2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

Cooperative Collision Avoidance for Coalitions of Connected and Autonomous Vehicles

Dhruvkumar Patel[1], Rishita Bansal[2], and Rym Zalila-Wenkstern[3]

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922287

***Abstract*— In this paper, we discuss collision avoidance for Connected and Autonomous Vehicles (CAVs) on a highway. CAVs are clustered into coalitions each managed by a leader. Within a coalition, collision avoidance is addressed using a Monte Carlo Tree Search (MCTS)-based approach. We propose algorithms for collision avoidance across coalitions. After an initial assessment of the impact of a potential collision on an affected coalition, leaders cooperate to define action plans that are free of intra-coalition and inter-coalition conflicts. The algorithms were validated through extensive realistic simulations in a multi-agent-based traffic simulator. Experimental results discuss the reliability and scalability of the algorithms for coalitions of different sizes. Moreover, we present an analysis to select the optimal coalition size and the optimal number of coalitions given a total number of CAVs.**

I. Introduction

We see an increased number of accidents with the increased road traffic. Most of these accidents or collisions take place due to human error. Automating the vehicles on the road can help solve this problem. Connected Automated Vehicles (CAVs) are capable of sensing their surroundings using a variety of sensors and communicating with each other to avoid collisions in an emergency situation. In many on-road situations, a CAV may not be able to avoid collisions by acting on its own. Such complex situations call for a cooperative decision making system, in which neighboring CAVs cooperate with each other to avoid potential collisions.

In recent years, a variety of cooperative planning approaches for CAVs have been proposed. Centralized approaches employ a central computing node, which is responsible for detecting and avoiding collisions by computing cooperative actions plans of all on-road CAVs [1], [2]. Centralized approaches deterministically find optimal solutions but are computationally expensive and scale poorly to a large number of CAVs [3]. In decentralized approaches, each CAV from a coalition of CAVs is individually responsible to compute its collision-avoiding action plan. Each CAV generally formulates the action planning problem as a form of Markov Decision Process (MDP), which can be solved using different reinforcement learning techniques such as Monte Carlo Tree Search(MCTS)[4][5]. These approaches work well in simple scenarios with small number of CAVs in the coalition. As the coalition size grows, MCTS algorithm becomes computationally intractable and the collision avoidance success rate also drops down. In our previous works[6][7], we proposed a modified MCTS algorithm and an improved reward function to improve the scalability and the collision avoidance success rate. Decentralized approaches can be effective for even larger number of CAVs if CAVs are grouped into multiple smaller-sized coalitions. However, CAVs in different coalitions must communicate with each other to come to a collision-free consensus across multiple coalitions. This is a complex problem and have not been addressed yet.

We present **Co**operative **Co**llision **A**voidance (CoCoA) algorithm for multiple interacting coalitions of CAVs. CoCoA employs a sequential decision making approach. Leaders of the coalitions in a coalition sequence cooperate to finalize action plans for their coalition members that are free of inter-coalition conflicts. CAVs inside a single coalition use a hierarchical decision making approach as proposed in our previous work[8]. We implement CoCoA in a large scale multi-agent-based traffic simulator to evaluate its scalability and collision avoidance success rate. The experimental results in the simulations prove the application of the work.

In the next section, we review related works. In Section III, we formalize the problem. In Section IV, we give an overview of CoCoA approach. In Section V, we present CoCoA algorithms and in Section VI, we present the experimental results showing applicability of CoCoA to multiple coalitions of CAVs.

II. Related work

Collision avoidance approaches discussed in the literature can be categorized as *centralized* or *decentralized* approaches. In centralized approaches, a central server is responsible for planning of actions for all affected CAVs. In decentralized approaches, each CAV is responsible for planning its individual actions and coordinating with other CAVs to avoid collisions.

A. Centralized Solutions

In centralized collision avoidance approaches, CAVs periodically communicate their respective states to a central node, which can be a vehicle or a server. When the central node detects a potential collision, it uses an optimization method to find action plans for all conflicting CAVs. Different optimization methods have been proposed.

