Conditional Max-Sum for Asynchronous Multiagent Decision Making

Dimitrios Troullinos
Technical University of Crete
Chania, Greece
dtroullinos@tuc.gr

Georgios Chalkiadakis
Technical University of Crete
Chania, Greece
gchalkiadakis@tuc.gr

Ioannis Papamichail
Technical University of Crete
Chania, Greece
ipapamichail@tuc.gr

Markos Papageorgiou
Technical University of Crete
Chania, Greece
mpapageorgiou@tuc.gr

ABSTRACT

In this paper we present a novel approach for multiagent decision making in dynamic environments based on Factor Graphs and the Max-Sum algorithm, considering *asynchronous* variable reassignments and distributed message-passing among agents. Motivated by the challenging domain of lane-free traffic where automated vehicles can communicate and coordinate as agents, we propose a more realistic communication framework for Factor Graph formulations that satisfies the above-mentioned restrictions, along with *Conditional Max-Sum*: an extension of Max-Sum with a revised message-passing process that is better suited for asynchronous settings. The overall application in lane-free traffic can be viewed as a hybrid system where the Factor Graph formulation undertakes the *strategic* decision making of vehicles, that of desired lateral alignment in a coordinated manner; and acts on top of a rule-based method we devise that provides a structured representation of the lane-free environment for the factors, while also handling the underlying control of vehicles regarding core operations and safety. Our experimental evaluation showcases the capabilities of the proposed framework in problems with intense coordination needs when compared to a domain-specific baseline without communication, and an increased adeptness of Conditional Max-Sum with respect to the standard algorithm.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent systems**; **Cooperation and coordination**; • **Applied computing** → **Transportation**.

KEYWORDS

Max-Sum algorithm; distributed AI; DCOP; autonomous driving

ACM Reference Format:
Dimitrios Troullinos, Georgios Chalkiadakis, Ioannis Papamichail, and Markos Papageorgiou. 2025. Conditional Max-Sum for Asynchronous Multiagent Decision Making. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

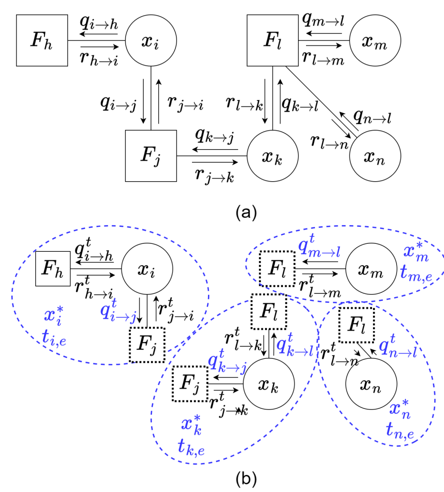

Figure 1: (a) An example of a typical FG; (b) The corresponding distributed structure for FGs.

1 INTRODUCTION

Distributed Constraint Optimization Problems (DCOPs) [3] effectively tackle multiagent decision making in large problems that can be formulated with a highly decomposable structure. DCOPs have been widely studied with several extensions that address broader domains such as dynamic and/or sequential environments. Algorithmic solutions for Dynamic DCOPs usually expect that agents update their existing configuration concurrently in an online dynamic environment, or that multiple message-passing iterations can be taken before agents update their current configuration as a group. These limitations can hinder the actual use of DCOPs in many multiagent environments requiring more flexible frameworks that incorporate communication or timing-related restrictions. Existing research either tackles the asynchrony of the agents' decision making problem without taking into account large-scale and open distributed environments, by resorting to pseudo-tree graph structures that are more immutable than Factor Graphs [16]; or requires the synchronization of agents' decisions at every time-step [3].

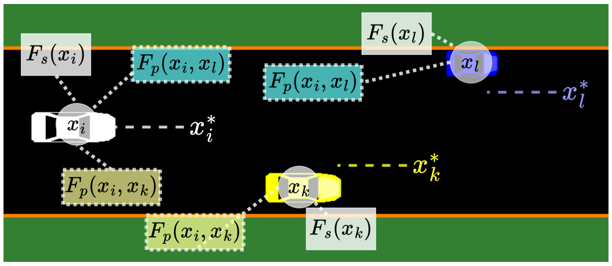

Figure 2: FG example of connected lane-free vehicles.

To this end, we propose a distributed communication framework for Factor Graphs (FGs) that relies on *information broadcasting* for communication among agents, in a way that treats agents as independent entities and not as a unified system. [1]

A visual illustration of the framework juxtaposed with the corresponding conventional FG (contains variables x_i that agents control, and decomposed factors F_j of local utilities depending on variables x_i) can be found in Fig. 1 (discussed in length in Sec. 3). FG formulations are often tied with the Max-Sum algorithm, which showcases great adaptability under different environments [2, 18, 21], and through its numerous extensions [19, 20, 24] that solve different types of DCOPs. In a similar vein, we put forward an extension of Max-Sum, termed as *Conditional Max-Sum*, which addresses the asynchrony on the agents' variable updates. Conventional Max-Sum is not fully fitted for asynchronous decision-making since it is designed for static environments; while existing research on Asynchronous Max-Sum [24] has a different focus, as it examines the effect of asynchronous communication schemes in static problems, instead of asynchronous updates in sequential problems.

The proposed approach is applied in *lane-free traffic* [14], a novel paradigm that investigates traffic environments where autonomous vehicles can fully utilize the lateral road capacity and do not obey the lane principle. This gives rise to a challenging multiagent coordination environment, with a FG formulation that we illustrate in Fig. 2. There, the vehicles have to coordinate their lateral placement through the control variable x_i, and update their desired lateral placement x_i^*, i.e., their target to reach. Existing work [21] in lane-free driving as a multiagent problem relies on the conventional framework of FGs and Max-Sum. As such, it does not take into account realistic restrictions regarding distributed communication and asynchrony of an open environment—especially in this setting where each agent is an independent moving vehicle with observations and dependencies that constantly change over time.

Summing up our contributions, here we propose a distributed communication framework for FGs which allows for asynchronous decision making, giving rise to the *Conditional Max-Sum* algorithm. Moreover, we instantiate our framework in a realistic lane-free traffic environment for which we devise a novel formulation based on FGs. Our experimental evaluation demonstrates the coordination efficacy of the overall framework in this domain, and autonomous vehicles under Conditional Max-Sum exhibiting an increased ability to respond quickly and better target their desired speed objective with smoother lateral maneuvers.

[1] The extended version of this paper containing the Appendix referenced throughout the text is available in arXiv with the same title. The codebase can be found at https://bitbucket.org/dtrou/conditional-max-sum-in-lane-free-traffic.

