Multiagent system architecture and method for group-oriented traffic coordination

Jana Görmer and Jörg P. Müller
Department of Informatics, TU Clausthal, Germany
Email: {Jana.Goermer|Joerg.Mueller}@tu-clausthal.de

***Abstract*—Next-generation traffic management systems will make use of on-board intelligence and communication capabilities of vehicles and traffic infrastructure. In this paper, we investigate a multiagent approach allowing vehicle agents to form groups in order to co-ordinate their speed and lane choices. Our hypothesis is that a decentralized approach based on a co-operative driving method can contribute to higher and smoother traffic flow, leading to higher speeds and less delays. Our focus is on automated vehicle decision models. We develop a group-oriented driving method with vehicle agents that perceive their environment and exchange information. The paper proposes decentralized dynamic vehicle grouping algorithm, a conflict detection and global coordination method, and defines individual driving strategies for vehicles. For validation, we compare our method with a driving method implemented in the commercial traffic simulation platform AIMSUN. Experimental results indicate that group formation and group coordination methods can improveme traffic network throughput.**

MAS coordination, MAS cooperation, traffic vehicle grouping (key words)

I. Introduction

Next-generation traffic systems will feature onboard intelligence and Car-to-X communication capability. This implies new control challenges and a shift from traditional hierarchical organization to a multiagent systems organization. But it also opens ample of new opportunities in terms of ad-hoc coordination and co-operation of vehicles and infrastructure components in order to maximize throughput and avoid breakdowns. Our conception is that in future cyber-physical traffic and transport ecosystems, vehicles and infrastructure components will be equipped with software agents that make decisions autonomously[1] on behalf of traffic participants and control authorities. In this work, we develop an agent-based model and corresponding distributed algorithms that represent different types of traffic participants with local goals, differing capacities (speed, acceleration, driving skills), and preferences (speed or lane). The driving behavior of a vehicle (e.g. acceleration, deceleration, or lane changing) is a function of its parameters and of the traffic regulations. However, the ability of a traffic participant to act according to its preferences is restricted by other participants: Often, fast vehicles are forced to drive slow because overtaking or lane changing is not possible due to slower vehicles (speed conflict problem). Consequently, traffic congestion occurs and the travel times increase.

In this work, we approach the problem illustrated in the scenario by proposing *Group-oriented driving* (GoD), a new autonomous co-operative driving method. In GoD autonomous vehicles are able to perceive their environment, communicate, form groups, and co-ordinate their behaviors in order to avoid and solve the conflict situations described above. We compare our method with a (non-cooperative) reference driving method implemented within the commercial traffic simulation platform AIMSUN. We give simulation experiments that suggest that the method can reduce travel times and delays, while the overall benefit of the approach depends on the structure of the overall vehicle population.

The paper is organized as follows: After discussing related work in Section II, we give an overview of the GoD method and the architecture of the multiagent system (MAS) used in this work in Section III. Section IV describes the cooperation protocol between autonomous vehicle agents and presents the driving strategy of individual autonomous vehicle in the context of the cooperative method. In Section V a case study is used to evaluate the performance of GoD in terms of speed and travel time.

II. Related work

Methods of coordination and cooperation in MAS are considered e.g. in [1]–[3]. Barrett et al. [4] examine models for ad-hoc agent teamwork for an abstract simplified domain without considering communication and sensory aspects. There are various multiagent grouping technologies applying general coordination and cooperation processes, e.g. [5], [6], but no solutions tailored for the traffic domain. Therefore, we provide a method for modeling, simulating, and analyzing various traffic scenarios with autonomous vehicles at microscopic level. On the run-time side, we use AIMSUN [7] as a traffic simulation system to model and simulate traffic scenarios, and integrate it with the JADE framework [8] for agent-based implementation. The resulting platform is "**A**gent-based **T**raffic **Sim**ulation System (ATSim)" [9]. We realize a solution to support global system throughput on a macro-level view while preserving decentralization and openness on the individual vehicle micro-level. Hence, the infrastructure elements of the traffic domain like traffic lights and/or vehicles

This work was funded by the NTH Focused School for IT Ecosystems (www.it-ecosystems.org).

