Networked malicious agents in transportation systems: a simulation study

Anderson Rocha Tavares[1] and Ana Lucia Cetertich Bazzan[1]

Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil,
{artavares,bazzan}@inf.ufrgs.br,

Abstract. Inter-vehicular communication (IVC) technologies are receiving increasing attention both from academy and industry. In a scenario where vehicles exchange road traffic information with one another, cheating agents that spread false information through IVC for their benefit may appear. In this work, we model a transportation network as a multiagent system with a group of networked agents that use IVC to spread false road traffic information. Performing microscopic traffic simulation, we assess the effect of this group of cheating agents on the road network both when they act individually, trying to divert other drivers from their own routes and when they act coordinately, trying to divert other drivers from the routes of all agents in the group.

Keywords: inter-vehicular communication, multiagent systems, intelligent transportation systems

1 Introduction

Traffic problems are a key topic in modern societies. In order to deal with the increasing demand for transportation, methods to optimize the management of both the supply (that expresses the infrastructure capacity) and the demand (the transport needs) are preferred over the capacity expansion of the infrastructure, since the second approach has social, economic and environmental consequences.

In the literature, we can find several works that tackle traffic issues using various multiagent approaches in simulated environments. Some examples include but are not restricted to minority games [3], reservation-based systems to intersection control [2], market-based approaches over the reservation-based system [9] and the use of inter-vehicular communication (IVC) [4].

The present work focuses on IVC, which consists of the exchange of information by vehicles through wireless networks, especially vehicular ad-hoc network (VANET). Although studies on IVC date back to the 1980's, only recently more attention has been paid to the presence of self-interested or malicious agents in IVC scenarios. Some works on this topic are detailed in Section 2.

In this work, we present a contribution on the topic of malicious agents in IVC scenarios. Here, we model a transportation network as a multiagent system with driver agents that are cost-minimizers. Those exchange messages with one

another in order to gain more information about the road network state, thus being able to build better routes to their destinations. We model malicious agents that spread false information in order to divert other drivers from their routes. We want to assess the effect of these malicious agents acting as a networked, coordinated group.

We identify two aspects to be investigated. First, from the malicious agents' point of view, we want to assess their performance when they coordinate themselves to cheat others. To answer this question, we test the gains obtained by these agents when malicious agents cheat individually (each one spread false information about its own route) compared to when they have a network of cooperation and act coordinately (each one spreads false information about the routes of all agents in the group of cheating agents). The second aspect to be investigated concerns the system as a whole. We want to assess how many networked malicious agents are necessary in order to cause a significant drop in the overall system's performance.

In order to investigate these aspects, we conduct simulations that take into account not only the agents' decision-making process but also consider their microscopic movement through the road network as well as issues related to IVC. This represents a contribution that meets our long term agenda, which consists of proposing a methodology to integrate behavioral models of human travelers reacting to traffic patterns and control measures of these traffic patterns, focusing on distributed and decentralized methods.

The remaining of this paper is organized as follows: in Section 2, we review works that consider the existence of malicious agents in IVC scenarios; in Section 3, we present our driver agent, traffic simulation and inter-vehicular communication models. Section 4 presents the studied scenario and our experiment results. Section 5 presents concluding remarks and points directions for future investigation.

2 Related work

In this section, we review works focused in the presence of malicious agents in inter-vehicular communication scenarios, showing their contributions, similarities and limitations that are tackled in the present work.

From a multiagent system perspective, since the statement of the *Byzantine generals problem* (BGP) [5], where processing units must deal with defective units that send false information to the system, several attempts have been made to model and detect malicious agents in multiagent systems. Examples include, but are not restricted to the OM algorithm that solves the BGP [5], resilient gossip protocols [6] and distributed reputation systems [11]. These approaches would have limitations in applicability over vehicular networks, as it is assumed that either the data exchanged through communication or the communication network topology is static and this is not the case on vehicular networks: drivers communicate with several different peers in short periods of time and the propagated road information is related to the traffic flow, thus being dynamic.

Moreover, the focus in the present work is to study the effect of the malicious information on drivers' behavior on the road network, an issue that is not addressed in these previously mentioned works.