2 BACKGROUND AND RELATED WORK

2.1 Factor Graphs and the Max-Sum Algorithm

Factor Graphs (FGs) [11] originate from probabilistic graphical models but are also well integrated within distributed AI as a common tool for DCOP formulation [2, 3]. Given an FG structure as the one in Fig. 1(a), we seek to obtain all control variable configurations $x_i \in \mathbf{x}$ that maximize the sum of the factors, i.e., solve the optimization problem: $\mathbf{x}^* = \arg\max_{\mathbf{x}} \sum_j F_j(\mathbf{s}_j)$, where $\mathbf{s}_j \subseteq \mathbf{x}$. Note that factors F_j connect different variables, and may potentially associate more than two (e.g., F_l). The corresponding vector $\mathbf{s}_j$ contains all variables x_i connected to factor F_j. For instance, in Fig. 1(a): $\mathbf{s}_l = [x_k, x_m, x_n]^T$. In general, the factors can depend on any subset $\mathbf{s}_j \subseteq \mathbf{x}$ of the control variables.

Max-Sum [2] is an iterative, inference-based algorithm that provides an approximate solution for this optimization problem through a message-passing operation involving two types of messages. The first type of messages concerns values sent from variable i to factor j: $q_{i\to j}(x_i) = c_{ij} + \sum_{k\in M_i\setminus j} r_{k\to i}(x_i)$, where M_i is the set of factor indices that variable i is connected to. For instance, $M_k = \{j, l\}$ in Fig. 1(a). As such, $q_{i\to j}(x_i)$ contains an estimate for each value of x_i to be sent to factor j. Essentially, $q_{i\to j}(x_i)$ performs propagation of evaluations from all the other connected factors $M_i \setminus j$, and can be viewed as the agent's current "intents" regarding their final configuration of variable x_i. Then, c_{ij} is a normalization constant that satisfies $\sum_{x_i} q_{i\to j}(x_i) = 0$. This normalization is important in cyclic graphs since we need to bound the messages' values. While convergence in cyclic FGs is not guaranteed, the use of c_{ij} has proven to be quite effective in many studies [2, 10, 12].

The second type of messages involves evaluations that a variable i receives from a connected factor j: $r_{j\to i}(x_i) = \max_{\mathbf{s}_j\setminus x_i}[F_j(\mathbf{s}_j) + \sum_{k\in N_j\setminus i} q_{k\to j}(x_k)]$, where N_j is the set containing the variable indices connected to factor j, and the maximization process involves all the variables $\mathbf{s}_j$ connected to factor F_j without x_i. For instance, in Fig. 1(a), $N_l = \{k, m, n\}$, meaning that variables x_k, x_m, x_n are connected to factor F_l. These messages calculate for each possible value x_i an estimate for the outcome considering both the associated factor, and all other messages sent to it. Since the $q_{k\to j}$ values provide these connected N_l variables evaluation for their respective decisions, agent i maximizes over all these variables, in order to have an evaluation that incorporates both the immediate factor's value and other agents' influence. In every new iteration, the previous evaluations of the messages are utilized. This operation is performed until some stopping criterion (e.g., time, iterations number), or if all message values converge (within a threshold). Finally, each agent computes the optimal value as: $x_i^* = \arg\max_{x_i} \sum_{j\in M_i} r_{j\to i}(x_i)$, i.e., the x_i that maximizes the sum of all the received messages. Without loss of generality, we consider that each agent controls one variable, and refer (same indexing) to agents and variables interchangeably.

2.2 Related Work

Dynamic DCOPs (D-DCOPs) [3] extend conventional DCOPs to dynamic environments. Consequently, the notion of time is introduced to the problem. In an FG problem formulation, we consider a dynamic graph structure with the possibility to change in time. At a time-step t, the FG contains a set of factors M, with $F_j(\mathbf{s}_j) \in M$, and $x_i \in N$, where N is the set of variables. There are two types of changes for every update from t to $t+1$: (a) existing factors $F_j(\mathbf{s}_j)$ can now contain different values; or (b) new variables x_i and factors $F_j(\mathbf{s}_j)$ can be introduced, and existing ones can be removed. Commonly, D-DCOPs do not model how the factor evolves over time and algorithmic extensions [3] re-solve every new static instance of the problem at time t, effectively incorporating domain knowledge to improve performance [18]. Prior work in lane-free environments [21] (and [20]) also solves every new instance of the problem by relying on the previous solution as a starting point, with the assumption that changes in the graph modelling vehicles' interactions will not be substantially different. In contrast, [13] explicitly model the FG's evolution as Markovian D-DCOPs, thereby they tackle the sequentiality of the problem and incorporate methods from RL. Moreover, [1] addresses D-DCOPs from the perspective of DecMDPs, which can be viewed as a similar problem, and perform planning based on Monte-Carlo tree search with Max-Sum.[2]

Finally, we note that papers applying Max-Sum on Mobile Sensor Teams (MSTs), dealing with exploration issues [23] or collision avoidance [15], exhibit conceptual similarities to our domain (which is natural since MSTs involve a dynamic environment of moving agents). Regardless, they do not deal with asynchrony in the agents' variable updates as we do in this paper.

3 ASYNCHRONOUS DECISION MAKING WITH THE MAX-SUM ALGORITHM

We now present a novel extension of Max-Sum for problems that can be formulated as D-DCOPs with specific constraints that render them more realistic in a large-scale distributed coordination environment. The first restriction we impose is we cannot have multiple iterations of the algorithm, since in an open distributed environment we cannot easily assume that agents will be able to communicate multiple times before updating their decision and iteratively propagate messages. We rather rely on a more realistic notion of *information broadcasting* from the perspective of each agent. This is visualized in Fig. 1(b), where each agent broadcasts all q messages to be sent to connected factors j that involve other agents as well, with additional information (indicated with blue color) relevant to the Max-Sum extension we later establish. At time-step t, agents first observe the broadcasted q messages of nearby agents, then calculate their r messages followed by the updated q messages to be broadcasted. As such, at time-step t, r^t messages now rely on the previously broadcasted messages q^{t-1}. Otherwise, we would need to either establish some form of order among agents' message updates (not realistic and potentially restrictive in large environments), or examine the effect of asynchronous message-passing (essentially a different problem) in this setting, as put forward and studied in [24]. Therefore, each agent i at time-step t can update its received messages as:

$$r^t_{j\to i}(x_i) = \max_{\mathbf{s}_j \setminus x_i} \left[F^t_j(\mathbf{s}_j) + \sum_{k \in N_j \setminus i} q^{t-1}_{k \to j}(x_k) \right] \tag{1}$$