[1]We acknowledge the role of the human-in-the-loop, but it is not the focus of this work.

978-1-4673-1703-0/12/$31.00 ©2013 IEEE

can use the agent paradigm. Research on multi-agent traffic light coordination and cooperation is done by Bazzan [10], whereas routing has been addressed by [11]. However, there is a lack of work on decentralized vehicle agent coordination and cooperation, which we consider in our work.

Today's traffic simulation systems are widely used to analyze and manage traffic flows. Most research on traffic simulation systems [7], [12], [13] simulates driver behavior using mathematical models, e.g., for car following and lane changing. However, using such models, it is not possible to take the effect of communication and cooperation into account; also, these models do not cover information, goal, and plan states of *autonomous vehicles*. In order to make vehicles of a traffic simulation system communicable and to support flexible autonomous behavior, we introduce a group-oriented driving method, as well spontaneous team formation and conflict solutions with autonomous agents in MAS controlled by a developed traffic simulation system.

III. GoD METHOD AND MULTIAGENT SYSTEM ARCHITECTURE

The objective of GoD is to coordinate autonomous vehicles such that fast vehicles are not blocked by slower vehicles. The method makes use of the communication ability of autonomous vehicles for coordinating them in a decentralized fashion. Because GoD works decentrally on vehicles, there is no need for modifying existing traffic infrastructures. The method is based on a vehicle grouping concept. State of the art traffic telematic solutions only allow a vehicle to communicate with others, which are in a limited area (Range of Perception (RoP)). Via grouping, vehicles can extend their RoP by multi-hop communication. Vehicles are coordinated at group level. GoD allows vehicles e.g. to form groups that platoon based on desired speed. In our approach, each group has a leader. In case fast groups are blocked by other slow groups, the group leaders will communicate with each other to arrange lanes (called group lanes). Group lanes are known to all its members. Driving on group lanes, fast vehicles can avoid being blocked by other slow vehicles and vice versa. In the following the main elements 1) Decentralized dynamic vehicle grouping, 2) Conflict detection, 3) Collaborative gap solution and 4) Driving strategy of individual vehicles of GoD are described in Section IV.

Multiagent system architecture: The usage of traffic simulation systems (TSS) [7], [12], [14], [15] for simulating traffic scenarios are widely applied in the traffic management domain. A TSS provides an easy way to model and configure various traffic scenarios. However, vehicles simulated by such systems use hard-coded mathematical models for simulating behaviors of traffic participants. Thus, they can neither communicate with others nor choose driving actions autonomously. Since communication and autonomy capacities of vehicles are the most important requirements of GoD, we employ an agent-based traffic system called ATSim as a test-bed for GoD [9].

As shown in Figure 1 ATSim is a composition of five main components: AIMSUN Simulation Model, MAS Connector, MAS Services, Agent Controller, and Agent Container.

1) **AIMSUN Simulation Model:** contains a traffic model with road networks, traffic lights and a traffic simulator. It executes simulation processes and communicates with external applications via provided API.
2) **MAS services:** is an interface for AIMSUN to communicate with agents of the multiagent system. It provides services for creating and controlling agent life cycles for controlling traffic objects like vehicles and traffic lights.
3) **Multiagent system connector:** is necessary since vehicles in Aimsun are created dynamically and controlled by agents of a MAS. Agents need information of vehicles and simulated model (e.g. highway, traffic light) for reasoning their actions. MAS connector is used for exchanging information between AIMSUN and MAS.
4) **Agent container:** provides the environment for managing agents. We use Jade and agents which exist in agent containers can communicate.

IV. COOPERATION TECHNIQUES FOR GROUPS AND CONFLICTS

Communication between vehicles requires specification of message protocols. We use the following simple message format for all messages exchanged between vehicles:

$$\texttt{msg(id,}id_s\texttt{,}id_r\texttt{,content)}$$

where id is identifier number of message. id_s and id_r are identifier number of sender and receiver. Content of message contains information exchanged between vehicles.