A work focused on the assessment of drivers' behavior on the presence of malicious messages on IVC scenarios appears in [8]. The authors model cooperation for the elements in a malicious coordinated agents (MCA) group through a social network. Malicious agents spread false information on the roads used by themselves and by the agents they are connected with. The authors theoretically determine the Nash equilibrium in the corresponding congestion game. Experimentally, they evaluate the performance of both the MCA group and the whole system according to the degree of the social network formed by the MCA. In the present work, we assume that all agents in the MCA group help one another, i.e., the MCA's social network is fully connected. We explain this assumption in Section 3.2.

In [4], the presence of malicious drivers in transportation systems is also studied. Malicious agents either try to divert drivers from their routes or cause disorder in the road network, that is, to increase the travel times of all agents in an act of vandalism. In the former behavior (divert other drivers from their routes), the group of malicious agents have the same origin and destination points and spread false information about the roads in the shortest path between these points in order to divert other drivers from them. In the present work, the group of malicious agents have different origin and destination points. They help one another by spreading false information about the roads in the routes of each malicious agent. In the present work, we do not model the situation where malicious agents try to cause disorder in the road network.

The experiments performed both in [4] and [8] have a limitation on the level of detail of the traffic simulation: in these works, the traffic movement model does not simulate the physical location of an agent in the road it is traversing. This forces the adoption of a simplified abstraction of the communication range: drivers can exchange messages when passing one another in opposing directions and with their peers in a queue that exists when they are about to exit a road. In the present work, this issue is tackled by the adoption of a microscopic traffic simulation model. In this level of detail, drivers are modeled as individual entities with their physical location on the road network and movement rules precisely defined. With the physical location of the drivers in the road network, we can simulate an actual communication radius in which they can exchange messages. This represents an advance towards the integration of traffic and data network simulators, which makes it easier to perform more detailed studies of IVC issues, thus accelerating the adoption of IVC technology in the real world. The traffic and communication model adopted in the present work are detailed in Section 3.3.

3 Description of the approach

3.1 Driver agents

Let D be the set of drivers. Each driver $d \in D$ is modeled as an agent whose goal is to minimize the travel time over its route R_d, which consists of a set of links from d's origin ($\ell_d^{\uparrow}$) to d's destination ($\ell_d^{\downarrow}$). Both $\ell_d^{\uparrow}$ and $\ell_d^{\downarrow}$ are assigned according to a probability distribution Ω over the origins and destinations of the road network. At departure time, the driver d estimates its travel time and stores it in $\tilde{t}_{d,l}$ for each $l \in R_d$.

Every driver has a model of the road network, which is represented by a graph $G = (N, L)$, where N is the set of nodes (intersections) and L are the links among these nodes, representing the road sections of the road network. We remark that this assumption is not far from reality given navigation systems already existing. The weight of a given link $l \in L$ that a driver $d \in D$ knows (denoted by $t_{d,l}$) is the travel time required to traverse that link. Also, a driver d records how old the gathered information about a link l is, and stores it in $a_{d,l}$.

Both $t_{d,l}$ and $a_{d,l}$ are updated whenever the driver leaves a given link or when it receives messages from other drivers. When driver d finishes traversing a link l, $t_{d,l}$ receives the travel time d spent on l and $a_{d,l}$ receives zero.

When d receives a message with an information about travel time on link l, d updates $t_{d,l}$ using the age of the received information to decide how the information will be incorporated. The weight of the received information decays exponentially with its age. The function $\gamma : \mathbb{R}_{\geq 0} \rightarrow [0 : 1]$ (Eq. 1) illustrates this. Parameter β adjusts how fast it decays, that is, how less important the information will become as it ages. The travel time information update rule is given in Eq. 2. In this equation, $t_{d,l}$ is the travel time that driver d knows for link l, $t'_{c,l}$ is the information about link l received from a given driver c and $a_{c,l}$ is the age of $t'_{c,l}$. Equation 2 also shows that driver d updates the age of the information about link l based on the age of the received information.

$$\gamma(x) = e^{(\frac{-x}{\beta})} \tag{1}$$

$$\begin{aligned} t_{d,l} \leftarrow \gamma(a_{c,l}) t'_{c,l} + [1 - \gamma(a_{c,l})] t_{d,l} \\ a_{d,l} \leftarrow a_{c,l} \end{aligned} \tag{2}$$

Regarding the message construction, regular (non-cheating) driver agents report the actual data they know. That is, for a non-cheating driver d, the reported travel time $t'_{d,l}$ receives $t_{d,l}$ for all $l \in L$ and the information age is truthfully transmitted.

