# Collaborative Decision Making for Lane-Free Autonomous Driving in the Presence of Uncertainty *

Pavlos Geronymakis[0000-0001-7697-1661], Dimitrios Troullinos[0000-0003-3228-3888], Georgios Chalkiadakis[0000-0002-0716-2972], and Markos Papageorgiou[0000-0001-5821-4982]

Technical University of Crete, Chania 73100, Greece
pgeronymakis@isc.tuc.gr, dtroullinos@dssl.tuc.gr, gehalk@intelligence.tuc.gr, markos@dssl.tuc.gr

**Abstract.** The recently introduced *lane-free traffic* paradigm removes the restrictions of the traffic lanes, so that autonomous vehicles can move anywhere laterally across the road's width. Previous research in this domain has employed the celebrated max-plus message-passing algorithm in order to allow the coordination of all (connected and autonomous) vehicles in the environment. However, when allowing for the realistic perspective that there exist vehicles that are unable or unwilling to communicate with others, the uncertainty introduced renders the aforementioned coordination approach ineffective. To combat this, in this paper we adjust the Max-plus algorithm accordingly so that agents using max-plus for coordination can also observe and take into consideration independent agents via emulated messages. We put forward different methods to form these messages—namely the Maximax, Maximin, Hurwicz, Minimax Regret and Laplace decision-making criteria. Finally, we provide a thorough evaluation of our approach, including a detailed comparison of all criteria used for message-forming.

**Keywords:** Max-plus algorithm · Uncertainty · Lane-Free Traffic.

## 1 Introduction

In recent years, there have been significant advancements in the field of automobiles and the automation of vehicular traffic. While research in this field mainly focuses on lane-based traffic, a recent development is the investigation of the novel lane-free traffic paradigm [11, 10].

In our work, we also consider agents operating in a lane-free environment, specifically on a lane-free one-way highway. As such, vehicles are not restricted by the lanes as in traditional highways, but can instead move freely across the

* The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 Research and Innovation programme/ ERC Grant Agreement n. [833915], project TrafficFluid.

entire highway width. Connected and Autonomous Vehicles (CAVs) enter the highway at random positions, with randomly assigned desired speeds. In their attempt to reach their desired speed and exit the highway with minimum delay, they may accelerate and move past other agents, and this kind of maneuvers may result into collisions among them.

Now, in existing work, the max-plus algorithm [7, 14] is used to coordinate the movement of the CAV agents, and assist them in reaching their desired speeds while avoiding collisions. Note that this line of work has only focused on homogeneous environments, where every agent in the highway decides upon its actions using the max-plus algorithm. By contrast, we introduce additional agents whose movement is independent of the max-plus algorithm, and modify max-plus in order to incorporate them within the algorithm. However, due to the lack of communication, this imposes uncertainty for max-plus agents. To this end, we adopt a range of different decision-making criteria to be embedded in our adjusted version of max-plus, so as to incorporate uncertainty within the algorithm. The incorporated criteria include: *Maximax, Maximin, Hurwicz, Minimax Regret, Laplace*; and also a simple opponent modelling technique we devised for our domain. Our experimental evaluation shows that the embedding of decision-making criteria in the face of uncertainty within max-plus, does in fact reduce collision occurrences; and that the more elaborate criteria provide incremental improvements.

In what follows, in Section 2 we provide the relevant background work, that will be used as our foundation to address the issues of uncertainty in the lane-free environment, while in Section 3 we present our approach involving the adjustment of max-plus algorithm and the incorporation of multiple criteria that address the uncertainty imposed by individual agents. In Section 4 we present our experimental evaluation and discuss the effectiveness of our approach by comparing each criterion in terms of reducing collisions among max-plus-coordinated and independent agents. Finally, in Section 5 we conclude our work and address potential future endeavors.

## 2 Background and Related Work

In this section, we present the technical background of this work, namely the framework of Coordination Graphs and the max-plus Algorithm, along with related work, with more focus towards the existing work that we build upon.

### 2.1 Coordination Graphs

Coordination Graphs (CGs) [4] are used in multi-agent systems to model coordination among agents. In a multi-agent environment, there is not always a need for explicit coordination among all agents. Local coordination between agents that interact with each other is often enough to achieve the global coordination task. CGs take advantage of this, allowing for scalability in the number of participating agents, and making the joint action of a set of agents that maximizes the global utility more easily obtainable.

In CGs, the agents are represented by a node in the graph, and the cross-agent interactions take the form of edges denoting a need for coordination between the connected agents. Each agent $i \in N$, where $N$ is the set of nodes (agents), performing an action $a_i \in A$, where $A$ is the action domain of $a_i$, has a local utility $f_i(a_i)$, while $f_{ij}(a_i, a_j)$ corresponds to a shared utility related to the edge $i, j \in E$, where $N$ is the set of edges. As such, the global utility $u(a)$ is defined as :

$$u(a) = \sum_{i \in N} f_i(a_i) + \sum_{(i,j) \in E} f_{ij}(a_i, a_j) \tag{1}$$

### 2.2 The Max-plus Algorithm

The Max-plus algorithm [7] is a message-passing algorithm that provides a solution to a CG representation of a coordination problem, i.e., provides an action for each participating agent $i$.