A. Decentralized vehicle grouping

The first step of GoDis group formation. For centralized vehicle grouping extra devices like central coordinators need to be part of the traffic network. Due to limitations of wireless communication and computation, a coordinator can not communicate with all vehicles of a traffic network. This approach is expensive when applying subcoordinators to a large traffic network. We consider a decentralized approach, in which group formation is accomplished via decisions and interaction between individual autonomous vehicles (assuming that the number of neighbors is known and there is no message loss). We define a vehicle group G_x with leader $x(id_x, a_x, d_x, ds_x)$ as set of vehicles y with following property:

$$\forall y \in G_x, f(x,y) \leq \alpha_x$$

where α_x is a fixed value predefined by x. The attributes id_x, a_x, d_x, ds_x are identifier number, maximal acceleration rate, maximal deceleration rate, and desired speed of x. Function $f(x,y)$ calculates the dissimilarity between vehicles x and y in terms of their maximal deceleration rate, maximal acceleration rate, and desired speed. Thus, a group leader x will accept any vehicle y as its member if the dissimilarity $f(x,y)$ is smaller than or equals α_x. Using static attributes as inputs for the dissimilarity function $f(x,y)$, a group leader x can decide if a vehicle y can participate in its group or not. Since there is no centralized device for

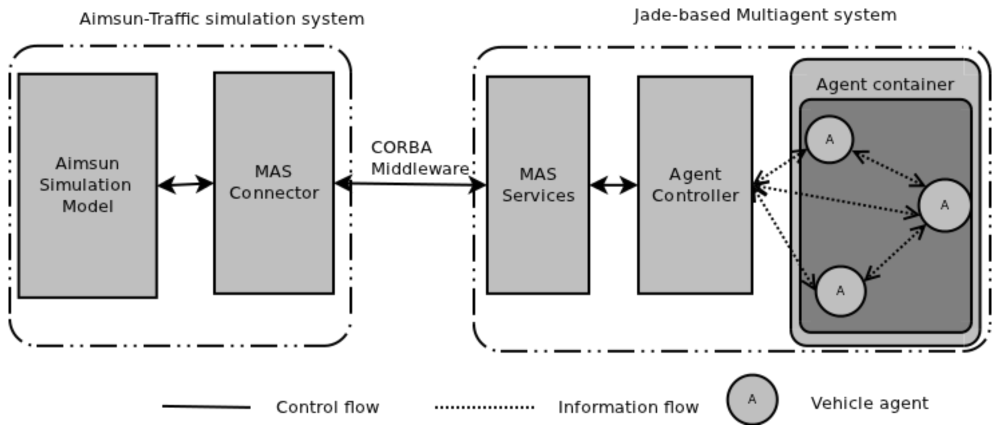

Fig. 1. ATSim system architecture

grouping vehicles, an autonomous vehicle x must decide. We assume that for each attribute a_x, d_x, ds_x of x there is a maximal acceptable difference $w_{a,x}, w_{d,x}, w_{ds,x}$. The maximal acceptable differences denote that x is willing to group with vehicles y, whose attributes a_y, d_y, ds_y are in the respective areas $[a_x - w_{a,x}, a_x - w_{a,x}]$, $[d_x - w_{d,x}, d_x - w_{d,x}]$, and $[ds_x - w_{ds,x}, ds_x - w_{ds,x}]$.

Consider vehicle $x(a_x, d_x, ds_x)$ and $y(a_y, d_y, ds_y)$ as two points in a three-dimensional space. The dissimilarity between x and y is the measured distance between them. We employed the Manhattan distance for calculating dissimilarities between vehicles. Thus, the dissimilarity between x and y is measured as follows:

$$f(x,y) = \alpha_1 \frac{|a_x - a_y|}{w_{a,x}} + \alpha_2 \frac{|d_x - d_y|}{w_{d,x}} + \alpha_3 \frac{|ds_x - ds_y|}{w_{ds,x}} \quad (1)$$

Where α_1, α_2, α_3 are weighting parameters used to denote the importance of dissimilarities of attributes. The choice of values of α_1, α_2, α_3 must satisfy the condition

$$\alpha_1 + \alpha_2 + \alpha_3 = \alpha_x \quad (2)$$

This work does not concentrate on developing methods for choosing optimal values of α_1, α_2, α_3, α_x. Values are given in Section V.