[2]The Max-Plus algorithm (employed in [1]) is effectively the same method with Max-Sum when the FG follows a structure that contains factors up to 2 variables.

and then its q^t messages to be broadcasted to other agents:

$$q^t_{i\to j}(x_i) \quad = \quad c_{ij} + \sum_{k \in M_i \setminus j} r^t_{k \to i}(x_i) \tag{2}$$

Naturally, this choice comes at the cost that a single iteration of the algorithm will probably result in a significantly subpar solution quality. However, in realistic dynamic environments such as the lane-free traffic domain, it is not reasonable for agents to update their configuration at every time-step, since they would not be able to commit to their decision and probably result in oscillatory behaviour. In [17], authors are motivated by the same assumption and establish the notion of commitment deadlines for agents regarding their update. Likewise, before introducing asynchronous operation, we can provisionally define a common time-period T for agents' updates, where all agents can adjust their configuration x^*_i based on the received messages r. More specifically, agents update $x^*_i = \arg\max_{x_i} \sum_{j \in N_i} r^t_{j\to i}(x_i)$ periodically, while in intermediate steps they can only communicate and perform one iteration of the algorithm, i.e., update with Equations 1 & 2, and maintain the last update of x^*_i. This effectively allows us to: *(1)* perform multiple iterations before agents update their decision but now in a distributed setting, and *(2)* avoid indecisive behaviour of agents due to the dynamic nature of the problem. Certainly, this assumes (as in related work) that the FG does not change abruptly within this time-period T required for the update. Otherwise, the previously exchanged messages have diminishing value.

Even with this common time-period T, there is a need for a *synchronized* clock among agents, imposing to them whenever they should update their decisions. We additionally lift this requirement, and allow for *asynchronous* variable updates x_i for each agent. This introduces a sense of autonomy as well since agents are not bound to make decisions alongside others, and can react in a timely manner depending on their local observations. As a result, at every time-step t, a subset of agents update their existing configuration. In this more flexible framework, one could simply apply the standard Max-Sum algorithm with the use of message passing operations in Equations 1 & 2, but with the distinction that each agent updates x^*_i at a potentially different time-step from others. However, the calculation of incoming r^t messages in Max-Sum (Eq. 1) entails a critical assumption: the local maximization from the perspective of an agent i is performed based the propagated q messages that reflect the final configuration, i.e., agent i updating its variable at time t *assumes* that all other agents connected to it through a factor will do so concurrently.

3.1 Conditional Max-Sum Algorithm

In order to efficiently tackle asynchronous decision making in this new context, we need to redefine the message passing equation for r^t messages. For this, we request supplementary information from agents, namely a *time-estimate* $t_{i,e}$ from each agent i regarding its next update and its last variable configuration x^*_i (cf Fig. 1(b)). As mentioned, the issue lies with the local maximization of connected variables in r^t messages, due to the underlying assumption of an one-time variable configuration for all agents. We extend this notion by performing a *conditional* maximization depending on the *relative* time-estimates between agent's $t_{i,e}$ and the other agents' estimates:

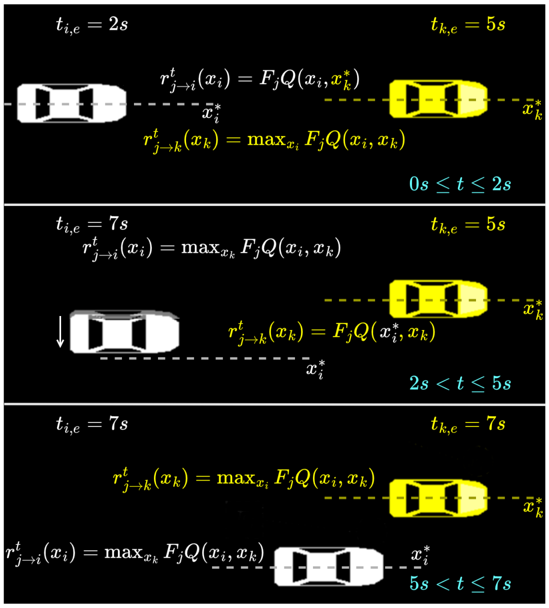

Figure 3: Illustrative example of two agents coordinating with Conditional Max-Sum to update their decisions asynchronously.

$\mathbf{t}_{j,e} = \{t_{k,e} : \forall k \in N_j \setminus i\}$, and their existing configuration $\mathbf{x}^*_j = \{x^*_k : \forall k \in N_j \setminus i\}$. In this manner, instead of maximizing over all connected variables $\mathbf{s}_j \setminus x_i$ on factor F_j, we examine for each variable $\forall x_k \in \{\mathbf{s}_j \setminus x_i\}$ whether to: (a) include variable x_k in the maximization operation; or instead (b) use directly the existing value x^*_k based on the broadcasted information.

This choice depends on the relative time-estimate of agent i with respect to its connected agents (through a factor j). We first consider factors $F_j(x_i, x_k)$ connecting exactly two agents i, k. At time-step t, both agents have broadcasted their time-estimates $t_{i,e}, t_{k,e}$ and current variable assignments x^*_i, x^*_k. From the perspective of agent i, its $r^t_{j\to i}$ messages will be updated as:

$$r^t_{j\to i}(x_i) = \begin{cases} \max_{x_k} \left[F^t_j(x_i, x_k) + q^{t-1}_{k\to j}(x_k) \right], & \text{if } t_{k,e} - t_{i,e} \le t_e \\ F^t_j(x_i, x^*_k) + q^{t-1}_{k\to j}(x^*_k), & \text{otherwise} \end{cases} \tag{3}$$

where t_e is a positive constant accounting for the reaction time of the underlying D-DCOP when agents update their assignments.[3] To put it simply, if agent i plans to update its assignment x^*_i before k does, then the implicit assumption on k's update through the $\max_{x_k}$ operator will be less accurate than directly embedding the broadcasted assignment x^*_k for the calculation. The following example containing two agents connected with a factor illustrates this aspect and the overall reasoning for the approach.