In every iteration, each agent $i$ sends locally maximized messages $\mu_{ij}(a_j)$ according to their current maximizing action $a_i$, to each one of their neighboring agents $j$ connected with an edge in the graph $i((i,j) \in E \forall j \in N_i$. Each message can be calculated by:

$$\mu_{ij}(a_j) = max_{a_i}\{f_i(a_i) + f_{ij}(a_i, a_j) + \sum_{k \in N_i \setminus \{j\}} \mu_{ki}(a_i)\} + c_{ij} \tag{2}$$

Convergence is only guaranteed when the CG does not contain cycles. A normalizing value of $c_{ij} = -\frac{1}{|N_k|}\sum_k \mu_{ik}(a_k)$ can be added to normalize the values of messages, so that they do not constantly accumulate when cycles exist in the graph. Finally, each agent $i$ selects the action $a_i$ that maximizes the received local messages $\mu_{ji}(a_i)$ along with $i$'s local payoff $f_i(a_i)$: $a_i = argmax_{a_i}\{f_i(a_i) + \sum_{j \in N_i} \mu_{ji}(a_i)\}$. Max-plus is an iterative algorithm, and is executed until convergence of the passing messages $\mu_{ij}$, or until a stopping criterion is met.

### 2.3 Max-plus in the Lane-Free environment

The adoption of the max-plus algorithm in the lane-free environment involves the construction of a CG as defined by the local interaction among agents [13]. Each lane-free vehicle is an agent $i$ depicted by a node $i \in N$ in the graph. Its interaction with nearby agents depends primarily on the distance between them. An agent $i$ considers nearby vehicles on the front and back within a certain longitudinal distance $dx$, which is set at $50m$. Now, each agent does not form connections with all observed agents, but only with those that there is an actual need for coordination, so as to avoid a potential collision. As such, the authors in [13] adopt Artificial Potential Fields to quantify the danger of collision between two agents $i$ and $j$, and incorporate this function into the local utilities. For that, the authors select the ellipsoid function to capture the potential collision in this domain. The form of the ellipsoid used is:

$$E(dx, dy) = \frac{m}{\left((\frac{|dx|}{\alpha})^{p_x} + (\frac{|dy|}{b})^{p_y} + 1\right)^{p_t}} \tag{3}$$

where $dx, dy$ are the longitudinal and lateral distance of the respective center points of the vehicles $i, j$. The parameters $a, b$ are used to adjust the range of the field for the $x, y$ axis, while the $p_x.p_y, pt$ affect the overall shape, and $m$ defines the magnitude when the distances are close to 0.

The local utility function contains two components, namely the "critical region" and "broader regions", as:

$$U_{ij}(s_{ij}) = E_c(dx_{ij}, dy_{ij}) + E_b(dx_{ij}, dy_{ij}, dv_{x,ij}, dv_{y,ij}) \tag{4}$$

Authors use a tuple of information relevant to the local state among the two agents with $s_{ij}$. The critical region $E_c$ is based solely on the distance of the agents, providing a positive value when agents are too close, while the broader region $E_b$ also accounts for the relative speed of the vehicles in both axes, capturing a broader view of the vehicles, informing when a collision is about to happen when vehicles approach one another with high speed. For more information on the Artificial Potential Fields used for the local utilities, we refer the interested reader to [13]. The maximum number of edges for forwards and backwards agents is also restricted, in order to control the graph's density. This selection process is performed based on the euclidean distance between agent $i$ and each neighbor agent $j$.

The agents' goal is to avoid collisions with their neighboring agents while trying to reach and/or maintain their assigned desired speed $v_{d,i}$. The local payoff $f_{ij}(a_i, a_j)$ incorporates that as a local edge utility function. The transition function is used for all combinations of joint action pairs, to provide the value of the potential field for the resulting state at the next time-step (depicted with $s'_{ij}$) to the local payoff $f_{ij}(a_i, a_j)$, that "informs" the agents on the outcome of their interaction.

Thus, the local payoff function $f_{ij}(a_i, a_j)$ shared by $i, j$ at local state $s_{ij}$ is:

$$f_{ij}(a_i, a_j) = \begin{cases} -U_{ij}(s'_{ij}), & U_{ij}(s'_{ij}) \neq 0 \\ c_s \cdot r_{v,ij}, & else \end{cases} \tag{5}$$

$$r_{v,ij} = r_{v,i} \cdot \frac{1}{|N_i|} + r_{v,j} \cdot \frac{1}{|N_j|} \tag{6}$$

where $|N_i|$ is the number of edges that contain agent $i$. The form of $r_{v,ij}$ is a linear function based on current speed $v_{x,i}$, normalized according to the desired speed $v_{d,i}$. This speed utility component is defined as: $r_{v,i} = (v_{d,i} - |v_{d,i} - v_{x,i}|)/v_{d,i}$. When the agents are close enough and in danger of a collision, the local payoff $f_{ij}(a_i, a_j)$ is negative. Otherwise, it is positive and reflects the goal of reaching the desired speed.

Finally, the action domain $A$ is discretized in order to comply with the max-plus algorithm, and each agent considers a set of 5 possible actions:

- a0: zero acceleration in both axes.
- a1: longitudinal acceleration of $2m/s^2$.
- a2: longitudinal deceleration of $2m/s^2$.
- a3: lateral acceleration $1m/s^2$ towards left.
- a4: lateral acceleration $1m/s^2$ towards right.

### 2.4 Related Work

Regarding the lane-free traffic application domain, many works already exist that propose relevant vehicle movement strategies, which tackle the problem from different research fields. First, the authors in [11] propose a rule-based vehicle movement strategy by adopting the notion of forces being applied to nearby vehicles, and this strategy is employed by the independent vehicles we introduce in the lane-free environment. Moreover, the work of [16] introduces an optimal control approach for the problem of lane-free driving, with vehicles optimizing their behavior by considering a future horizon and updating their trajectories online based on model-predictive control. Finally, the authors in [6] design a two-dimensional lane-free cruise controller with more emphasis on control theory.