We propose a decentralized grouping algorithms described in Figure 2. Assume that a vehicle y is trying to find and participate in a group. As a first step y requests every neighbor vehicles x for group information by sending them messages msg(id, id_y, id_x, reqGinfor). After receiving all reply messages (msg(id, id_x, id_y, G_x)), y uses the dissimilarity function $f(y,x)$ to find a suitable group ($f(y,x) \leq \alpha y$). If a vehicle group G_x is found, y sends a request (msg(id, id_y, id_x, reqPar) to the leader x of G_x to ask for participation. Group leader x uses $f(x,y)$ to decide if y is allowed to join its group or not. This means, in case of $f(x,y) \leq \alpha_x$ (or $f(x,y) > \alpha_x$) x replies y with a positive (or negative) response message msg(id, id_x, id_y, yes|no)). In case of unsuccessful participation, y will consider to create a group on its own (step 4) and waits for the participation of other vehicles. However, the creation of a new group requires fulfillment of following conditions:

1) y is not member of any group.
2) y knows that there is at least one candidate vehicle z with $f(y,z) < \alpha_y$.
3) y has the greatest id in compare with id of other candidate vehicles.

Group creator takes the role of a group leader. The leader maintains number of members and decides when it wants to remove its group. It happens that a created group receives no participation request from candidate vehicles. Maintaining an empty group does not allow leader to participate to other groups. Thus, the leader should remove its empty group after some time (we use three simulation steps in the following experiments).

B. *Group conflicts and solution*

A conflict between vehicle groups is defined as a situation, in which a group of fast vehicles is driving behind a group of slow ones. It can be detected by leader or members. This means, all members x_a of a vehicle group G_x maintain information of its group. At each simulation step, x_a sends msg(id, id_{x_a}, id_z, reqGinfor) message to its neighbors z to find other groups. Receiving the neigbhors response, x_a uses the following binary decision function to determine conflict between his own G_x and neighbor group G_z.

$$conf(G_x, G_z) = \begin{cases} yes & \text{if } ds_x - ds_z > w_{ds,x} \wedge \\ & \exists z_n \in G_z, p_{z_n} < p_{x_a} \\ no & \text{otherwise} \end{cases} \quad (3)$$

where p_{x_a} and p_{z_n} are positions of x_a and z_n on the highway. Once a conflict is detected ($conf(G_x, G_z)$), x_a sends information of group G_z to its leader. A group leader coordinates its members via *group lanes*. Group lanes are only reserved for members of a vehicles group. The choice of group lanes bases on following two criteria:

1) The group lanes should require the least possible lane changing of members.

978-1-4673-1703-0/12/$31.00 ©2013 IEEE

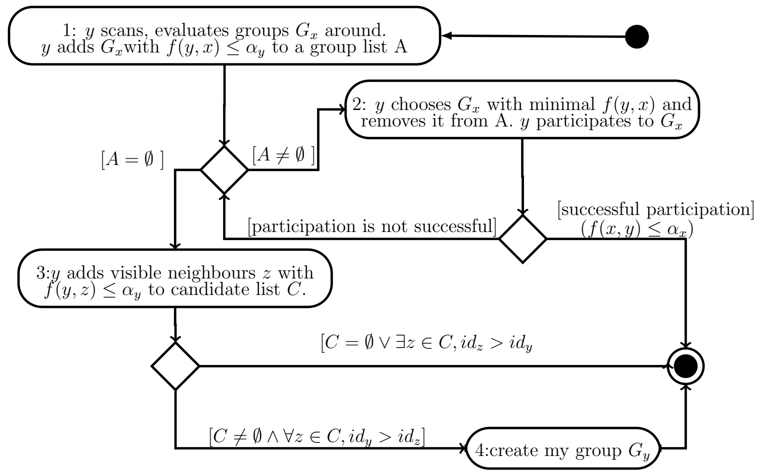

Fig. 2. Activity diagram

2) The group lanes should warranty that fast groups are not blocked by slow groups.