Example 3.1 (Two agents in lane-free traffic connected with a factor). Consider two lane-free agents i, k connected with a factor $F_j(x_i, x_k)$ at time $t = 0s$. The variables control the vehicles' lateral alignment, and the factor F_j aims to coordinate them so that the vehicle on the back can overtake if it desires to (see more details in Sec. 4). In this example, agent i wishes to overtake and factor F_j incorporates this information. At time $t = 0s$, the last variable assignments x^*_i, x^*_k and shared time estimates $t_{i,e}, t_{k,e}$ are shown in Fig. 3 at the top segment, and the dashed lines crossing the center of each vehicle showcase where the desired lateral positioning is according to x^*_i, x^*_k respectively. For the sake of simplicity, we set $t_e = 0$ in this example and consider that the time-estimates shared by agents will be fully accurate. Additionally, for reasons of compactness in the figure, we define $FQ_j(x_i, x_k) = F_j(x_i, x_k) + q_{k\to j}(x_k)$.

Focusing at the top segment, agent i has an earlier time-estimate for its update, meaning that it should not maximize over x_k for its $r^t_{j\to i}(x_i)$, since at that time, agent k will remain in its current lateral position. Contrariwise, k plans to update its variable much later, therefore it is more rational from its end to maximize over variable x_i in order to reach a decision based on i's potential movement, and not its current lateral positioning according to x^*_i. Then, at the middle segment the situation is inverted due to the updated time-estimates, and finally at the bottom part, both agents perform the conventional maximization of Max-Sum due to the synchrony of their upcoming reassignment.

As of now, we have only addressed factors connecting 2 agents. Notably, factors with only 1 agent are a special case and do not require this procedure. However, the same notion can be directly applied in larger factors as well, albeit with a compact form of Eq. 3. For this, we revise Eq. 3 in a way that combines the two cases as: $r^t_{j\to i}(x_i) = \max_{s^{t,e}_j} [F^t_j(x_i, x_k) + q^{t-1}_{k\to j}(x_k)]$, where $s^{t,e}_j = \{x_k : t_{k,e} - t_{i,e} \le t_e\}$. Following this, we can directly generalize for factors of various sizes accordingly:

$$r^t_{j\to i}(x_i) = \max_{\mathbf{s}^{t,e}_j} \left[F^t_j(\mathbf{s}_j) + \sum_{x_k \in N_j \setminus i} q^{t-1}_{j\to k}(x_k) \right] \tag{4}$$

where $\mathbf{s}^{t,e}_j$ contains only the variables in factor j that should be maximized based on the relative time-estimate of agent i:

$$\mathbf{s}^{t,e}_j = \{x_k : \forall k \in N_j \setminus i, t_{k,e} - t_{i,e} \le t_e\} \tag{5}$$

and the input vector $\mathbf{s}_j$ is formed by $\mathbf{s}_j = \{x_i\} \cup \mathbf{s}^{t,e}_j \cup \mathbf{s}^{t-,e}_{j^*}$, where:

$$\mathbf{s}^{t-,e}_{j^*} = \{x^*_k : \forall k \in N_j \setminus i, t_{k,e} - t_{i,e} > t_e\} \tag{6}$$

contains the broadcasted x^*_k assignments for the connected variables that do not comply with the time-estimate criterion, i.e., for all excluded neighboring variables from $s^{t,e}_j$, the broadcasted information $\mathbf{x}^*_j$ is utilized to fill the arguments in $F^t_j(\mathbf{s}_j)$ and $q^{t-1}_{k\to j}(x_k), \forall x_k \in N_j \setminus i$ instead of maximizing over these variables as well. In Appendix A, we discuss the impact of this revised equation and its connections to local search-based methods. The full algorithmic process for the distributed update of each agent per time-step is outlined in Algorithm 1. The steps concerning lines 2,5,6 of the algorithm can depend on additional domain-specific mechanisms that we later establish (see Sec. 4.3) for lane-free traffic.

[3] For instance, the x_i variables control the lateral placement of lane-free vehicles. A small time difference is negligible when accounting for the reaction time of the underlying system.

Algorithm 1 Agent i Distributed Update

Input: Surrounding agents and their previously broadcasted information $\forall j \in M_i \langle q^{t-1}_{\cdot \to j}, \mathbf{x}^*_j, \mathbf{t}_{j,e} \rangle$
Output: Updated broadcasted information $\forall j \in M_i \langle q^t_{i \to j} \rangle, \langle x^*_i, t_{i,e} \rangle$

1: Observe surroundings and broadcasted information
2: Update connected factors' information (for existing factors: change values/remove, or form new connections)
3: Update r^t messages from broadcasted q^{t-1} messages (Eq. 4 for Conditional Max-Sum)
4: Update q^t messages (Eq. 2)
5: Decide whether to update x^*_i
6: Update time-estimate information $t_{i,e}$
7: Broadcast q^t messages for all connected factors, variable assignment x^*_i and time-estimate for next variable update $t_{i,e}$

4 MULTIAGENT COORDINATION IN LANE-FREE TRAFFIC

In this section, we present the FG formulation based on lateral regions that coordinates the vehicles' lateral alignment in lane-free traffic.

4.1 Problem Description

In the examined lane-free traffic environment, each vehicle populating the road operates on a 2-dimensional space consisting of a longitudinal (front/back) and lateral (left/right) axis. The investigated scenarios consist of a lane-free highway that is either static (contains a specific set of vehicles) or an open environment (new vehicles constantly enter the highway). Each vehicle possesses a separate *desired speed* v^d_i objective that pursues, consequently resulting in many instances where vehicles wish to overtake while being surrounded by nearby traffic in a lane-free setting. The underlying policy of the vehicles is a rule-based method that automatically adjusts their acceleration in response to surrounding traffic. As such, the vehicles have by default a "reactive" behaviour that follows safety rules in order to avoid collisions, and do not take strategic initiatives, i.e., perform overtake maneuvers or give priority to other vehicles. This type of behaviour is handled by our D-DCOP formulation of the problem, where the vehicles need to coordinate their lateral placement y_i in order to perform cooperative maneuvers that benefit their own and/or nearby traffic's objectives. To this end, each vehicle targets a *desired lateral placement* y^d_i through our approach, and they are modelled as agents in a FG structure that evolves over time due to the dynamic nature of the problem; following the distributed communication framework we propose with asynchronous updates of their desired lateral placements for responding timely to their respective local situation.

4.2 Lane-Free Traffic Environments with Dynamic Lateral Regions

The primary tool that enables coordination in a manner suitable for strategic coordination in lane-free traffic is that of *dynamic lateral regions*. With this the vehicles can interpret the observed and/or communicated information from nearby traffic in a way that

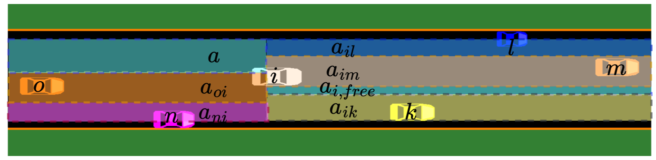

Figure 4: Formed lateral regions with acceleration estimates from the perspective of agent i.

provides a *real-time structured representation* of the environment, and allows them to decide upon their low-level control.