In [9], a single CAV detects the conflicting neighboring CAVs and derives action plans for all conflicting CAVs in a priority order using Mixed Integer Programming (MIP). In [1], authors propose to use a two step method. In Path Planning step, server uses SRRT (Smooth Rapidly exploring

[1]Dhruvkumar Patel, [2]Rishita Bansal,[3]Rym Zalila-Wenkstern are with Department of Computer Science, University of Texas at Dallas, 800 W Campbell Rd, Richardson, Texas 75080. dhruv@utdallas.edu, rishita@utdallas.edu, rymw@utdallas.edu

Random Trees) algorithm to find non-colliding trajectory of each CAV. Server then uses Motion Planning step to maximize the speed of each CAV along the computed trajectories. In [10], authors propose to use Mixed Integer Quadratic Programming(MIQP) along with collision constraints to find non-colliding CAV trajectories. In [11], authors propose to decompose the centralized Quadratic Programming problem using Alternating Direction Method of Multipliers(ADMM). The decomposed optimization objective is then solved by on a cluster of computing nodes to take advantage of parallel processing. In [2], authors formulate the CAV trajectory calculation problem as an Integer Programming problem with collision constraints that is solved using a commercial SAT solver.

Main limitation of centralized approaches is their scalability as they cannot handle a large number of vehicles with a fixed amount of centralized computational resource. Since central server alone is responsible for planning actions of all CAVs, the whole system fails in case the central server fails or is unreachable.

B. Decentralized Solutions

Decentralized approaches are generally coalition-based. When a collision is detected by a coalition member, each CAV in the coalition cooperates with other coalition members to plan its individual actions by taking into account their current positions and future actions. Decentralized approaches can be classified into single coalition approaches and multiple coalitions approaches.

1) Single Coalition: Single coalition approaches group all neighboring CAVs participating in the collision avoidance decision-making process together in a single group. In some single coalition approaches, CAVs in the coalition cooperate with each other directly by communicating their action plans with each other. In other approaches, CAVs use indirect cooperation using sensors to estimate other CAV states and actions.

In [12], authors represent a set of CAVs driving on a multi-lane road as nodes on a distributed graph. Each CAV is represented using a potential field function. CAVs use gradient descent method to compute non-colliding trajectories. This approach uses indirect cooperation and is limited to simplistic collision avoidance scenarios. In [13], authors propose the desired versus planned trajectory (DVP) approach. Each CAV derives and broadcasts two trajectories: a planned trajectory that it is currently following and the desired trajectory that it wants to follow. When a different CAV receives this information, it checks and replans if it can accommodate the first CAV's desired trajectory and broadcasts its updated trajectory. Although this approach is successful for two CAVs in simple conflicting situations, it is not scalable to large number of CAVs as the algorithm takes many replanning steps for each CAV to converge. In [14], authors propose a coalition-based leader-follower approach to improve computational resource utilization. Leader CAV executes an actor-critic method known as Co-DDPG to learn parameters of CAV state-action policy and communicates these parameters with the follower CAVs. All coalition members use the learned state-action policy to select collision-avoiding actions. This method achieves indirect cooperation through the use of a reward function. This method has only been tested with up to three cooperative CAVs scenarios. Monte Carlo Tree Search is a very common approach used for cooperative decision making [4], [15], [5], [16]. In MCTS-based approaches, each CAV iteratively constructs an MCTS tree where each tree node represents states of coalition members and each tree edge represents actions of coalition members. Each CAV ranks the tree edges and picks the highest-ranked edge representing the coalition action plan. Each CAV executes its individual action from the selected coalition action plan. Indirect cooperation is achieved by considering rewards for all coalition members in the reward function. The main limitation of most MCTS-based approaches is scalability as the tree branching factor grows exponentially with the number of CAVs. Moreover, methods with indirect cooperation require solutions computed by all CAVs in isolation to be conflict-free. This is not always guaranteed and may lead to collisions. In our previous work [6], we proposed a hierarchically decentralized algorithm that improves the scalability of MCTS by reducing the branching factor of the MCTS tree exponentially. CAVs in the coalition use direct cooperation to ensure that their selected action plans are collision-free. This approach works well for a single coalition of CAVs, however the application to multiple coalitions of CAVs is not discussed.