We use a dominant method to determine group lanes of conflict groups. Suppose h_n is lane n of highway h and G_x is in following conflicts: $conf(G_x, G_{z_1}), ..., conf(G_x, G_{z_n})$. Dominant value $v_{x,n}$ of G_x on lane h_n is calculated using the following function:

$$v_{x,n} = num_{x,n} - \sum_{i=z_1}^{z_n} num_{i,n} \quad (4)$$

where $num_{x,n}$ is the number of members of G_x on lane h_n. It was defined that the leader of a fast group is allowed to choose its group lane first. Calculating dominant values of its groups for all lanes, x chooses lanes with $v_{x,n} \geq 0$ as its group lanes. If all dominant values are negative, a lane with maximal dominant value is chosen. The group lanes of x are marked as *busy* and can not be chosen by other conflict (slower) groups $G_{z_1}, ..., G_{z_n}$. Leader vehicle x publishes its group lanes to members and leaders of conflict groups via message `msg(id,`id_x`,[`$id_{z_{1..n}}$`,`id_{x_a}`],infGlanes)`.

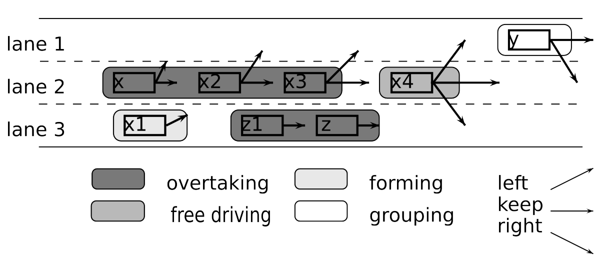

Fig. 3. Two vehicle groups $G_x = [x1, x2, x3, x4]$ and $G_z = [z, z1]$ are in conflict

Consider the example in Figure 3. Assume that a fast group vehicle $G_x = [x, x1, x2, x3, x4]$ and a slow $G_z = [z, z1]$ are in conflict $conf(G_x, G_z)$. Dominant values of G_x for lanes $1, 2, 3$ are respectively $0, 4, -1$. Thus x chooses lanes $1, 2$ as its group lanes.

C. Collaborative gap problem solution

As described in previous section, group lanes are used for coordinating members of a vehicle group to avoid conflict situation. Member vehicles not driving on the group lane may want to change their current lane to the group lane. In normal traffic, it is impossible for a driver to change lane, if the free gap on the future lane is too small (or does not exist). Therefore the driver must stay on its lane until he finds a gap, which is large enough for safe lane changing (gap problem). Using communication, an autonomous vehicle can work with others to create its own gap for lane changing. Consider the gap problem of the following example illustrated in Figure 4:

Assume that vehicle V_c wants to change to lane 1. Each vehicle V_a, V_b, V_c needs a distance to stop, which is denoted by the length of arrow in front of it (see Figure 4). Let $s(V_a, V_c)$ denote the gap between V_a and V_c. $s(V_c, V_b)$ is the gap between V_c and V_b. In normal traffic, V_a always drives at a secure speed, which allows it to stop behind V_b. This speed can be calculated using the car-following model of Gipps [16]. As shown in Figure 4 the stop position $p(V_c)$ of V_c is before the stop position $p(V_a)$. This means, if V_c changes to lane 1 and has to stop there, V_a would not be able to stop in time and collide with V_c.

For safe lane changing the following conditions must hold for V_c:

1) The rear gap $s(V_a, V_c)$ should be big enough to allow V_a to stop behind V_c.
2) The front gap $s(V_c, V_b)$ should be big enough to allow V_c to stop behind V_b.