Within the framework of CGs, there are works that tackle uncertainty in the environment already, but to the best of our knowledge, there is no work that extends max-plus based on our formulation, addressing the uncertainty of independent agents with decision-making criteria. In more detail, authors in [1] tackle coordination problems under uncertainty by devising Fuzzy Coordination Graphs, as they view the problem from the perspective of fuzzy games [8] and propose a variant of the variable elimination algorithm [4] to obtain the joint action. Moreover, in [9], agents' interactions are depicted in a graph structure, as in CGs, and the authors address the uncertainty for decentralized planning under uncertainty regarding the agents' observations. To do so, they incorporate the notion of beliefs into the Monte Carlo Tree search algorithm used for planning and use heuristic-based policies to predict other agents' actions.

## 3 Max-plus under Uncertainty

The main goal of this work is to extend the use of the max-plus algorithm to non-homogeneous lane-free environments. A non-homogeneous lane-free environment consists of additional agents that do not operate following the max-plus algorithm. We introduce new independent agents, with different behavior, that have no form of communication with other agents. This restriction imposes the challenge of predicting and modeling these agents in a way that is compatible with the max-plus algorithm, i.e., the message-passing operation of locally maximized messages sent among communicating agents. As such, to incorporate this new type of agent into the max-plus algorithm we emulate the messages that would be sent from independent agents to max-plus agents.

### 3.1 Emulated Messages

To apply the max-plus algorithm in a non-homogeneous lane-free environment, we first incorporate the independent agents in the CG accordingly. We consider a CG modeled as in [13], where each agent is represented by a node in the graph. Now, as in [13], we assume that each agent possesses observational capabilities, therefore agents can observe their surrounding vehicles' current status (position, speed, dimensions). As such, max-plus agents also observe independent agents nearby. Therefore, the observed independent agents are again represented by nodes, and edges that indicate a need for coordination can be formed (only) from the perspective of max-plus agents. However, the coordination between max-plus and independent agents cannot be achieved since there is no actual communication with independent agents.

The inability of non-cooperative agents to receive and read messages sent from max-plus agents means that sending any messages to them is ineffective. Thus, we establish a one-way communication between non-cooperative and max-plus in the form of emulated messages sent only from (observed) independent to max-plus agents. These messages are only emulated when the respective max-plus agents do observe independent agents and an edge that connects them exists within the CG. As mentioned, max-plus agents' can detect their neighbors' position and speed. This means that during the calculation of these messages, the longitudinal and lateral position of non-cooperative agents can be considered known, as well as the speed of the vehicles in both axes.

In Fig. 1 we present an example of our non-homogeneous environment and visualize the messages exchanged or emulated. Agents 2 and 3 follow the max-plus algorithm, while the agents 1 and 4 are independent, and they receive no messages. Agents 2 and 3 exchange messages for their actions, and also emulate messages from the observed agents 1 and 4.

### 3.2 Prediction Under Uncertainty

We reformulate the max-plus algorithm in order to incorporate emulated messages from other agents and take them into account for the decision-making process regarding the actions of the max-plus agents. For each neighbor, the agent checks if it is a max-plus agent, a fact ascertained by the receipt of the corresponding $\mu_{ij}$ message from that agent. Otherwise, they emulate the message from that agent. The pseudocode of our max-plus extension is provided in Algorithm 1. Messages emulated from observed agents that are not operating

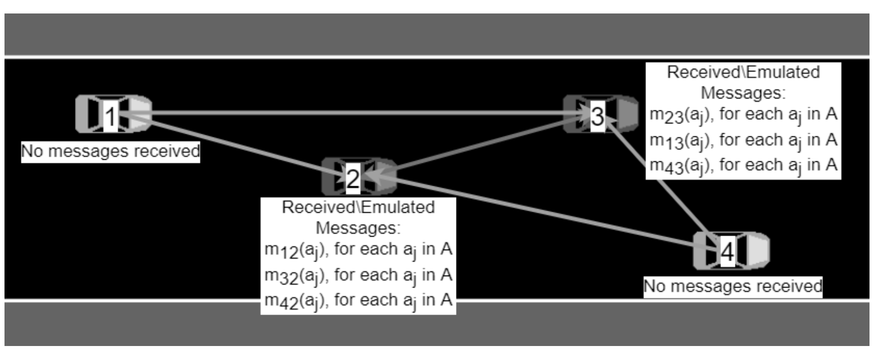


**Fig. 1.** The messages that each agent will receive.

**Algorithm 1** Max-plus algorithm with independent agents

```
procedure MAX_PLUS(N, E, A, class)
    for i ∈ N do                                          ▷ N is the set of agents
        neighbors ← ∪_{∀(i,j)∈E} {j}              ▷ ∀(i, j) ∈ E, only i may be independent
        for j ∈ neighbors do
            for a_j ∈ A do
                if class(i) ∈ maxplus then
                    μ_ij(a_j) ← max_{a_i} [f_i(a_i) + f_ij(a_i, a_j) + Σ_{k∈neighbors\i} μ_ki(a_i)] + c_ij
                    action[i] ← max_g_action(A, i, μ, neighbors)
                else
                    μ_ij(a_j) ← μ_toEmulate(a_j, i, j, A)
                end if
            end for
        end for
    end for
    return action
end procedure
```

according to max-plus, are calculated based on the "$\mu\_toEmulate$". Finally, the "$max\_g\_action$" simply returns the action for agent $i$ that maximizes its received messages $\mu_{ji}(a_i)$ (see [7] for more details). We should note that we also employ the anytime implementation of max-plus [7], but do not include it in this pseudocode in order to maintain simplicity. In what follows, we provide multiple criteria for the calculation of emulated messages, i.e., the implementation of "$\mu\_toEmulate$".