2) Multiple Coalitions: Single coalitions algorithms work well for a small number of CAVs, but do not scale well to a large coalition size. It is more practical to group CAVs into multiple interacting coalitions that cooperate with each other at coalition level. However, collision avoidance problem for multiple coalitions of CAVs have not been addressed yet at the algorithm level.

In this paper, we present **Co**operative **Co**llision **A**voidance (CoCoA) algorithm for multiple coalitions of CAVs. We propose a hierarchical and sequential decision-making approach where the inter-coalition decision-making is performed sequentially by each coalition along the coalition sequence. Hierarchical approach is used for the intra-coalition decision making. The unique contributions of our approach are:

- We propose the first cooperative collision avoidance algorithm that works across multiple coalitions of CAVs to our best knowledge.
- We propose a sequential decision-making approach for multiple coalitions of CAVs that builds upon our previously proposed hierarchical decision-making approach for a single coalition using an improved MCTS algorithm[6].
- We conducted extensive realistic experiments in the multi-agent-based simulator to evaluate the reliability and the scalability of CoCoA algorithms for multiple coalitions of CAVs. Additionally, we prove inapplicability of the current state-of-the-art algorithms
- We justified the need for multiple coalitions structure for an efficient collision avoidance through experimental

analysis.

III. Model Definition

We assume that a set of n CAVs are navigating on a straight highway. These CAVs are distributed in a sequence of coalitions denoted by $\mathcal{N}$. Each coalition is identified by its ID $i \in \mathcal{N}$. In each coalition i, one coalition member is assigned the role of a leader and is responsible for the coalition management. Each coalition $i \in \mathcal{N}$ is formally defined by a coalition state vector $C^i = \{l^i, s^i, M^i, \psi^i\}$ where l^i is the coalition leader, s^i is coalition size, M^i is a set of member CAVs, and ψ^i is a set of neighbor coalitions. Coalition formation and leader assignment algorithms are adapted from our previous work [17].

A member CAV of coalition i can be identified using its own ID $j \in M^i$ and is defined using a CAV state vector $V^j = \{p^j, v^j, lane^j, \theta^j, x^j, A^j\}$ where p^j is the CAV's position, v^j is its velocity, $lane^j$ is its current lane, θ^j is its orientation, x^j is the size of CAV containing length and width values, and A^j is the action set of CAV j available in a given state considering road boundaries. We consider five primitive actions in each CAV's action set notably fixed acceleration, fixed deceleration, maintain speed, change lane to the left lane, and change lane to the right lane.

A mitigation action plan α_h^j is a sequence of h actions performed to avoid collisions by CAV j in case of perceived danger or warning related to a misbehaving vehicle. It is defined as $\alpha_h^j = \{a_{t_k}^j\}_{k=1}^h$ where $a^j \in A^j$ is a single CAV primitive action from its action set and t_k is the start of the execution of the mitigation plan's k'th action. h is known as the planning horizon. Although individual CAV's primitive action set is small, each CAV has many more number of action plans options. We define the coalition mitigation action plan α_h^i for coalition i as a set of individual CAV mitigation action plans, one for each coalition member $j \in M^i$, i.e, $\alpha_h^i = \{\alpha_h^j | j \in M^i\}$.