In this work, we model drivers that perform en-route replanning, i.e., route recalculation during the trip. Drivers that are able to perform en-route replanning are represented by the set D°. A driver $d \in D^{\circ}$ triggers the route recalculation process when it becomes aware that the time to complete its trip will be higher than it is willing to tolerate. When this happens, a new path is calculated from

the current location to d's destination and the expected travel time to complete the trip is updated. A driver that does not perform en-route replanning belongs to the $D - D^{\circ}$ set. A driver from $D - D^{\circ}$ can, however, calculate a different route when a new trip starts.

3.2 Networked malicious drivers

Malicious or cheating drivers spread false information to other drivers through inter-vehicular communication. The set of malicious drivers is denoted by $D^{\sharp}$. As $D^{\sharp} \subset D$, a malicious driver has the same properties of a regular driver described in Section 3.1 plus a tampered IVC device that the agent can configure to send messages with false information both about travel time and information age.

A malicious driver d sends false information about the link l that it considers interesting to cheat about by reporting a high travel time in $t'_{d,l}$ and setting the information age $a_{d,l}$ to zero. This means that false information is considered by non-cheating drivers with a high weight (refer to the information update rule at Eq. 2).

A malicious driver d will report the actual data it knows for a link l that is not interesting for it. In this case, $t'_{d,l}$ receives the actual $t_{d,l}$ and the information age $a_{d,l}$ is not altered. The set of interesting links depends on the behavior adopted by the malicious drivers.

In this work, we define two behaviors for the malicious agents regarding their coordination. First, as uncoordinated agents, each $d \in D^{\sharp}$ will send false information about the links on its own route. In this case, each driver $d \in D^{\sharp}$ can be regarded as an individual who has tampered with its IVC device on its own, in order to benefit itself by diverting other drivers from its route.

In the second behavior, drivers from $D^{\sharp}$ are networked and act coordinately. In this configuration, a malicious driver will report false information for all links in $L^{\sharp} = \{l \mid l \in R_d \ \forall \ d \in D^{\sharp}\}$, that is, false information will be reported for all links that belong to a route from any malicious driver. The reason for this is, for instance, $d \in D^{\sharp}$ belongs to a fleet or deliver service, where this kind of group behavior is either mandatory or benefits all. For instance, the group of networked malicious agents can be understood as the drivers from a delivery company with several branches and clients whose goal is to deliver their goods in the lowest possible time. In this case, we denominate the group of networked cheating agents as the malicious fleet. Before leaving the branches of the company, the fleet agents communicate their routes with one another, building their shared knowledge of the links in $L^{\sharp}$ used by them. As an example, this prior communication could be done via the company's centralized information system, which would have the information about all clients served and the planned routes from the branches to the clients.

Due to deliver constraints, malicious drivers rarely perform route replanning, so that $D^{\sharp} \subset D - D^{\circ}$. Besides, because a malicious driver $d \in D^{\sharp}$ tries to divert other drivers, he does not change its own route while on trip, thus avoiding to enter a route that is congested by other drivers that were diverted from d's original route. Also, d ignores information reporting congestion on the links that

belong to either (a) its route R_d when acting individually or (b) the set of links used by the fleet ($L^\sharp$) when acting in a networked, coordinated way.

3.3 Traffic simulation and IVC

Contrarily to many works that use an abstract, macroscopic simulation model, in this work we use a microscopic simulation model based on car-following. In this model, a vehicle's operational behavior regarding acceleration or braking is influenced by its leading vehicle. The adopted model is accident-free and implements driving behavior regarding lane-changing, priorities of roads and reaction to traffic lights.