We now must specify the content of those emulated messages. Max-plus agents shall choose the best action for them and their neighbors, while still abiding by the max-plus algorithm, by choosing the action that maximizes the summation of the received messages. Considering there is no way of knowing for certain the intentions of non-communicative agents, our best option is to make assumptions regarding their action and the emulated messages should reflect this. These conditions of uncertainty, render the use of *decision rules (or decision-making criteria) under uncertainty* necessary. The ones examined in this paper are: the *Maximax criterion*; the *Wald's Maximin criterion*; the *Hurwicz criterion*; the *Savage's Minimax Regret*; and the *Laplace's criterion.*

**Maximax, Maximin and Hurwicz's Criterion** First, we examine three standard approaches for problems under uncertainty, the Maximax, Wald's Maximin [15] and Hurwicz [5] criteria. The Maximax criterion is an optimistic approach, since it makes the assumption that the best case scenario will always occur, and suggests an action that fits those conditions. On the contrary, Maximin considers the worst-case scenario due to the uncertainty that is associated with the complete lack of information about the possibilities, leading to a more pessimistic decision-making process. The Hurwicz criterion [5], introduced by Leonid Hurwicz in 1951, offers a "middle ground" option between the Maximax and Maximin criteria. The Hurwicz criterion attempts to find a balance between the extremes of the pessimism of Maximin and the optimism of Maximax. Hurwicz makes use of a $\beta$ temperature parameter, which acts as a measure of confidence in the decision maker regarding the probability of the best case scenario occurring, i.e., the $\beta$ value reflects the decision maker's willingness to take risks. The variable $\beta$ can take any value between 0 and 1.

When $\beta$ is set to 1, the Hurwicz criterion is reduced to the Maximin criterion, while $\beta = 0$ reduces Hurwicz to the Maximax criterion.

When the Maximax criterion is adopted, for an independent agent $i$ and a max-plus agent $j$, the message associated with an action $a_j$ of agent $j$ in accordance to the actions of agent $i$ is calculated as:

$$\mu_{max,ij}(a_j) = max_{a_i}\{f_{ij}(a_i, a_j)\} \tag{7}$$

Likewise, for the Maximin criterion:

$$\mu_{min,ij}(a_j) = min_{a_i}\{f_{ij}(a_i, a_j)\} \tag{8}$$

For the Hurwicz criterion, the message associated with each $a_j$ results from the weighted average of maximum payoff (multiplied by $\beta$) and minimum payoff (multiplied by 1 - $\beta$). For any value assigned to $\beta$, the message value that will be sent is formed by:

$$\mu_{ji}(a_j) = \beta \cdot \mu_{max,ji}(a_i) + (1 - \beta) \cdot \mu_{min,ji}(a_i) \tag{9}$$

**Dynamic calculation of $\boldsymbol{\beta}$ in Hurwicz Criterion** Typical uses of the Hurwicz criterion make use of $\beta$ as a constant, with a value between 0 and 1. However, our lane-free environment contains dynamic interactions among agents, and they encounter situations of interactions where a predetermined degree of optimism/pessimism may not be appropriate. As such, we consider that the distance between agents $i$ and $j$ can affect the optimism for the outcome of $i$'s action. A simple way of modelling a dynamic $\beta$ based on the distance between two agents is calculating the longitudinal distance between them and normalizing that value accordingly so that $\beta \in [0, 1]$. Therefore, $\beta$ is calculated as:

$$\beta = \frac{|dx_{ij}|}{dx_{max}} \tag{10}$$

where $|dx_{ij}|$ is the longitudinal distance between $i$ and $j$, and $dx_{max}$ is the maximum distance that two vehicles can be apart in the $x$ axis, and still be considered neighbors in the CG.

**Savage's Minimax Regret** Savage's Minimax Regret criterion [12] is an extension of Wald's Maximin criterion. Minimax Regret provides an alternative approach that tackles the unpredictability of the environment, by incorporating the notion of regret. To handle the uncertainty of the choices of other agents, instead of just maximizing the minimum possible payoff, we calculate the regret of each action. An action's regret in a specific state refers to the difference between the best payoff in that state, and the actual payoff produced when a particular action is performed. The Minimax Regret criterion minimizes the maximum regret an action of agent $j$ may have across all actions of agent $i$.

In systems with two agents $i$ and $j$, the regret of an action of $j$ is defined based on the possible outcomes when $i$ performs any of its available actions (of the set of actions $A$). Consequently, there are $|A|$ possible states. In the lane-free environment, a max-plus agent may have multiple neighbors, whose combination of actions result into different states. If $|N_j|$ is the number of agents neighboring a max-plus agent $j$ in the CG, the number of possible states occuring are $|A|^{|N_j|}$. First, considering only one independent agent $i$ observed by a max-plus agent $j$, for any state that is generated by the selected action $a_i$, the maximum regret of an action $a_j$ is defined as:

$$R(a_j) = max_{a_i \in A}\{max_{a_k \in A}\{f_{ij}(a_i, a_k)\} - f_{ij}(a_i, a_j)\} \tag{11}$$

where the calculation within the max operator for the actions of $i$ depicts the element $a_i, a_j$ of the regret table.