From the perspective of an ego vehicle i, we distinguish between upstream (vehicles on the back of i) and downstream traffic (vehicles on the front of i). Given an observational distance, all vehicles are monitored and the lateral space is accordingly partitioned into lateral regions, as visualized in Fig. 4, which correspond to where the center point of vehicle i can be positioned laterally. At any time, i is located at a specific lateral region, in which its longitudinal behaviour (gas/brake) is being influenced by the front vehicle occupying this region. This is decided according to a car-following method as done typically in lane-based environments, with the vehicle in front as the leader to follow. For this task, we employ the Enhanced Intelligent Driver Model (EIDM) [9], which is an extension of one of the most popular car-following methods, that calculates the longitudinal acceleration a_i of the vehicle, taking into account i's desired speed v^d_i while respecting a time-gap value with the vehicle in front to avoid critical situations. In this manner, given two vehicles (i, j) with j being in front of i, we can calculate the acceleration $a_{i,j}$ of i when j is in front according to EIDM. Likewise, we can calculate this acceleration for each lateral region, meaning we can have an *estimate* of the acceleration of ego vehicle depending on its lateral placement. These acceleration estimates are instrumental in our approach, as they quantify the *value* of residing at a lateral region, and consequently the *benefit* of shifting laterally to a different region by simply comparing the corresponding acceleration evaluations. Vehicles are also influenced by upstream traffic, resulting in nudging behaviour which is an important characteristic of lane-free traffic [14]. The acceleration estimates of all lateral regions for downstream and upstream traffic are integrated in underlying safety rules that regulate the vehicles' control input and therefore their behaviour. A more detailed presentation can be found in Appendix B.

4.3 Factor Graphs in Lane-Free Traffic

We can form a FG of connected vehicles as visualized in Fig. 2, assuming the necessary communication capabilities for vehicle within close proximity. The control variable x_i for each vehicle i is the lateral deviation dy_i which determines the updated *desired lateral alignment* y^d_i of the vehicle. For all calculations relevant to the FG formulation, we examine candidate lateral positions y'_i according to a value for x_i accordingly: $y'_i = y_i + x_i$. Each vehicle can be connected to two types of factors in our formulation. First, the single factor involves only one vehicle and accounts for motivating the vehicle to remain within the road boundaries. Its form is: $F_s(x_i) = -B_c \cdot outOfBounds_i(x_i)$, where the $outOfBounds_i(x_i)$ element yields a negative utility according to the coefficient B_c

if the examined configuration x_i results in a lateral placement y'_i that would exceed the road boundaries. More importantly, the FG formulation contains a second type of a pairwise factor $F_p(x_i, x_j)$ that connects two vehicles i and j, with j preceding i. Its presence serves to motivate both i and j at moving laterally according to i's *desire* to overtake through *regret minimization*. Notably, since the factor affects both i and j's decision due to their involvement, they can accordingly control their lateral behaviour in a coordinated manner. This is accomplished by the following formulation:

$$F_p(x_i, x_j) = regret(x_i, x_j) + comfort(x_i, x_j) \tag{7}$$

where the first term is the calculated *regret* from the perspective of the receding agent i, and has the following form:

$$regret(x_i, x_j) = -R_c \cdot (a_{i,free} - a_{i,j})^2 \cdot overlap_{ij}(x_i, x_j) \tag{8}$$

and the value $a_{i,free}$ is the calculated acceleration of i when located in lateral regions without a leader, meaning that this value only accounts for the desired speed objective of the agent and that $a_{i,free} \geq a_{i,j}$. As such, the difference $(a_{i,free} - a_{i,j})$ expresses the *regret* of agent i for having j *in front* of it,[4] using a positive coefficient R_c and a negative sign to comply with the algorithm's maximization criterion. Note that whenever this type of factor connects two vehicles, this regret value is assigned to it only for configurations of x_i, x_j that would result in the agents having their lateral alignments "overlap" at any point during lateral deviation from the current placement y_i, y_j towards the examined one y'_i, y'_j. More details can be found in Appendix C.1.

The second term of Eq. 7 serves to mitigate unnecessary lateral deviations through a *comfort* utility with the form: $comfort(x_i, x_j) = -C_c \cdot \left(|x_i| + |x_j|\right)$ with C_c as a coefficient. The tuning of R_c, C_c regulates the behaviour of agents regarding overtaking and comfortable driving. A similar idea for comfort can be found on prior work in Proactive DCOPs [5] and RS-DPOP [17], where authors contain an additional cost term that penalizes abrupt and unnecessary changes in variables' values. We instead consider this as a domain-specific aspect of our approach, and do not embed it within the algorithm.

Each agent is associated with one single factor, as shown in Fig. 2. A pairwise factor $F_p(x_i, x_j)$ between agents i, j is considered according to their proximity. Additionally, each agent i prunes its own connections with others in order to conform to upper limits regarding the number of pairwise connections it can have downstream and upstream, respectively. This process is part of line 2 in Alg. 1, with more details in Appendix C.2.

4.4 Asynchronous Decision Updates in Lane-Free Traffic

With the FG formulation above, we have the necessary components to model the lane-free traffic environment as a DCOP. However, additional elements need to be prescribed for asynchronous decision-making as discussed in Sec. 3.1. Agents update the values of all connected factors according to their real-time observation of nearby vehicles and the information from the constructed lateral regions, but they also take into account the broadcasted assignments $\mathbf{x}^*_j$ of all factors j connecting them to other vehicles. The information for the lateral placement of vehicles is informed by these assignments so that the FG formulation integrates these in the lateral positioning y_i, y_j for the calculation of the factor's values. This is an important nuance of the formulation, as the vehicles (through the DCOP formulation) can proactively "argue" regarding their lateral alignment decisions and not take into account intermediate states while they maneuver from one lateral alignment to the next.