We require that all coalitions in set $\mathcal{N}$ are navigating in a non-overlapping sequence on the highway, meaning that no two CAVs from two different coalitions are navigating laterally together in different lanes. This assumption allows us to define the definitions of preceding and succeeding coalitions. A *preceding coalition* or a *predecessor* of the coalition i is the coalition that is navigating ahead of the coalition i and is denoted by $pred(i)$. A *succeeding coalition* or a *successor* of the coalition i is the coalition that is navigating after the coalition i and is denoted by $succ(i)$. We call the coalition that is directly affected by the misbehaving vehicle a *primary coalition* in the decision-making context. Coalitions other than the primary coalition are called *secondary coalitions*. We assume that the primary coalition is always located at either the beginning or at the end of the coalition sequence.

IV. General Approach

The decision making problem of a single coalition i is formulated as a Multi-agent Markov Decision Process (MMDP) [18]. Unlike decentralized planning with implicit cooperation using the reward function [5], our approach uses explicit V2V communication to coordinate between all members of a coalition. The solution to an MMDP is given by state-action value function Q^*, which can be used to select an optimal CAV action a^j in a given CAV state V^j. In the first step, Member CAVs of coalition i execute a modified version of Monte Carlo Tree Search (MCTS) algorithm to approximate optimal Q^* values. Using the approximated Q^* values, each CAV j selects a set of ranked individual action plans $\{\alpha_h^j\}$. In the second step, coalition leader l^i receives the sets of ranked individual CAV action plans $\{\{\alpha_h^j\}, \forall j \in M^i\}$ and selects top three non-conflicting coalition action plans ranked using Q^* values. In the regular MCTS algorithm, each CAV iteratively constructs an MCTS tree, which can grow exponentially large with the coalition size and limits the scalability of MCTS. In our modified MCTS algorithm, each CAV constructs an MCTS tree with a reduced branching factor that is not dependent on the coalition size to improve its scalability.

Our general approach for the multi-coalition collision avoidance decision-making works in a sequential manner starting from the primary coalition along the coalition sequence $\mathcal{N}$. In the sequential decision-making approach, the task of choosing the final coalition action plan for any coalition is performed by the next neighboring coalition in the coalition sequence $\mathcal{N}$. Our sequential decision-making approach uses two main algorithms: a primary coalition algorithm, and a secondary coalition algorithm. When a CAV from the primary coalition detects the misbehaving vehicle, the coalition leader of the primary coalition uses the primary coalition algorithm to select the top three coalition action plans according to Q^* values. Since the primary coalition is located either at the beginning or at the end of the coalition sequence $\mathcal{N}$, it only has one neighboring secondary coalition. The primary coalition sends the three action plans to the coalition leader of its neighboring secondary coalition. The secondary coalition leader uses the secondary coalition algorithm to choose one of the three action plans as the final plan for the primary coalition and sends it back to the primary coalition leader. The secondary coalition algorithm also generates three coalition action plans for the secondary coalition itself, which are then forwarded to the next secondary coalition leader for the final plan selection. This process is repeated until we reach the last secondary coalition in the coalition sequence $\mathcal{N}$. The last secondary coalition uses the secondary coalition algorithm to generate a single best coalition action plan for itself and sends it to its members.

V. Algorithms

A. Primary coalition algorithm

Without any loss of generality, assume that the primary coalition i is the last coalition in the sequence $\mathcal{N}$. When a CAV of the coalition i is affected by the misbehaving vehicle m, it sends an alert to all members of i. Upon receiving the alert, each member j of i executes the modified MCTS algorithm for a single coalition to estimate Q^* values for each CAV action in the MCTS tree. Each CAV shares its