The criteria we examined so far only form messages based on the actions of individual neighbors. However, for Minimax Regret, viewing each neighbor individually is inappropriate, as the resulting messages, consisting of regret values are not properly combined through a simple summation process. As such, Minimax Regret takes into account all independent neighbors from the perspective of each max-plus agent. Consequently, we must calculate the payoffs for each state created by the combination of action of the neighbors. Thus, given a max-plus agent $j$, and a set of $p$ independent agents $\{i_1, \cdots, i_p\}$ connected with $j$ within the CG, the maximum regret of each action $a_j$ is calculated as:

$$R(a_j) = max_{\{a_{i_1}, \cdots, a_{i_p}\} \in A^p}\Big\{max_{a_k \in A}\Big\{\sum_{i=i_1}^{i_p} f_{ij}(a_i, a_k)\Big\} - \sum_{i=i_1}^{i_p} f_{ij}(a_i, a_j)\Big\} \tag{12}$$

where we are interested in minimizing $R(a_j)$ instead of maximizing. As such, the associated message is: $\mu_{ij}(a_j) = -R(a_j)$, where the index $i$ now reflects the whole set of independent agents that $j$ observes and is connected to, meaning that in contrast to all other criteria, we emulate a single set of messages for all independent agents connected to $j$.

To calculate $R(a_j), \forall j \in A$, we use a tree to construct the joint action space, and obtain the sum of the local functions associated with each independent agent. Starting at the root, we create $|A|$ children and attach to them the associated local message payoff, resulting from the joint action of $j$ with the neighbor $i_1$. We then iteratively expand each child node according to the actions of the $i_k$ neighbor and attach to each node the associated local message payoff plus the value of parent node. This is repeated for all neighbors $p$. As such, each leaf node will contain the sum of local payoffs associated with the corresponding joint action of all neighbors.

**Laplace's Equal Likelihood Criterion** The Laplace criterion [3] is based on the principle of insufficient reason. Essentially, it states that if there is no sufficient reason to assume the probabilities of any scenario occurring, we can only infer that all possible outcomes occur with the same probability. For each action agent $i$ may take, we assign the same probability. Since we consider all agents in our environment have the same set of available actions as our agents, the probability assigned to each action is $\frac{1}{5} = 0.2$. The message attached to the action $a_j$ of max-plus agent $j$ from non-cooperative agent $i$, is formed by calculating the average payoff for all actions of $i$:

$$\mu_{ij}(a_j) = \sum_{a_i \in A} \frac{1}{|A|} \cdot (f_{ij}(a_i, a_j)) \tag{13}$$

Thus, the Laplacian criterion considers each action to be occurring with the same frequency. This of course cannot possibly hold true for autonomous agents in a lane-free environment, which are expected to be adopting different driving behaviours and strategies. As such, we expect that classifying independent agents into different behavioural types, and tracking their actions in an opponent modelling fashion, could be beneficial in terms of computing more accurate average payoff estimates and thus coordination messages.

As a first step towards that direction, we devise a simple opponent model by classifying each independent agent according to its surroundings. We detail that model immediately below.

**Opponent Modelling** The behavior of drivers in real-life scenarios is heavily dependent on the vehicles in close proximity. For instance, a driver will not accelerate when another is directly in front of her and will be reluctant to slow down to avoid hitting cars that are in her rear.

We use a simple heuristic in order to classify each independent agent by the number of their respective neighbors. For an independent agent $i$, we distinguish each neighbor $k$ (within distance of $d_o = 50m$ from the perspective of $i$) based on the relative position from $i$, i.e., we recognize that $k$ is in front of $i$ when its relative longitudinal position is greater than 0 ($dx = x_k - x_i$). Similarly, $k$ is considered to be on the left or right w.r.t. $i$ based on their respective lateral placement. Based on these values we consider each neighbor of the independent agent to be either at its *front-left, front-right, rear-left* or *rear-right.*

An illustrative example is provided in Fig. 2, where vehicle 0 is an independent agent with five other agents $1, 2, 3, 4$ and 5 in its surroundings. Each neighboring agent of 0 must be in one of the 4 regions to be characterised as a front-left, front-right, rear-left or rear-right neighbor. Based on these areas, we consider be 1 on the rear-left, 2 and 3 on the rear-right, 4 on the front-right, and 5 on the front-left of independent agent 0.

Then, for independent agent $i$ we count the number of neighbors on each region and classify $i$ by this information. To bound the number of classes, we consider at most five agents within each aforementioned region, prioritizing according to the agents' distance from $i$. That means each agent $i$ belongs to a

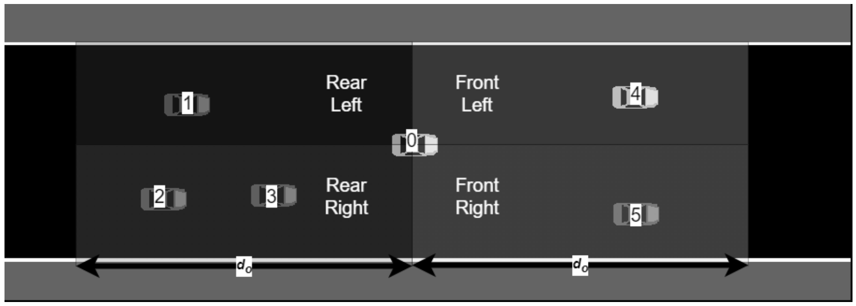


**Fig. 2.** Showcasing the assignment of an independent agent's neighbors to each region.

category described by a tuple: $\langle n_0, n_1, n_2, n_3 \rangle$, where elements are the number of front-left, front-right, rear-left, and rear-right neighbors respectively. This results in $6^4 = 1296$ different classes of agents (each element has six possible states, from 0 to 5 agents). To determine the probability of an action that an independent agent may take, we first observe their actions and update the frequencies of their actions accordingly.