The simulation is continuous in space and discrete in time, so that we have the precise physical location of the vehicles in the road network. The basic time unit of the simulation is one timestep, which corresponds to one second in the real world.

Each driver d in the network exchanges messages with other drivers physically located within a communication radius δ around d.

A message constructed by a given driver d consists of a set of tuples ($t'_{d,l}$, $a_{d,l}$) for every link l in the road network. The first element of the tuple is the travel time reported by driver d for link l. The second element is the age of information $t'_{d,l}$.

3.4 Driver agents behavior

For each driver d, the set $\{t_{d,l} \ \forall \ l \in L\}$ represents d's knowledge base. Drivers initialize their knowledge bases optimistically with the free-flow travel time[1] of the road network links. The information age is set to infinity for each link of the road network, meaning that information about the links has never been gathered.

The process of en-route replanning is implemented as follows: when the trip begins, the driver calculates its expected travel time, which consists of the sum of the travel times that it knows for the links in its route. During the trip, if the travel time spent since trip started plus the travel time on the remaining links of the route is higher than the expected travel time calculated at trip start multiplied by a delay tolerance factor, the driver will recalculate its route and update the expected travel time in it. Route recalculation is activated only when driver has completed between 20% and 70% of its trip, because obviously recalculating routes too early or too late does not improve the performance.

Regarding the message construction, for a malicious driver d spreading false information about link l, the reported travel time ($t'_{d,l}$) is set to $3 \times f_l$, where f_l is l's free-flow travel time. This is described as the lowest level of service in [7, p. 10-5] for urban roads. This way, d is reporting that the link is congested and should be avoided.

[1] This is the achieved travel time when traffic volumes are sufficiently low that drivers are not influenced by the presence of other vehicles [7].

Algorithm 1 describes the drivers initialization, route calculation and en-route replanning procedures. In this algorithm, the function *shortestPath* receives the origin and destination links plus the weights of the road network links that, for each driver, correspond to its knowledge base.

Algorithm 1 Driver agents

procedure InitializeDrivers(Ω)
  for all $d \in D$ **do**
    $(\ell_d^{\uparrow}, \ell_d^{\downarrow}) \leftarrow selectOrigDest(\Omega)$
    for all $l \in L$ **do**
      $t_{d,l} \leftarrow f_l$ ▷ free-flow travel time of l
      $a_{d,l} \leftarrow \infty$
    end for
  end for
end procedure

procedure CalculateRoutes
  for all $d \in D$ **do**
    $R_d \leftarrow shortestPath(\ell_d^{\uparrow}, \ell_d^{\downarrow}, \{t_{d,l} \ \forall \ l \in L\})$
    $\tilde{t}_{d,l} \leftarrow t_{d,l} \ \forall \ l \in R_d$
  end for
  $L^{\sharp} \leftarrow \{l \mid l \in R_d \ \forall \ d \in D^{\sharp}\}$
end procedure

procedure RecalculateRoutes(τ)
  for all $d \in D^{\circ} - D^{\downarrow}$ **do**
    if {d has completed between 20% and 70% of the trip} **then**
      $\ell_d^{\circ} \leftarrow$ {current link of driver d}
      $\hat{t}_d \leftarrow$ {travel time since trip started}
      $R'_d \leftarrow$ {links not yet traversed on route R_d}
      $R_d^{*} \leftarrow shortestPath(\ell_d^{\circ}, \ell_d^{\downarrow}, \{t_{d,l} \ \forall \ l \in L\})$
      if $\hat{t}_d + \sum_{l \in R'_d} t_{d,l} > \tau \times \sum_{l \in R_d} \tilde{t}_{d,l}$ **then**
        $R_d \leftarrow (R_d - R'_d) \cup R_d^{*}$
        $\tilde{t}_{d,l} \leftarrow$ {travel time spent on $l \ \forall \ l \in R_d - R'_d$}
        $\tilde{t}_{d,l} \leftarrow t_{d,l} \ \forall \ l \in R_d^{*}$
      end if
    end if
  end for
end procedure

3.5 Simulation

For the microscopic simulation, we use the traffic simulator SUMO [1]. The IVC simulator and the drivers' behavior modules are built in an external program that communicates with SUMO during runtime through its implementation of

the traffic control interface protocol (TraCI) [10]. This way, at every timestep, SUMO sends the information requested by our modules. The information is processed, and the modules return commands to SUMO regarding the drivers actions. These commands are performed in the next timestep and the process is repeated until the simulation is ended.