Algorithm 1 Primary coalition algorithm

Output: $\alpha_h^j, \forall j \in M^i$
1: $T \leftarrow$ ReceiveMCTSTrees()
2: $currentNodes \leftarrow$ null
3: **for** j in M^i **do**
4: $\quad currentNodes(j) \leftarrow T(j).n_\mu$
5: **end for**
6: $\{\alpha_h^i\} \leftarrow$ RecursivePlansSelection($currentNodes$)
7: $pred(i) \leftarrow$ getPredecessorCoalition()
8: $succ(i) \leftarrow$ getSuccessorCoalition()
9: **if** $pred(i) \neq$ null **then**
10: $\quad$ sendPlans($pred(i), \{\alpha_h^i\}$)
11: $\quad \alpha_h^i \leftarrow$ receiveFinalPlan($pred(i)$)
12: **else**
13: $\quad$ **if** $succ(i) \neq$ null **then**
14: $\quad\quad$ sendPlans($succ(i), \{\alpha_h^i\}$)
15: $\quad\quad \alpha_h^i \leftarrow$ receiveFinalPlan($succ(i)$)
16: $\quad$ **else**
17: $\quad\quad \alpha_h^i \leftarrow \{\alpha_h^i\}[1]$
18: $\quad$ **end if**
19: **end if**
20: **for** j in M^i **do**
21: $\quad \alpha_h^j \leftarrow \alpha_h^i(j)$
22: **end for**

MCTS tree with the coalition leader l^i. Each path from the tree's root node to the leaf node represents an individual mitigation action plan prioritized by Q^* values. The coalition leader receives the individual mitigation action plans in form of MCTS trees from all members in step 1 and selects a set of top three coalition action plans α_h^i ranked using Q^* values in step 6 of the Algorithm 1. The primary coalition leader l^i sends the top three plans ranked using Q^* values to its preceding coalition $pred(i)$'s leader $l^{pred(i)}$ in step 10. The task of choosing the final coalition plan for i is left to $l^{pred(i)}$, since the final plan can affect the CAVs of $pred(i)$. When l^i receives the final action plan α_h^i from $l^{pred(i)}$ in step 11, it extracts individual plans from this final plan in steps 20-22 and sends it to its member CAVs for execution.

B. Secondary coalition algorithm

We now describe the algorithm for secondary coalitions, given in Algorithm 2. Without loss of generality, assume that a leader CAV l^i of the secondary coalition i received a set of top three coalition action plans from its succeeding coalition leader $l^{i'}$. Upon receiving the set of coalition action plans, l^i determines the impact for all i''s action plans (step 2-4 in Algorithm 2) and chooses the one which has the least impact on the members of i (step 5). The impact is defined as the number of coalition members of secondary coalition that will have collision with the members of the succeeding coalition if the succeeding coalition follows the action plan under consideration. l^i sends the chosen plan $\alpha_h^{i'}$ back to $l^{i'}$ as the final coalition action plan for i'. l^i also sends the message to all its members to derive their individual mitigation action plans and stores them (step 6). This message includes i''s

Algorithm 2 Secondary coalition algorithm

Input: $\{\alpha_h^{i'}\}$
Output: $\alpha_h^j, \forall j \in M^(i)$
1: Initialize $impactFactors$ to a null map
2: **for** i in 1,2,3 **do**
3: $\quad impactFactors(i) \leftarrow$ computeImpact($\{\alpha_h^{i'}\}[i]$)
4: **end for**
5: $\alpha_h^{i'} \leftarrow$ pickMinImpactPlan($\{\alpha_h^{i'}\}, impactFactors$)
6: sendToCoalition($i', \alpha_h^{i'}$)
7: $T \leftarrow$ ReceiveMCTSTrees()
8: $currentNodes \leftarrow$ null
9: **for** j in M^i **do**
10: $\quad currentNodes(j) \leftarrow T(j).n_\mu$
11: **end for**
12: $\{\alpha_h^i\} \leftarrow$ RecursivePlansSelection($currentNodes$)
13: $pred(i) \leftarrow$ getPredecessorCoalition()
14: $succ(i) \leftarrow$ getSuccessorCoalition()
15: **if** $i' = pred(i)$ **then**
16: $\quad$ **if** $succ(i) \neq$ null **then**
17: $\quad\quad$ sendPlans($succ(i), \{\alpha_h^i\}$)
18: $\quad\quad \alpha_h^i \leftarrow$ receiveFinalPlan($succ(i)$)
19: $\quad$ **else**
20: $\quad\quad \alpha_h^i \leftarrow \{\alpha_h^i\}[1]$
21: $\quad$ **end if**
22: **else**
23: $\quad$ **if** $pred(i) \neq$ null **then**
24: $\quad\quad$ sendPlans($pred(i), \{\alpha_h^i\}$)
25: $\quad\quad \alpha_h^i \leftarrow$ receiveFinalPlan($pred(i)$)
26: $\quad$ **else**
27: $\quad\quad \alpha_h^i \leftarrow \{\alpha_h^i\}[1]$
28: $\quad$ **end if**
29: **end if**
30: **for** j in M^i **do**
31: $\quad \alpha_h^j \leftarrow \alpha_h^i(j)$
32: **end for**