D. Individual vehicle driving strategy

"When should a vehicle follow the coordination of leader and when not?". To answer this, we developed a state-based lane choosing strategy. A vehicle is considered always to be in one of four states `Grouping, Forming, Overtaking, Free Driving`. At each state vehicle is defined to choose lanes based on coordination or lane-utility.

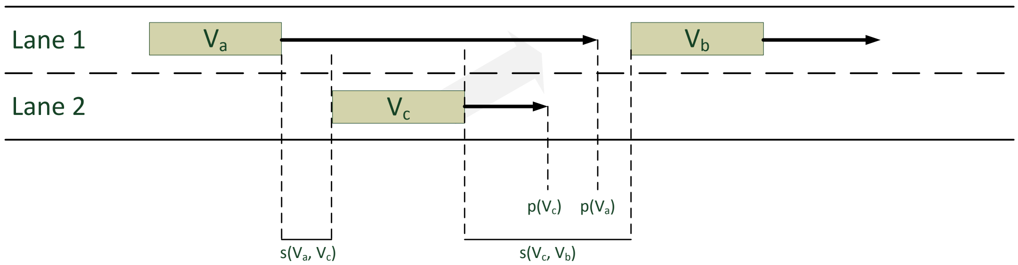

Fig. 4. V_c tries to switch lanes from V_a and V_b

- *Grouping:* This state denotes that a vehicle does not belong to any group. Being in this state the vehicle will try to drive on lane, which permits it to reach its desired speed (utility-based decision).
- *Forming:* This state a vehicles belongs to a group and is not driving on group lanes. The vehicle ignores its own utility and follows the coordination of leader. This means, the vehicle accepts to reduce its current speed to change to group lanes if necessary.
- *Overtaking:* This state denotes that a vehicle belongs to a group and is driving on one of group lanes. The vehicle uses utility-based decision only on its group lanes. This means, vehicle will try to change to another group lane if the lane maximizes its utility.
- *Free Driving:* A vehicle in this state is a group member and driving in front of all others. Being in this state the vehicle ignores the coordination of (or behinds) group leader and uses utility-based decision for choosing its future lane.

We used the following function for calculating utility-value of vehicle x on a lane n:

$$U_{x,n} = [V_{s,n}^{new} - V_{s,n}^{current}] + [V^n ew_{x,n} - V_{x,n}^{current}] \quad (5)$$

Where $V_{s,n}^{new}$ is potential speed of successor of x on lane n. $V_{s,n}^{current}$ is actual speed of successor of x. $V^n ew_{x,n}$ is the potential speed of x if it changes to lane n. $V_x^{current}$ is the current speed of x on its current lane. We employed the car-following model of Gipps [16] for calculating the speed V of vehicle. The utilities-based decision allows x to choose lane, on which x can accelerate its speed as much as possible and blocks its follower vehicle as little as possible.

The example in Figure 3 shows states of vehicles. Being member of G_x and driving behind member z of conflict group G_z, vehicles $x, x2, x3$ are in state `Overtaking`. Thus, $x, x2, x3$ can use its utility-based lane choice on group lanes $1, 2$ (see example in Section IV-B). Being in state `Forming` $x1$ obeys the coordination of x. It tries to change to group lanes 2 to avoid slow vehicles $z1, z$ of conflict group G_z. Vehicle $x4$ has overtaken all members $z, z1$ of G_z. Thus it is in state `free Driving` and must not follow the coordination of leader x. It uses utility-based lane choice strategy to find future lane. Vehicle y does not participate to any group and is at state `Grouping`. Therefore the choice is between staying on its own lane or changing to lane 2 based on its utility.

V. Experimental Validation

This section presents a case study to validate the scalability and performance of ATSim by means of simulation experiments. The results of speed and delays of each test show many advantages in the simulation of the GoD method compared to a simulation with standard traffic parameters.