The message construction and exchanging processes are formalized in Alg. 2.

Algorithm 2 Communication

procedure MsgConstruction
  for all $d \in D - D^{\downarrow}$ **do**
    for all $l \in L$ **do**
      if {malicious agents act as the fleet} **then**
        $L'_d \leftarrow L^{\sharp}$
      else
        $L'_d \leftarrow R_d$
      end if
      if $d \in D^{\sharp}$ **and** $l \in L'_d$ **then**
        $t'_{d,l} \leftarrow 3 \times f_l$
        $a_{d,l} \leftarrow 0$
      else
        $t'_{d,l} \leftarrow t_{d,l}$
      end if
    end for
  end for
end procedure

procedure MsgExchanging(δ)
  for all $d \in D - D^{\downarrow}$ **do**
    $D^{\oplus} \leftarrow \{c \mid c \in D - \{d\} \wedge distance(d, c) < \delta\}$
    for all $c \in D^{\oplus}$ **do**
      for all $l \in L$ **do**
        if $a_{c,l} < a_{d,l}$ **then**
          $t_{d,l} \leftarrow \gamma(a_{c,l}) t'_{c,l} + [1 - \gamma(a_{c,l})] t_{d,l}$
          $a_{d,l} \leftarrow a_{c,l}$
          ▷ γ and update rule from Eq. 1 and 2.
        end if
      end for
    end for
  end for
end procedure

In our experiments, we simulate a commuting scenario. Each iteration simulates a fixed period of a working day: the same drivers will travel from the same origins to the same destinations. The simulation consists of η iterations.

At each iteration, before launching the main driver set (D) whose data is measured in the experiment, we generate load in the road network with an

auxiliary set of vehicles that do not perform IVC. This set is generated in order to ensure a constant load at any moment of the simulation. This load needs to be adjusted according to the road network that the experiment is being run. A small load is not interesting as it will not generate congestions to be avoided by the drivers. On the other hand, if the load is too high, it can lead to congestions in a high portion of the road network, making it difficult for the drivers to construct attractive routes to avoid them.

The simulation procedure is as follows: during an iteration, at each simulation timestep, the vehicles are moved according to the rules of the underlying traffic model; information age increases by one unit and route recalculation conditions are tested. The iteration finishes when all drivers arrive to their destinations.

Drivers' knowledge base is preserved between iterations so that they calculate new routes using the knowledge they acquired in previous trips.

The simulation procedure is formalized in Alg. 3.

Algorithm 3 Experiment

```
procedure SIMULATION(η, Ω, δ, τ, β)
    InitializeDrivers(Ω)                          ▷ Procedure from Alg. 1
    i ← 0
    while i < η do
        {pre-load network with auxiliary vehicles}
        CalculateRoutes()                         ▷ Procedure from Alg. 1
        D↓ ← ∅
        s ← 0                                     ▷ Starts the timesteps counter
        repeat
            moveVehicles()                        ▷ Apply the movement rules
            for all d ∈ D − D↓ do
                a_{d,j} ← a_{d,j} + 1 ∀ j ∈ L
                if {driver d left link l in this timestep} then
                    t_{d,l} ← {travel time spent on l}
                    a_{d,l} ← 0
                end if
                MsgConstruction()                 ▷ Proc. from Alg. 2
                MsgExchanging(δ)                  ▷ Proc. from Alg. 2
                if {driver d arrived at its destination} then
                    D↓ ← D↓ ∪ {d}
                end if
            end for
            s ← s + 1                             ▷ Increase timesteps counter
            RecalculateRoutes(τ)                  ▷ Procedure from Alg. 1
        until D − D↓ = ∅
        i ← i + 1                                 ▷ Increase iterations counter
    end while
end procedure
```

4 Scenario, results and discussion

4.1 Road network

The road network studied in this work is a 6x6 grid consisting of 36 nodes connected by 60 one-way links as shown in Fig. 1. Every link has one lane with the length of 300m, except for links from B3 to E3 and E3 to E5 (thicker lines on Fig. 1) which have three lanes. With this feature, this abstract scenario becomes more realistic, as roads capacities are not homogeneous. Drivers can turn to two directions in each intersection, except in the corners of the road network, in which there is only one direction to turn. The free-flow speed is 13.89 m/s (50 km/h) for any link[2] $l \in L$, resulting in a f_l of 21.60 seconds.

