems*, in Transportation Research Record No. 1588, Intelligent Transportation Systems and Artificial Intelligence, vol. 1588, pp. 10-17, 1997
- [3] R. CLAES, T. HOLVOET, AND D. WEYNS, *A decentralized approach for anticipatory vehicle routing using delegate multi-agent systems*, in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 2, pp. 364-373, 2011.
- [4] JOACHIM WAHLE, ANA LÚCIA C BAZZAN, FRANZISKA KLÜGL, MICHAEL SCHRECKENBERG, *Decision dynamics in a traffic scenario*, Physica A: Statistical Mechanics and its Applications, Vol. 287, Issues 34, pp. 669-681, 2000
- [5] NOAM NISAN, TIM ROUGHGARDEN, EVA TARDOS, AND VIJAY V. VAZIRANI, *Algorithmic Game Theory*, Cambridge University Press, New York, NY, USA. 2007
- [6] *Dynamic Traffic Assignment: A Primer*, in Transportation Research Circular Number E-C153, Transportation Research Board, 500 Fifth Street, NW, Washington, DC 20001, 2011
- [7] TIM ROUGHGARDEN AND ÉVA TARDOS, *How bad is selfish routing?*, J. ACM 49, 2, 236-259 (March 2002)
- [8] D. BRAESS, *Über ein Paradoxon aus der Verkehrsplanung*, Unternehmensforschung 12, 258 (1968)
- [9] D.C. PARKES, *Online mechanisms*, in [5] pages 411-439
- [10] S. ALBERS, *Online algorithms*, in Interactive Computation: The New Paradigm edited by D.Q. Goldin, S.A. Smolka and P. Wegner, Springer, pp. 143-164, 2006.
- [11] A. GALSTYAN, S. KOLAR, AND K. LERMAN, *Resource Allocation Games with Changing Resource Capacities*, in Proc. of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2003), Melbourne, Australia, page 145-152, 2003
- [12] J. GÖRMER, J. F. EHMKE, M. FIOSINS, D. SCHMIDT, H. SCHUMACHER, AND H. TCHOUANKEM, *Decision support for dynamic city traffic management using vehicular communication*, in Proc. of 1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications. SciTePress Digital Library, pp. 327332, 2011
- [13] J. DALLMEYER, R. SCHUMANN, A.D. LATTNER, AND I.J. TIMM, *Dont Go with the Ant Flow: Ant-inspired Traffic Routing in Urban Environments*, in 7th International Workshop on Agents in Traffic and Transportation (ATT 2012) held at AAMAS 2012, pp. 59-68. 2012

- [14] ROBERT ROSEN, *Anticipatory Systems: Philosophical, Mathematical, and Methodological Foundations (IFSR International Series on Systems Science and Engineering)*, Pergamon Press, 1985
- [15] TOM DE WOLF AND TOM HOLVOET, *Design Patterns for Decentralised Coordination in Self-organising Emergent Systems*, in *Engineering Self-Organising Systems*, Lecture Notes in Computer Science, Vol. 4335, pp. 28-49, 2007
- [16] O.BABA OGLU, G.CANRIGHT, A.DEUTSCH, G.A.D.CARO, F.DUCATELLE, L.M. GAMBARDELLA, N. GANGULY, M. JELASITY, R. MONTEMANNI, A. MONTRESOR, AND T. URNES, *Design Patterns from Biology for Distributed Computing*, ACM Trans. Auton. Adapt. Syst., Vol. 1, No. 1, pp. 2666, 2006
- [17] JOSE LUIS FERNANDEZ-MARQUEZ, GIOVANNA DI MARZO SERUGENDO, SARA MONTAGNA, MIRKO VIROLI, JOSEP LLUIS ARCOS, *Description and Composition of Bio-inspired Design Patterns: a Complete Overview*, Natural Computing, Volume 12, Issue 1, pp 43-67, 2013
- [18] SARA MONTAGNA, MIRKO VIROLI, JOSE LUIS FERNANDEZ-MARQUEZ, GIOVANNA DI MARZO SERUGENDO, FRANCO ZAMBONELLI, *Injecting Self-Organisation into Pervasive Service Ecosystems*, Mobile Networks and Applications, Volume 18, Issue 3, pp. 398-412, 2013

Edited by: Giacomo Cabri and Emma Hart

Received: Dec 9, 2014

Accepted: Jul 7, 2015