Our simulation scenario describes three different types of vehicles: (1) fast, (2) medium and (3) cautious. The properties of the different class types are allegorized through desired speed, maximum delay, and acceleration. This test is generated twice with AIMSUN. First, data is collected with a standard driving setting of AIMSUN before ATSim is used in a second run to control the group-oriented driving method. The three classes of vehicles have the following properties in all simulations:

- fast: desired speed $160km/h$, maximum delay $-8m/s^2$, acceleration $6m/s^2$;
- medium: desired speed $80km/h$, maximum delay $-8m/s^2$, acceleration $4m/s^2$;
- cautious: desired speed $60km/h$, maximum delay $-6m/s^2$, acceleration $2m/s^2$.

With GoD fast vehicles can go with their desired speeds and improve delays and acceleration times reaching their goals without blocking others. Therefore the delay of all is reduced seen in Figure 5. Here, (a) depicts the simulation results for manned vehicles, GoD was *not* applied during the simulation, while (b) shows the results for *autonomous* vehicles. In (a), the delays of the medium class are distributed in the range from 0 to 23 sec/km. The average delay of a medium fast class varies at 3 sec/km (green horizontal line). Hence, the delay of medium manned deviates up to 20 sec/km. In comparison, when GoD is applied, the average delay is approximately 1 sec/km lower. Remark that the maximal deviation of average delay decreases to only 5 seconds during the simulation. Looking at the fast class, we see similar results, they have better travel times in the autonomous simulation. For the cautious class results are almost identical which is not surprising as cautious vehicles are the slowest and rather block other traffic participants than being blocked very often themselves.

In real traffic, vehicles in a group have small differences in their properties. Parts (c) and (d) of Figure 5 show simulation results with slight variations which allow vehicles in one group to have a maximum difference in the desired speed of -10 km/h, maximum delay and acceleration by $-0.5m/s^2$.

978-1-4673-1703-0/12/$31.00 ©2013 IEEE

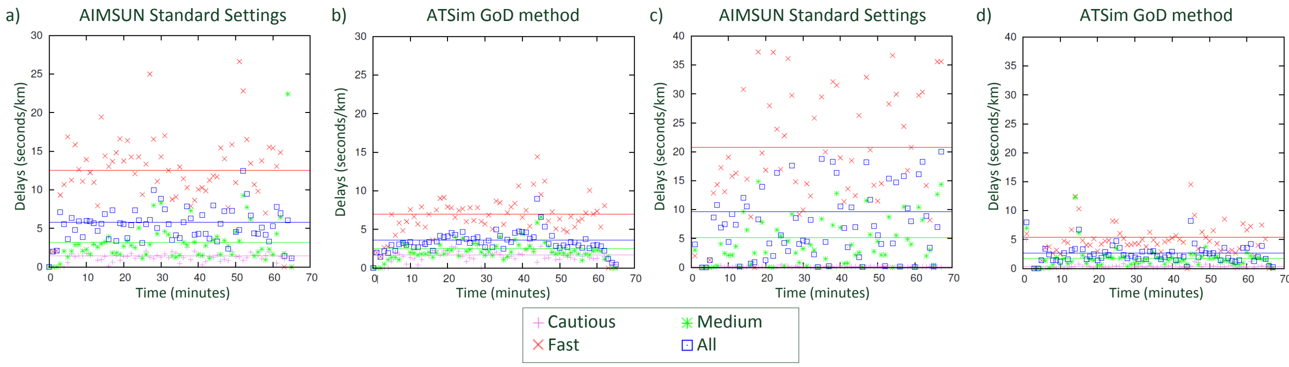

Fig. 5. Simulation results

Property differences have negative influence on the speed of autonomous vehicles in one group. As a consequence, it is important to form groups in such a way that property differences inside its group members is harmonized. Both tests illustrate big differences *between* vehicle groups (by means of desired speed, delay, and acceleration) have in total better results with the GoD method.

ATSim is relatively slow in simulating the traffic model for larger numbers of vehicles. For instance, up to 1.2 seconds and 160 megabyte memory are required to simulate one simulation step in the case of 2000 vehicles. Thus, at the current stage, ATSim is suitable to simulate small to medium traffic models with limited number of agents (simulation at microscopic level). The good news is that both time and memory grow approximately linear with the number of agents, which may induce that the size of the simulation can be increased by a linear factor by using more powerful hardware or distributed computing resources. This preliminary hypothesis needs to be confirmed by further experiments.