Notice that the acceleration of independent agents can be observed implicitly by the max-plus agents at each time-step, through the speed update. Thus, the acceleration of an independent vehicle is calculated by $a_c = \frac{v_t - v_{t-1}}{time\_step}$, where $v_{t-1}, v_t$ is the longitudinal speed of an independent agent at two consecutive time-steps. Any independent agent may operate directly in the continuous domain, i.e., have continuous values for acceleration. We convert these to the available set of discrete actions in order to be compatible with max-plus. We remind the reader that we have a set of 5 discrete actions, with action $a_0$ being equivalent to zero acceleration across all axes. We set a threshold value of $c_t$. If the acceleration of an independent agent does not exceed these threshold in both axes, we assume they perform the action $a_0$. The actions $a_1, a_2$ that correspond to movement in the x axis (acceleration and deceleration respectively), are assumed when the agent's longitudinal acceleration exceeds $c_t$ in the corresponding direction. Similarly for the lateral acceleration.

Finally, after collecting information from independent agents, the emulated message attached to the action $a_j$ of a max-plus agent from an independent agent $i$, is formed by calculating the weighted average payoff for all actions of $i$, and Equation 13 now becomes:

$$\mu_{ij}(a_j) = \sum_{a_i \in A} w_{class}(i, a_i) \cdot (f_{ij}(a_i, a_j)) \tag{14}$$

where $w_{class}(i, a_i)$ returns the measured weight (i.e., frequency of occurrence) of action $a_i$ for the associated class of $i$, by accessing information regarding $i$'s neighbors for the classification.

## 4 Experimental Evaluation

In this section we present our experimental evaluation where we first introduce independent lane-free agents. In order to investigate more "extreme" conditions, we also examine independent agents with added noise, which naturally adds to the uncertainty. Then, we provide our experimental results for 2 different distributions of lane-free agents and independent agents, and for all levels of noise considered.

### 4.1 Lane-free Independent Agents (with Noise)

We introduce independent agents based on a rule-based approach in lane-free traffic environments [11]. These agents behave rationally, in the sense that they try to maximize speed while actively trying to avoid collisions with other agents by observing nearby vehicles. We refer the interested reader to [11] for more information on these agents' movement strategy. In order to properly evaluate our proposed approach, and increase the uncertainty induced, we add noise to the control of these independent agents. The two acceleration values $a_x, a_y$ of each independent agent in a particular time step, resulting from its policy, is filtered with additional noise. The actions $a'_x$ and $a'_y$ that the agent will actually perform are: $a'_x = a_x + n_{px} \cdot a_x$ & $a'_y = a_y + n_{py} \cdot a_y$, where $n_p \sim \mathcal{U}(-c_p, c_p)$ and $\mathcal{U}$ is a uniform distribution. Note that for small values of $a_x, a_y$, i.e., when the agents maintain the same speed, the added noise will have a negligible effect since it depends on the values of the initially chosen accelerations $a_x, a_y$. As such, we also examine a second type of noise, $n$, which is independent of the accelerations of the new vehicles, $a'_x = a_x + n_x$ & $a'_y = a_y + n_y, n \sim \mathcal{U}(-c', c')$, and has an increased effect as we observe from the experimental evaluation.

Summarizing, the three types of independent agents we introduce to our environment are:

- Type A: Lane-free agents with no noise
- Type B: Lane-free agents with noise $n_p \sim \mathcal{U}(-0.5, 0.5)$
- Type C: Lane-free agents with noise $n \sim \mathcal{U}(-1.0, 1.0)$

As mentioned, the independent agents incorporate a different policy, that does not rely on communication/coordination among agents. Also, their acceleration values are continuous, while max-plus agents operate on a discretized action domain, thus making the prediction of what the next action for each agent will be even more difficult.

### 4.2 Simulation environment

To examine the effectiveness of each criterion we use an extension of SUMO, designed for lane-free traffic [13]. We extend the SUMO environment setup in [7], to include both max-plus agents and agents based on [11], and can adjust the distribution of the different varieties of agents (e.g., max-plus and independent in

**Table 1.** Simulation parameters.

| Parameter | Value |
|---|---|
| Highway Length | 5 km |
| Highway Width | 10.2 m |
| Vehicle length | 3.2 m |
| Vehicle width | 1.6 m |
| Simulation time | 1 hr |
| time-interval | 0.25 s |
| $v_d$ (desired speed) | $[25, 35]\frac{m}{s}$ |
| $v_{x,init}$ (initial speed) | $25\frac{m}{s}$ |
| Inflow rate | $7200\frac{veh}{hr}$ |

our case) entering the simulation environment. This gives us the opportunity to control the penetration rate of max-plus and independent agents in the highway, and observe the interaction between them. In this environment, we examine and compare the number of collisions between max-plus agents and independent ones. In every time-step, we consider that a collision occurs when two vehicles' positions overlap.

### 4.3 Experiments and Results

For our evaluation, we introduce a baseline criterion with a simplistic assumption, to provide more incentives for the use of decision-making criteria. Specifically, as a baseline criterion, we assume that independent agents always perform action $a_0$, i.e., 0 acceleration in both axes, meaning that the emulated messages have the form: $\mu_{ij}(a_j) = f_{ij}(a_0, a_j)$.

The parameters relevant to the lane-free scenario we examined are shown in Table 1. We examined two different configurations regarding the distribution of vehicles. Specifically, in our first scenario, 40% of our CAVs population consists of independent agents; while in the second scenario, the independent agents are 60% of all CAVs. Both distributions contain results for the three types of independent agents (i.e., types A, B, and C), as discussed in Sec. 4.1.

**Results** Experimental results are provided in Figs. 3 and 4, for our first and second scenario respectively. The results shown are averages across 10 runs with different seed values each (the seed value for each run are the same across all experimental configurations). Code was written in Python 3, and simulations were executed on a PC with an Intel i7-7700k CPU and 16GB of RAM. Each run of 1 hour of simulation required approximately 45 minutes, with the Minimax Regret criterion adding a small overhead of around 5 minutes. We report that we observed an average speed within the range $[28.6, 29.3]m/s$ for all different seed values. The speed deviation between each criterion was not significant, and variations were observed due to the different seed values.