At each time-step, all agents independently decide (line 5 in Alg. 1) whether to update their variable assignment x^*_i based on independent time-windows $[T_{min}, T_{max}]$. Once T_{min} time has passed since the agent's last reassignment, only then i examines whether it has reached the selected lateral positioning y^d_i within a small error y_e. If so, then it updates x^*_i based on the received messages r^t. This update can also take place without the aforementioned condition being satisfied if the time-window is exhausted, i.e., T_{max} time has passed since the agent's previous update. Finally, each agent needs to provide a time-estimate for the next update in order to comply with the proposed communication scheme (line 6 in Alg. 1). This is done in a very straightforward manner by computing the remaining time-steps to reach the selected lateral positioning y^d_i. The lateral maneuver of the vehicle solely depends on the use of the movement dynamics and is deterministic. Thereby, the future lateral trajectory of the vehicle can be fully predicted (see Appendix C.3 for more details). Of course, the intended lateral goal of the agent can be compromised in practice by other agents blocking its path towards it, due to safety rules for lateral alignment, as mentioned in Sec. 4.2. Consequently, the shared time-estimates cannot be fully accurate in a realistic environment under uncertainty, hence the reason they are updated at every time-step.

5 EXPERIMENTAL EVALUATION

We empirically evaluate our approach by comparing 3 different Max-Sum variants in the proposed distributed framework with asynchronous variable reassignments, along with a baseline heuristic method without communication among agents. Specifically, we have: (a) **Max-Sum**: the standard Max-Sum algorithm; (b) **No-Max-Sum**[5]: we rely solely on the broadcasted assignments from neighbors instead of maximizing in Eq. 4, i.e., $\mathbf{s}^{t,e}_j$ is empty; (c) **Cond-Max-Sum**: Conditional Max-Sum, as in Sec. 3; and finally (d) **MOBIL** (baseline): Rule-based method based on the popular lane-change model MOBIL [8] that does not employ any communication among agents (more information in Appendix D.1).

Evaluation metrics for experiments contain: (a) the **average speed** of all vehicles throughout the simulation: v^{avg}_x in meters per second (m/s). [6] This metric indicates the efficiency of the vehicles' movement, with higher values suggesting more efficient behaviour, given that the desired speed goals of vehicles are not lower on average than the speed measurements. Additionally, an integral metric we include is: (b) the **average speed deviation** of all vehicles v^{avg}_{dev} in meters per second m/s. This metric effectively measures the vehicles' deviation $|v^x_i - v^d_i|$ of their current speed v^x_i from their desired speed v^d_i objective, and consequently how close vehicles are towards their respective desired speed goal. Moreover, we measure:

[4] i observing j as leader for the examined configuration y'_i, y'_j.

[5] Reflects simpler local search-based methods, see Appendix A.

[6] Note that for the examined range $[25, 35]m/s = [90, 126]km/h$.

Table 1: Results for the coordination problem and the two flow configurations on the $2km$ open highway.

Coord. Problem	v_x^{avg} (m/s)	v_{dev}^{avg} (m/s)	j_y^{avg} (m/s^3)	$TTS(h)$
Max-Sum	25.02	2.00	128.4e−03	-
No-Max-Sum	24.81	2.32	**93.1e−03**	-
Cond-Max-Sum	**25.18**	**1.48**	98.6e−03	-
MOBIL (Baseline)	24.80	2.53	176.8e−03	-
Flow:10000 veh/h	v_x^{avg} (m/s)	v_{dev}^{avg} (m/s)	j_y^{avg} (m/s^3)	TTS (h)
Max-Sum	29.04	1.71	145.0e−3	191.71
No-Max-Sum	29.00	1.74	**98.5e−3**	191.96
Cond-Max-Sum	**29.09**	**1.64**	127.8e−3	**191.39**
MOBIL (Baseline)	28.68	2.13	162.5e−3	193.94
Flow:15000 veh/h	v_x^{avg} (m/s)	v_{dev}^{avg} (m/s)	j_y^{avg} (m/s^3)	TTS (h)
Max-Sum	28.42	2.22	86.4e−3	293.61
No-Max-Sum	28.36	2.29	**62.0e−3**	294.19
Cond-Max-Sum	**28.44**	**2.21**	71.3e−3	**293.41**
MOBIL (Baseline)	28.04	2.63	87.2e−3	297.37

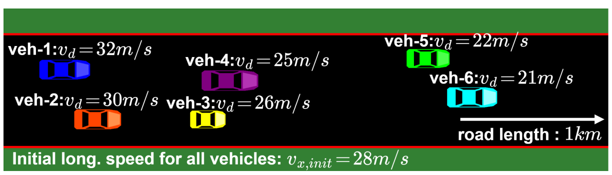

Figure 5: Initial placement of vehicles for the lane-free coordination problem.

(c) the **average *jerk*** of vehicles in m/s^3 to be minimized, which is the derivative of acceleration m/s^2, and a commonly used metric for discomfort of passengers [6]. As we focus on the way vehicles move laterally in the lane-free environment, we show the jerk regarding lateral maneuvers j_y^{avg}. Finally, a system-level measurement is included for Sec. 5.2: (d) the **total-time-spent** (TTS) hours (h), a standard metric in transportation that depicts the accumulated travel time summed over all vehicles in simulation.[7]

5.1 Lane-Free Coordination Problem

The first type of environment we examine in order to gain empirical insights for our approach is the small experiment with initial placements for vehicles as visualized in Fig. 5. There, we establish three rows where the desired speeds are set so that vehicles on the back need to overtake the vehicles on the front. Each vehicle has a different time-estimate for its own next update, meaning that while the vehicles communicate, each one will update its lateral alignment at a different time-step. The decision regarding the timing of subsequent lateral alignment updates is affected by the proximity of the new selection, resulting again in asynchronous updates due to the independency of agents. More details on initial conditions and parameter tunings can be found in Appendix D.2.

[7]Videos demonstrating the performance of all methods can be found at: https://bit.ly/4k6FXx1.

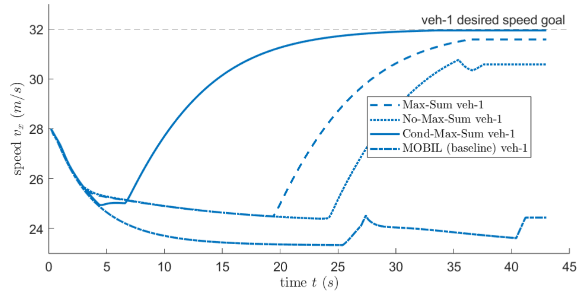

Figure 6: Speed trajectories of agent veh-1 for all examined methods in the lane-free coordination problem.

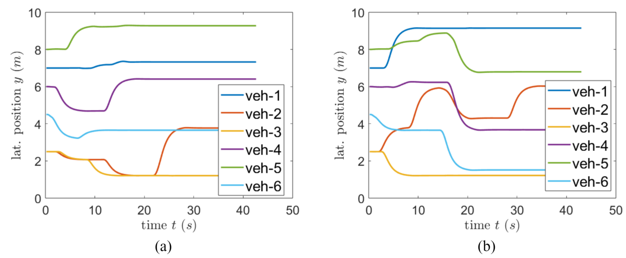

Figure 7: Trajectories of agents' lateral placement for (a) Cond-Max-Sum and (b) Max-Sum, respectively.