plan $\alpha_h^{i'}$ too as it affects the members of i. Upon receiving this message, each CAV j of the coalition i executes the single coalition algorithm to derive its set of individual action plans α_h^j. Each member sends its set of individual action plans to the leader l^i. Upon receiving the plans, l^i selects a set of top three non-colliding coalition action plans α_h^i. Since the primary coalition was the successor of i, it checks if there is any preceding coalition. If there is a preceding coalition to i, l^i sends the top three plans to its preceding coalition's leader (in steps 24-25), which is responsible for choosing the final plan for i. If there is no preceding coalition left, l^i chooses the best coalition action plan α_h^i according to Q^* values (in step 27). After receiving the final action plan or deciding it by itself, l^i extracts individual CAV plans and sends them to its members for execution in steps 30-32.

VI. Experimental evaluation

In this section, we evaluate the proposed CoCoA algorithms by implementing them in a multi-agent-based simulator. We evaluate the reliability and the scalability of our

multi-coalition approach. We define reliability as a collision avoidance algorithm's ability to find a non-colliding coalition action plan in fixed computational processing time and scalability as the maximum number of coalitions for which the algorithm can complete its execution in fixed computational processing time. We also implement current state-of-the-art single-coalition approaches including centralized algorithm [2], the decentralized algorithm DeCoC-MCTS [15] and the hierarchically decentralized algorithm ISM [8] to show their inapplicability to a multi-coalition setting. Additionally, we perform an analysis to select optimal values of the coalition size and the number of coalitions on the road for a fixed total number of CAVs [1].

A. Multi-agent-based simulator

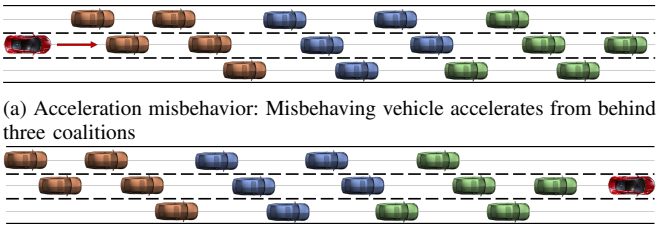
We created a CAV model and tested the algorithms in a microscopic multi-agent based traffic simulator. We implemented the virtual CAV as an autonomous decision-making agent. The virtual CAV has the capability to dynamically sense its surroundings through simulated sensors and to communicate with other virtual CAVs through simulated V2V communication. During the simulation, a CAV's properties (e.g., sensor range) and behavior (e.g., speed) can be modified and the outcome can be witnessed in simulated real-time.

B. Simulation Experimental Setting

In the virtual simulation environment, the unit of time is called $cycle$, and the unit of length is simply called $unit$. In our multi-coalition simulation experiments, a sequence of coalitions is navigating on a three-lane highway. Each CAV member in each coalition of the coalition sequence is cruising at the speed of 2 units/cycle. During the simulation experiment, we add a misbehaving virtual CAV to the simulation environment that misbehaves to cause a potential collision with one or more coalition CAVs. When an affected CAV detects the misbehaving vehicle through one of its simulated sensors, it sends an alert to all of its coalition members. The coalition containing the affected CAV is known as the primary coalition. In our experiments, the primary coalition is located at either end of the coalition sequence. In our experiments, we use the planning horizon $h = 60$ cycles, the Time To Collide (TTC) value $TTC = 20$ cycles, and the action duration $\delta t = 10$ cycles. Each coalition deliberates to choose a coalition action plan containing 6 consecutive actions for each of its coalition members. We perform three types of experimental evaluation: reliability evaluation, scalability evaluation, and trade-off analysis.