Regarding collisions, a first observation is that the baseline criterion exhibits more collisions than any of the criteria we used: as shown in both figures, for each

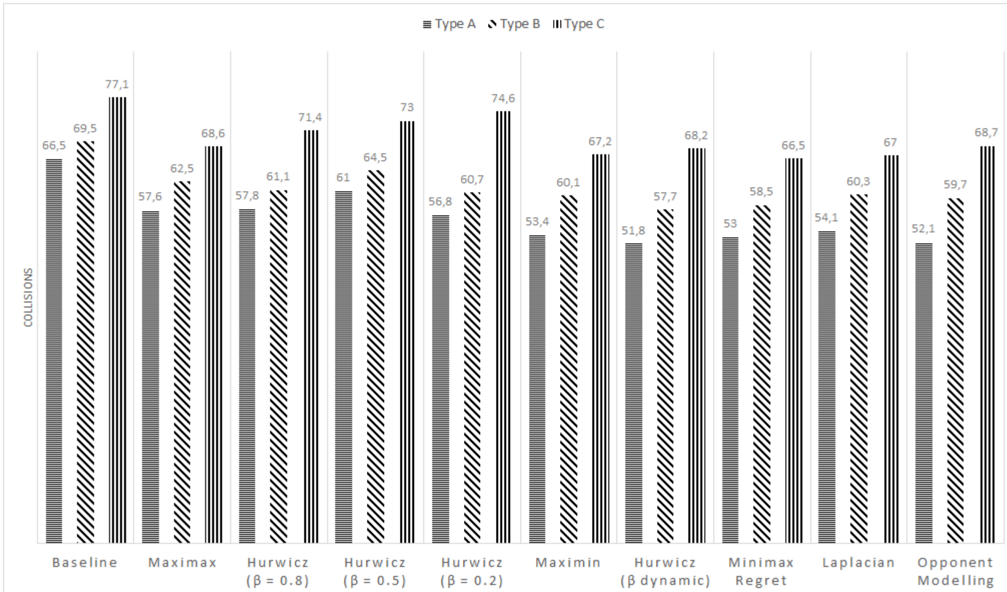


**Fig. 3.** Collisions per hour for experiments with a distribution of 60% max-plus agents and 40% non-max-plus agents.

independent agent type, the performance of the baseline approach is consistently worse than any decision-making criterion used. This motivates the use of more elaborate ways to address the uncertainty regarding other agents, as it clearly affects performance. It is important to note that while the baseline performs worse when compared to the agents following the more elaborate decision-making criteria, agents using the baseline criterion still use our extension of the max-plus algorithm and do observe the other vehicles in the highway.

Hurwicz allows us to balance both the best- and worst-case scenarios. However, standard uses of Hurwicz under-perform, resulting into more collisions when compared even to the more naive Maximax criterion. Only the use of a dynamic $\beta$ provides a noticeable improvement, which allows us to adjust our optimism depending on how close the vehicles are. This leads to fewer collisions compared to the use of the Hurwicz criterion with fixed $\beta$. Intuitively, one could assume that the pessimism of Maximin may be excessive, especially when there is no noise added. This intuition proved false, as Maximin performs better than both Maximax and Hurwicz with constant $\beta$, presumably due to the fact that independent agents have a distinct methodology of choosing their actions that does not match with the local functions of max-plus agents. That means the conservative approach of Maximin fairs better with them than initially expected.

The performance of the Minimax Regret approach lies between that of Maximin and Hurwicz with dynamic $\beta$, but Minimax Regret provides somewhat better results for the high noise levels (Type C agents), as can be observed in

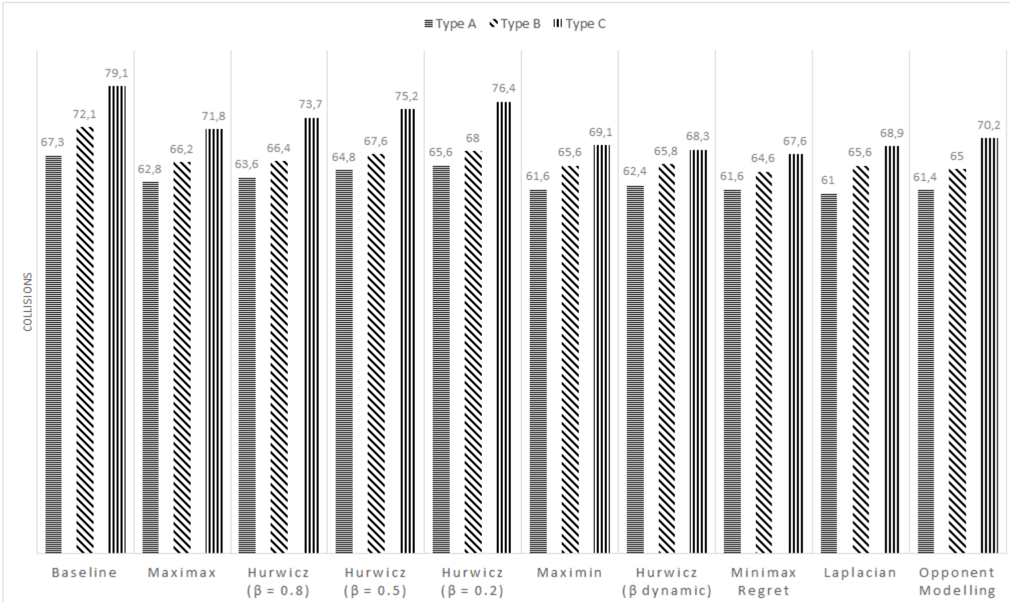


**Fig. 4.** Collisions per hour for experiments with a distribution of 40% max-plus agents and 60% non-max-plus agents.

both figures. Apparently, the notion of regret along with the joint view of all independent agents, helps in environments with higher unpredictability (increased noise levels).