Results are presented in Table 1, where we compare all methods using the above-mentioned metrics. A more detailed capture of the performance can be found in the longitudinal speed trajectories of Fig. 6, where we focus on agent veh-1 located at the beginning of the road, showing how fast and to what extend its desired speed goal is accomplished throughout the simulation time. Since veh-1 has the highest desired speed, this information—stemming from the regret minimization term of the pairwise factor—is communicated through the message-passing operation. To put it simply, a vehicle with increased desired speed will result in a heightened regret value when faced with a slower vehicle in front, thereby resulting in a lateral configuration of agents so that veh-1 overtakes. We can certainly pinpoint Cond-Max-Sum as the superior method when compared either to the standard algorithm or the No-Max-Sum case. After the initial time period where veh-1 needs to slow down since other vehicles are in front, it then reacts and coordinates more timely with its surroundings, with its desired speed goal being better accommodated without negatively affecting the goals of other agents, as evident in the v_{dev}^{avg} metric in Table 1. Then, we can see in Fig 7 that Cond-Max-Sum achieves this with fewer lateral maneuvers when compared to the standard algorithm, due to the enhanced message passing. The value of lateral jerk j_y^{avg} directly provides a measure for the passenger discomfort from these lateral maneuvers. Even though the No-Max-Sum case has the lowest value of j_y^{avg}, when combined with the much inferior speed metrics, it shows that vehicles do not properly harness opportunities for lateral coordination that facilitate overtakes. Supplementary results can be found in Appendix D.3.

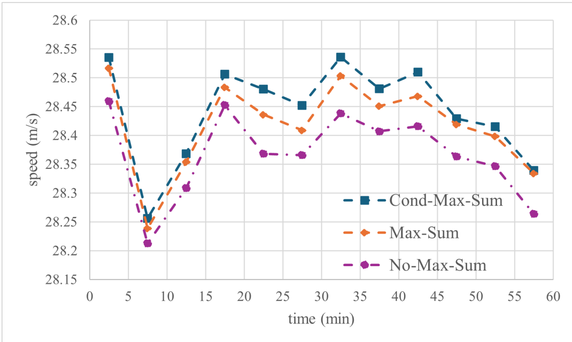

Figure 8: Comparison of measured speed for 5-minute intervals in $15000veh/h$.

5.2 Open and Large Lane-Free Traffic Settings

While experiments in a small environment with a specific set of agents allow us to present a detailed depiction of the efficiency by even looking into vehicles' trajectories, we further investigate the use of our distributed framework and Cond-Max-Sum in a much more demanding large-scale open highway environment. There, new vehicles constantly enter a $2km$ highway's entry point (and are introduced to D-DCOP) while old ones reach the road exit, with traffic flow rates 10000, $15000veh/h$ that result in hundreds of vehicles populating the road at any given time, specifically around 200, $300veh$ respectively. There is an induced uncertainty in this type of environment due to the constant emergence of new vehicles (new variables) and frequent changes regarding vehicles' connections (creation/removal of pairwise factors) due to overtakes. Each vehicle samples a desired speed from a uniform distribution within $[25, 35]m/s$ upon entrance. The environment settings for the open highway can be found in Appendix D.2 and D.4. Then, in D.5 we include measurements on the size of the FG under such settings and a discussion on the communication overhead per agent.

The inclusion of a baseline method (MOBIL) without online communication among vehicles serves to further motivate the proposed framework. As evident in Table 1, all metrics exhibit worse values w.r.t. to any Max-Sum variant, with the exception of jerk (that is close to the standard algorithm for $15000veh/h$), indicating redundant lateral maneuvers that are not necessarily followed by the intended overtakes. This shows the effect of the communication among agents when juxtaposed to only observing nearby traffic.

Regarding the three Max-Sum variants in Table 1, results on the average metrics are consistent with our findings in Sec. 5.1, albeit with a lower margin on the differences between them. This follows from the frequent graph changes in this open environment, which pose an additional challenge compared to the setup in Sec. 5.1 as new pairwise factors can occur unexpectedly and render the calculated messages less compatible under these situations. Yet, Cond-Max-Sum still consistently outperforms the other variants. While the benefit in v_{dev}^{avg} is modest on average, the TTS metric showcases better system level performance. Then, No-Max-Sum—the variant that is akin to simpler local-search methods (cf. Appendix A)—always demonstrates inferior speed, but as a consequence, exhibits lower discomfort levels since agents do not explore mutually beneficial decisions to the same extent. This is also less aligned with the main focus of the factors towards motivating agents to overtake in a coordinated manner whenever desired.

In order to fully capture the efficiency in the large-scale environment and underline this "marginal, but consistent" performance increase for the revised message update of Cond-Max-Sum, we additionally provide Fig. 8, where we divide the $1h$ time span of the simulation to $5min$ intervals and measure the average speed v_x^{avg} of all agents. There, we see how each Max-Sum variant handles the open environment during each time interval of the simulation. Furthermore, in Appendix D.6, we verify that the results in Fig. 8 are statistically significant with $\alpha = 0.05$, and contain a (histogram) plot that shows the percentages calculated from all jerk measurements, divided into bins. For the central bin (ideal case, containing $0m/s^3$), we have measured 90.8%, 89.4% and 87.2% for No-Max-Sum, Cond-Max-Sum and Max-Sum, respectively. With these additional measurements, we can conclude with more confidence that Cond-Max-Sum achieves higher speeds with lower passenger discomfort across all cases, thereby combining two (practically opposing) goals better than the standard Max-Sum update in large-scale settings.

6 CONCLUSIONS AND FUTURE WORK

In this work, we proposed a framework for asynchronous decision making in multiagent environments, along with Conditional Max-Sum for enhanced coordination in these settings. Experimental evaluation in our FG formulation for lane-free traffic exemplifies the applicability of this more realistic framework, along with the enhanced efficiency of Conditional Max-Sum. In future work, a natural extension involves environments with *external* agents, that is, other entities not complying with the DCOP formulation, and thereby introducing uncertainty to the formulation. Existing work [4] already addresses this issue, which can be bundled directly with our approach for a more expansive framework. Moreover, other algorithms (outside of Max-Sum) in the literature [3] could be alternatively considered, albeit with possible correspondent adjustments/extensions (as in this case) to accommodate the introduced flexibility. Finally, in relation to existing work in lane-free traffic, we should point out that to the best of our knowledge this is the only approach that constructs a hybrid system for lane-free environments using safety rules based on dynamic lateral regions for collision avoidance, instead of relying on mathematical models in static environments [7] or soft constraints as part of a utility/cost function [21, 22]. Nevertheless, outside of these qualitative differences, direct comparisons could shed light on other trade-offs.