VI. Discussion and Future Work

This work considered a specific traffic scenario. Deeper analysis is required to answer whether the GoD method provides advantages compared to the non-cooperative driving method for situations when vehicles drive very close to each other in dense traffic. Yet, first results in evaluating GoD have been promising. Future work will examine and compare different group coordination architectures and strategies for determining group life time, and strategies of initiating and abandoning groups. We plan to investigate richer vehicle models including strategy learning, and consider interaction of agents and infrastructure components (such as traffic lights). We reported first results that study local and group decision-making strategies based on agent-oriented data-mining and reinforcement learning in [17]. Additionally, we considered different traffic management scenarios including optimizing throughput at individual and subsequent intersections, or co-operative group-based routing of platoons. A further venue for our research are communication strategies (with whom to exchange what information).

References

[1] A. Consoli, J. Tweedale, and L. Jain, "An architecture for agent coordination and cooperation," in *Proc. of 11th Int. Conf. on Knowledge-Based Intelligent Engineering Systems*. Springer, 2007, pp. 934–940.
[2] N. Findler and G. Elder, "Multi-agent coordination and cooperation in a distributed dynamic environment with limited resources," pp. 229–238, 1995.
[3] N. Salazar, J. A. Rodriguez-Aguilar, J. L. Arcos, A. Peleteiro, and J. C. Burguillo-Rial, "Emerging cooperation on complex networks," in *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems*, 2011, pp. 669–676.
[4] S. Barrett, P. Stone, and S. Kraus, "Empirical evaluation of ad hoc teamwork in the pursuit domain," in *In Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems*, 2011, pp. 567–574.
[5] P. R. Cohen and H. J. Levesque, "Teamwork," Noûs, Artificial Intelligence Center, SRI International, Menlo Park, CA, Technical Note 504, 1991.
[6] J. E. Doran, S. Franklin, N. R. Jennings, and T. J. Norman, "On cooperation in multi-agent systems," 1997.
[7] J. Barceló, E. Codina, J. Casas, J. L. Ferrer, and D. García, "Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems," *Journal of Intelligent and Robotic Systems*, vol. 41, pp. 173–203, 2005.
[8] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE: A FIPA-compliant agent framework," in *Proceedings of PAAM*, London, UK, 1999, pp. 97–108.
[9] V. H. Chu, J. Görmer, and J. P. Müller, "ATSim: Combining AIMSUM and jade for agent-based traffic simulation," in *14th Conf. of the Spanish Association for Artificial Intelligence*, 2011.
[10] A. L. C. Bazzan, D. de Oliveira, and B. C. da Silva, "Learning in groups of traffic signals," *Eng. Appl. Artif. Intell.*, vol. 23, pp. 560–568, 2010.
[11] H. Prothmann, J. Branke, H. Schmeck, S. Tomforde, F. Rochner, J. Hähner, and C. Müller-Schloer, "Organic traffic light control for urban road networks," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 2, no. 3, pp. 203–225, 2009.
[12] G. D. B. Cameron and G. I. D. Duncan, "Paramics: Parallel microscopic simulation of road traffic," *The Journal of Supercomputing*, vol. 10, pp. 25–53, 1996.
[13] *VISSIM 4.10 User Manual*, PTV AG, Karlsruhe, Germany, 2005.
[14] S. Jones, A. Sullivan, N. Cheekoti, M. Anderson, and D. Malave, "Traffic simulation software comparison study," *UTCA Report*, vol. 2217, 2004.
[15] A. Lansdowne, "Traffic simulation using agent-based modelling," *Science*, 2006.
[16] P. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, pp. 105–111, 1981.
[17] M. Fiosins, J. Fiosina, J. P. Müller, and J. Görmer, "Decision making for autonomous vehicles in urban traffic," in *9th Int. Conf. on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2011, pp. 173–178.