In this scenario, the probability distribution over the origins and destinations (Ω) is uniform, that is, every link has equal chance of being selected as origin or destination for any driver.

Bigger scenarios were already studied in previous works, but, as our traffic and communication models are more detailed, we study this abstract scenario. Neverthless, this is a complex scenario from the point of view of route choice, as the number of possible routes between two locations is relatively high.

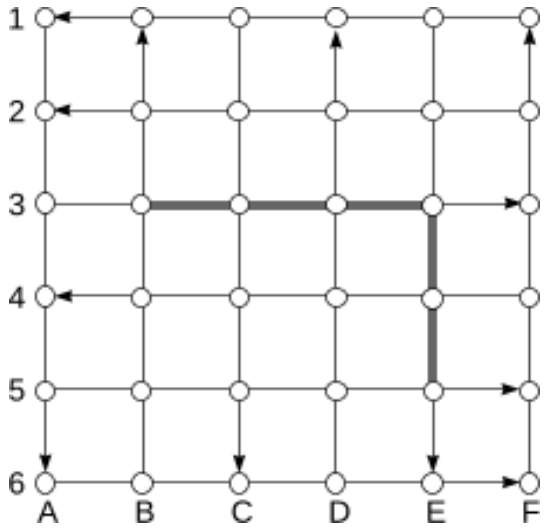


Fig. 1. 6x6 grid scenario. Arrows show the streets' directions. Thicker lines are the three-lane links.

In our experiments, we keep 700 vehicles in the road network as the auxiliary load discussed in Section 3.5. The number of drivers in the main set is $|D| =$

[2] This is the defined free-flow speed for a principal arterial urban road. This kind of road connects important activity centers and major traffic generators, according to the classification in [7].

200. Prior experimentation has shown that this load is reasonable in the 6x6 grid road network, i.e., the number of drivers is not too low in a way that it would not generate congestions to be avoided nor it is too high in a way that attractive routes to avoid congestions would not exist. To ensure comparability, the set of auxiliary vehicles is the same between iterations and between different experiments. In this configuration (200 main drivers and 700 auxiliary drivers), the fraction of vehicles equipped with IVC devices is $200/900 = 22,22\%$.

4.2 Experimental results

In our experiments, the performance of the drivers is measured by their average speed (total journey length / total travel time). Instead of measuring travel time, as normally found in other works, we measure drivers' average speed because it is not affected by different route lengths. Malicious agents are successful when they achieve higher average speeds than other drivers. This means that other drivers are avoiding links in the routes of the malicious agents either by performing en-route replanning or by calculating different routes at departure.

Regarding the values of the parameters, the communication radius (δ) is set to 200m. Prior experimentation has shown that this radius ensures a proper information dissemination on the studied scenario. The decay adjustment factor of information relevance given its age (β from Eq. 1), is set to 50. With this value, the function γ approaches zero and drivers disregard the received information when its age approaches 250 timesteps. For lower values of β, the information is disregarded too quickly and, for higher values, drivers consider it with a high weight even when the state of the link it refers has changed. The delay tolerance factor (τ) is set to 1.3. Prior experimentation has shown that for lower values, drivers perform en-route replanning too many times and, for higher values, the route recalculation procedure is seldom triggered. In our experiments, all drivers except the malicious can perform en-route replanning, i.e., $D^{\circ} = D - D^{\sharp}$.

The experiments consists in $\eta = 10$ iterations. For each iteration, the average speed for all drivers of a given type is plotted.

Figure 2 shows the performance of the cheating agent group both when they act individually and as the networked fleet. We plot the average speed for both cases with $|D^{\sharp}| = 4$ and $|D^{\sharp}| = 20$.