ACKNOWLEDGMENTS

The research described in this paper was carried out within the framework of the National Recovery and Resilience Plan Greece 2.0, funded by the European Union - NextGenerationEU (Implementation Body: HFRI. Project name: DEEP-REBAYES. HFRI Project Number 15430). Moreover, 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.

REFERENCES

[1] Shushman Choudhury, Jayesh K. Gupta, Peter Morales, and Mykel J. Kochenderfer. 2021. Scalable Anytime Planning for Multi-Agent MDPs. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 341–349.

[2] Alessandro Farinelli, Alex Rogers, Adrian Petcu, and Nick R. Jennings. 2008. Decentralised Coordination of Low-Power Embedded Devices Using the Max-Sum Algorithm. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '08)*. International Foundation for Autonomous Agents and Multiagent Systems, Estoril, Portugal, 639–646.

[3] Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. 2018. Distributed constraint optimization problems and applications: A survey. *Journal of Artificial Intelligence Research* 61 (2018), 623–698.

[4] Pavlos Geronymakis, Dimitrios Troullinos, Georgios Chalkiadakis, and Markos Papageorgiou. 2022. Collaborative Decision Making for Lane-Free Autonomous Driving in the Presence of Uncertainty. In *European Conference on Multi-Agent Systems (EUMAS '22)*. Lecture Notes in Computer Science (LNAI,volume 13442), Düsseldorf, Germany, 171–187.

[5] Khoi D Hoang, Ferdinando Fioretto, Ping Hou, William Yeoh, Makoto Yokoo, and Roie Zivan. 2022. Proactive dynamic distributed constraint optimization problems. *Journal of Artificial Intelligence Research* 74 (2022), 179–225.

[6] Quanan Huang and Huiyi Wang. 2004. *Fundamental study of jerk: evaluation of shift quality and ride comfort*. Technical Report. SAE Technical Paper.

[7] Iasson Karafyllis, Dionysios Theodosis, and Markos Papageorgiou. 2022. Lyapunov-based two-dimensional cruise control of autonomous vehicles on lane-free roads. *Automatica* 145 (2022), 110517.

[8] Arne Kesting, Martin Treiber, and Dirk Helbing. 2007. General lane-changing model MOBIL for car-following models. *Transportation Research Record* 1999, 1 (2007), 86–94.

[9] Arne Kesting, Martin Treiber, and Dirk Helbing. 2010. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368, 1928 (2010), 4585–4605.

[10] Jelle R. Kok and Nikos Vlassis. 2006. Collaborative Multiagent Reinforcement Learning by Payoff Propagation. *Journal of Machine Learning Research* 7, 65 (2006), 1789–1828.

[11] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* 47, 2 (2001), 498–519.

[12] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (Stockholm, Sweden) *(UAI '99)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 467–475.

[13] Duc Thien Nguyen, William Yeoh, Hoong Chuin Lau, Shlomo Zilberstein, and Chongjie Zhang. 2014. Decentralized Multi-Agent Reinforcement Learning in Average-Reward Dynamic DCOPs. *Proceedings of the AAAI Conference on Artificial Intelligence* 28, 1 (2014), 1447–1455.

[14] Markos Papageorgiou, Kyriakos-Simon Mountakis, Iasson Karafyllis, Ioannis Papamichail, and Yibing Wang. 2021. Lane-Free Artificial-Fluid Concept for Vehicular Traffic. *Proc. IEEE* 109, 2 (2021), 114–121.

[15] Arseniy Pertzovsky, Roie Zivan, and Noa Agmon. 2024. Collision Avoiding Max-Sum for Mobile Sensor Teams. *Journal of Artificial Intelligence Research* 79 (2024), 1281–1311.

[16] Adrian Petcu and Boi Faltings. 2006. ODPOP: an algorithm for open/distributed constraint optimization. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1 (AAAI '06)*. AAAI Press, Boston, Massachusetts, 703–708.

[17] Adrian Petcu and Boi Faltings. 2007. Optimal Solution Stability in Dynamic, Distributed Constraint Optimization. In *2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT '07)*. IEEE, Fremont, CA, USA, 321–327.

[18] Sarvapali D. Ramchurn, Alessandro Farinelli, Kathryn S. Macarthur, and Nicholas R. Jennings. 2010. Decentralized Coordination in RoboCup Rescue. *Comput. J.* 53, 9 (2010), 1447–1461.

[19] Alex Rogers, Alessandro Farinelli, Ruben Stranders, and Nick R. Jennings. 2011. Bounded approximate decentralised coordination via the max-sum algorithm. *Artificial Intelligence* 175, 2 (2011), 730–759.

[20] Ruben Stranders, Alessandro Farinelli, Alex Rogers, and Nick R. Jennings. 2009. Decentralised Coordination of Continuously Valued Control Parameters Using the Max-Sum Algorithm. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1 (AAMAS '09)*. International Foundation for Autonomous Agents and Multiagent Systems, Budapest, Hungary, 601–608.

[21] Dimitrios Troullinos, Georgios Chalkiadakis, Vasilis Samoladas, and Markos Papageorgiou. 2022. Max-Sum with Quadtrees for Decentralized Coordination in Continuous Domains. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI '22)*. International Joint Conferences on Artificial Intelligence Organization, Wien, Austria, 518–526.

[22] Venkata Karteek Yanumula, Panagiotis Typaldos, Dimitrios Troullinos, Milad Malekzadeh, Ioannis Papamichail, and Markos Papageorgiou. 2023. Optimal trajectory planning for connected and automated vehicles in lane-free traffic with vehicle nudging. *IEEE Transactions on Intelligent Vehicles* 8, 3 (2023), 2385–2399.

[23] Harel Yedidsion, Roie Zivan, and Alessandro Farinelli. 2018. Applying max-sum to teams of mobile sensing agents. *Engineering Applications of Artificial Intelligence* 71 (2018), 87–99.

[24] Roie Zivan, Ben Rachmut, Omer Perry, and William Yeoh. 2023. Effect of asynchronous execution and imperfect communication on max-sum belief propagation. *Autonomous Agents and Multi-Agent Systems* 37, 2 (2023), 40.