The Laplacian approach provides similar results with all aforementioned criteria, and has a slight advantage when the levels of noise are high.

However, for lower noise levels (i.e., for independent agents of Type A & B), the Hurwicz with dynamic $\beta$ provides slightly better results, indicating that our heuristic function for $\beta$ performs better in low noise environments.

For the opponent model, we set the associated threshold to $c_t = 0.5$, and collect data from 10 1-hour simulations (with different seed values) using the Laplacian approach. Opponent modeling provides only a marginal improvement w.r.t. the Laplacian when the vehicles do not have noise for the first configuration (cf Fig. 3), since the observations used for estimating the frequencies of actions for each class do coincide with the policy of the independent agents (not filtered with noise). Of course, this is not the case when noise is added, due to the observed behavior being partially inconsistent. We believe that a more refined opponent model would enhance the results, and therefore it constitutes an imminent future research endeavor.

In general, replacing max-plus agents with more independent agents leads to more uncertainty in our decision making (cf Fig. 4), resulting in more collisions and smaller margins between both the different criteria, and the different types.

## 5 Conclusions and Future work

In this paper we extended the application of the max-plus algorithm to the lane-free environment [13] in order to render it compatible with agents not obeying the same algorithm. We did so by estimating the actions of the other agents using different criteria that tackle uncertainty, and our experimental evaluation exhibited improvement when those criteria are in effect. Notably, max-plus agents now encounter vehicles whose available range of actions is significantly larger than their own set of actions.

In future work, we plan to expand our work and establish ways to incorporate continuous actions of observed agents on the emulated messages. Moreover, it would be interesting to combine the proposed approach with the work of [14] which introduces a dynamic discretization variant of the algorithm, that enables its use in continuous action domain by lifting the task of predetermining a constant number of appropriate discrete actions. Furthermore, as mentioned already, the opponent model is quite simplistic, and can be re-examined so as to incorporate more features that are important (such as the speed of the vehicles)—and to also address the quantification of uncertainty, potentially by using probabilistic opponent modelling techniques [2] along with incorporation of domain knowledge (i.e., the expected behavior of vehicles). Finally, we intend to investigate the application of the distributed max-plus variant [7], and compare with our work in this paper.

## References

1. Ahmadzadeh, H., Masehian, E.: Fuzzy coordination graphs and their application in multi-robot coordination under uncertainty. In: 2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM). pp. 345–350. IEEE (2014)
2. Albrecht, S.V., Stone, P.: Autonomous agents modelling other agents: A comprehensive survey and open problems. Artificial Intelligence **258**, 66–95 (2018)
3. Aldea, C., Olariu, C.: Selecting the optimal software solution under conditions of uncertainty. Procedia - Social and Behavioral Sciences **109**, 333–337 (2014)
4. Guestrin, C., Koller, D., Parr, R.: Multiagent planning with factored mdps. In: Advances in Neural Information Processing Systems. vol. 14. MIT Press (2001)
5. Hurwicz, L.: Some specification problems and applications to econometric models. Econometrica **19**(3), 343 – 344 (1951)
6. Karafyllis, I., Theodosis, D., Papageorgiou, M.: Two-dimensional cruise control of autonomous vehicles on lane-free roads. In: 60th IEEE conference on Decision and Control. pp. 2683–2689. CDC (2021)
7. Kok, J.R., Vlassis, N.: Collaborative multiagent reinforcement learning by payoff propagation. J. Mach. Learn. Res. **7**, 1789–1828 (2006)
8. Larbani, M.: Non cooperative fuzzy games in normal form: A survey. Fuzzy Sets and Systems **160**(22), 3184–3210 (2009)
9. Li, M., Yang, W., Cai, Z., Yang, S., Wang, J.: Integrating decision sharing with prediction in decentralized planning for multi-agent coordination under uncertainty. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. p. 450–456. IJCAI (2019)

10. Mulla, A.K., Joshi, A., Chavan, R., Chakraborty, D., Manjunath, D.: A microscopic model for lane-less traffic. IEEE Transactions on Control of Network Systems **6**(1), 415–428 (2019)
11. Papageorgiou, M., Mountakis, K.S., Karafyllis, I., Papamichail, I., Wang, Y.: Lane-free artificial-fluid concept for vehicular traffic. Proceedings of the IEEE **109**(2), 114–121 (2021)
12. Pérez-Galarce, F., Álvarez Miranda, E., Candia-Vejar, A., Toth, P.: On exact solutions for the minmax regret spanning tree problem. Computers & Operations Research **47**, 114–122 (07 2014)
13. Troullinos, D., Chalkiadakis, G., Papamichail, I., Papageorgiou, M.: Collaborative multiagent decision making for lane-free autonomous driving. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems. p. 1335–1343. AAMAS (2021)
14. Troullinos, D., Chalkiadakis, G., Samoladas, V., Papageorgiou, M.: Max-sum with quadtrees for decentralized coordination in continuous domains. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22. pp. 518–526. International Joint Conferences on Artificial Intelligence Organization (2022)
15. Wald, A.: Statistical decision functions which minimize the maximum risk. Annals of Mathematics **46**(2), 265–280 (1945)
16. Yanumula, V.K., Typaldos, P., Troullinos, D., Malekzadeh, M., Papamichail, I., Papageorgiou, M.: Optimal path planning for connected and automated vehicles in lane-free traffic. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). pp. 3545–3552 (2021)