The performance of malicious agents with $|D^{\sharp}| = 4$ is more stable when they act as the networked fleet. This means that the fleet is more efficient at keeping other drivers away from their routes. Individually, malicious drivers have a drop on their performance on iterations 9 and 10, meaning that other drivers eventually return to the routes used by the malicious agents. For $|D^{\sharp}| = 20$, the performance of the networked fleet and the individual malicious agents is similar. For the individual malicious drivers, this happens because, when a malicious agent succeeds at diverting a regular driver from its route, this driver uses the route of a second malicious agent. In a next iteration, a regular driver that diverted from the route of the second malicious agent may use the route of another malicious agent. In this case, on average, the malicious agents do not

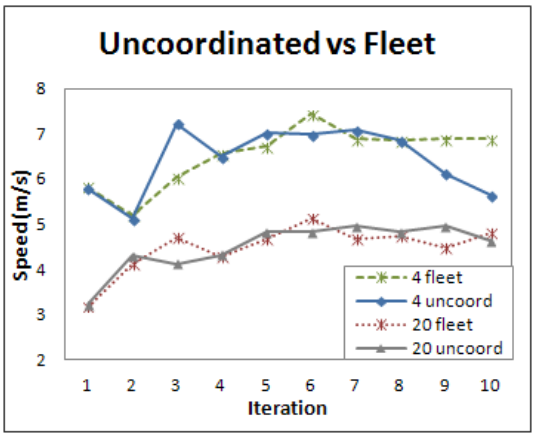


Fig. 2. Comparison of malicious drivers behavior

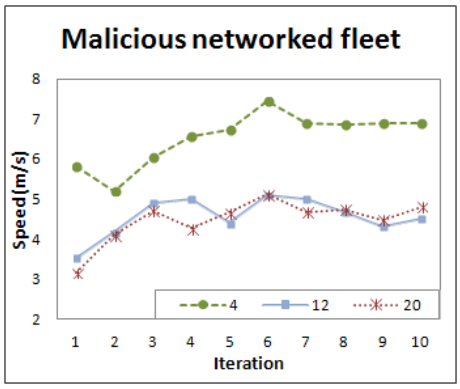
benefit from cheating. Figure 3 shows that for $|D^\sharp| = 20$, the performance of malicious drivers is similar to the performance of the other drivers.

The explanation for the performance of the networked fleet is that, as Table 1 shows, the routes of the networked fleet cover almost all links in the road network. In this situation, the other agents do not find attractive alternative routes, as they believe that most roads are congested, thus they use links on the routes of the networked fleet.

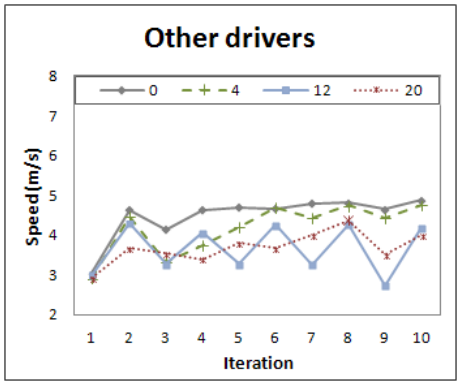
Figure 3 shows the impact of the networked fleet size on its performance as well as on the other drivers'. For the other drivers, we plotted the baseline, which corresponds to their performance when the fleet size is zero, i.e., the road network has only non-cheating agents.

The performance of other drivers approaches the baseline in the last 5 iterations when the networked fleet has 4 drivers. This happens because other drivers adapt themselves to the existence of the fleet: they find attractive alternative routes to traverse. The performance of other drivers oscillates when the networked fleet has 12 drivers. This happens because the other drivers choose a fast route in the beginning, but then they receive false information from the drivers of the fleet, thus choosing a different, more congested route in the next iteration.

From the networked fleet's point of view, it achieves its best performance with 4 drivers. This performance is better than the baseline. This shows that it is advantageous for the agents to cheat cooperatively in this configuration. However, as the number of drivers on the networked fleet increases, the fleet drivers do not benefit from cheating. This happens because, as discussed earlier, Table 1 shows that the number of links used by the networked fleet drivers is high, leaving few alternatives for the other drivers to use. As false information is spread for the majority of links on the road network, the other drivers believe that the costs for traversing the links are evenly high.



(a)



(b)

Fig. 3. Performance of malicious networked fleet and other drivers according to fleet size. Legends indicate the fleet size, zero is the baseline.

The results of our experiments showed that, for the studied scenario, the behavior of the malicious agent group, that is, whether they are networked or not, makes little difference on their performance. In contrast, the number of agents in the malicious group has a significant impact on the performance of both malicious and regular agents. When the malicious group has few agents, they all benefit from cheating while the regular drivers adapt themselves and find attractive alternative routes, minimizing the drop on their performance. On the other hand, when the size of the group of malicious agents approaches 10% of all agents with IVC (which corresponds to 20 out of 200 agents in our studied scenario), the performance of all drivers is negatively affected.

$\lvert D^\sharp \rvert$	% of links used by $d \in D^\sharp$
4	40%
12	81%
20	96%

Table 1. Fraction of links used by the drivers in $D^\sharp$.

5 Conclusions and future work

In this work we modeled drivers as intelligent agents and assessed the effect of cheating agents both when they act individually and when they cooperate with each another, forming a network of cheating drivers. Also, we adopted a traffic simulation model with high level of detail.

In the approach presented in this work, the networked malicious fleet spreads false information over almost all the links of the road network when its size increases. By doing this, the malicious fleet does not benefit from cheating. In order to be successful in this situation, the malicious fleet must devise a way to control only a portion of the road network. This can be a topic for further investigation.

In our experiments, as we load the road network with auxiliary vehicles, only a small fraction of all the drivers on the road network is equipped with IVC devices. This could correspond to an initial phase of implementation of IVC technology in the real world. In this setting, a small group can benefit for cheating cooperatively. But, as this group increases, the performance of the system as a whole drops. For this reason, the study of countermeasures to prevent the spread of false information should be taken in account, especially for the initial phase of IVC implementation. An example of counter measure could be the history-based scheme proposed in [4].

Future work could also address issues such as the effect of information not only on the route calculation but on departure time of drivers. Also, the combination of learning, or history-based knowledge with IVC could be explored as well as the existence of competitive fleets that intend to take control of portions of the road network by the exploitation of IVC protocols.

6 Acknowledgments

Both authors are partially supported by CNPq, FAPERGS and CAPES.

References

1. M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - simulation of urban mobility: An overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 63–68, Barcelona, Spain, October 2011.

2. K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In N. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 530–537, New York, USA, July 2004. New York, IEEE Computer Society.
3. S. M. Galib and I. Moser. Road traffic optimisation using an evolutionary game. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, GECCO '11, pages 519–526, New York, NY, USA, 2011. ACM.
4. S. Kraus, R. Parshani, and Y. Shavitt. A study on gossiping in transportation networks. *Vehicular Technology, IEEE Transactions on*, 57(4):2602 –2607, july 2008.
5. L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.
6. D. Malkhi, E. Pavlov, and Y. Sella. Gossip with malicious parties. Technical report, 2003.
7. National Research Council. *Highway capacity manual*. Transportation Research Board, 2000.
8. Z. Rui, T. Fu, D. Lai, and Y. Jiang. Cooperation among malicious agents: a general quantitative congestion game framework. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '12, pages 1331–1332, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.
9. M. Vasirani and S. Ossowski. A market-based approach to reservation-based urban road traffic management. In K. Decker, J. Sichman, C. Sierra, and C. Castelfranchi, editors, *Proc. of the 8th Int. J. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 617–624, Budapest, May 2009. IFAAMAS.
10. A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J. Hubaux. TraCI: an interface for coupling road traffic and network simulators. In *11th communications and networking simulation symposium*, pages 155–163. ACM, 2008.
11. B. Yu and M. Singh. A social mechanism of reputation management in electronic communities. In M. Klusch and L. Kerschberg, editors, *Cooperative Information Agents IV - The Future of Information Agents in Cyberspace*, volume 1860 of *Lecture Notes in Computer Science*, pages 355–393. Springer Berlin / Heidelberg, 2000.