In the reliability evaluation experiments, we consider three coalitions in the coalition sequence. Each coalition consists of 5 coalition members. We consider two types of misbehaviors by the misbehaving vehicle (see Figure 1) as listed below:

[1]Demo videos are available at:
https://www.utdmavs.org/itsc2022/



(a) Acceleration misbehavior: Misbehaving vehicle accelerates from behind three coalitions

(b) Full stop misbehavior: Misbehaving vehicle breaks in front of three coalitions

Fig. 1: Misbehavior types for multi-coalition reliability evaluation

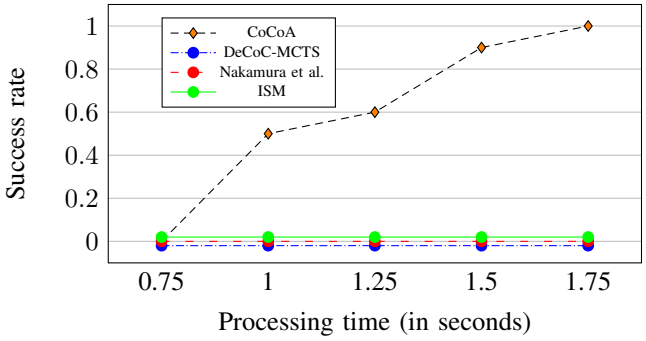


Fig. 2: Reliability evaluation: Acceleration misbehavior

- Acceleration misbehavior- The misbehaving vehicle is positioned behind the primary coalition in the middle lane and speeds at 5 units/cycle in a straight path.
- Full stop misbehavior- The misbehaving vehicle is positioned in front of the primary coalition in the middle lane and comes to a full stop at 0 units/cycle.

C. Reliability Evaluation

Here, we define reliability as a collision avoidance algorithm's ability to find a non-colliding coalition action plan for each of the three coalitions in the given computational processing time. We consider three coalitions, each consisting of five CAVs, and two misbehavior types as shown in Figure 1. For each misbehavior type, we set the processing time to different values and find the collision avoidance success rate for 10 simulation experiments per data

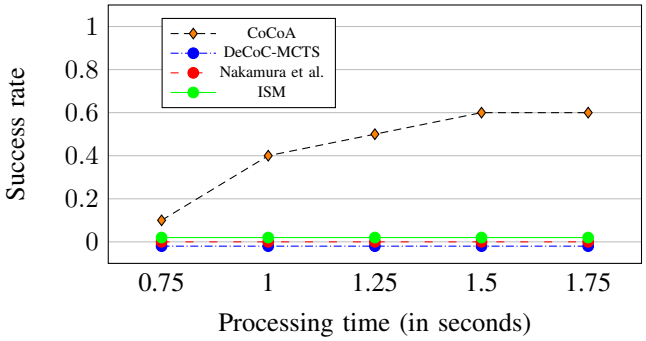


Fig. 3: Reliability evaluation: Break misbehavior

point per algorithm, totalling 200 simulation experiments for each scenario. We plot the collision avoidance success rate versus the processing time for both misbehavior types in Figures 2 and 3. For CoCoA algorithm, as we increase the allowed computation time, the collision avoidance success rate gradually increases demonstrating the reliability of our approach. For single coalition approaches, we consider a single coalition of 15 CAVs instead of three coalitions of five CAVs each, as these algorithms do not work for multiple coalitions. Still the success rate remains zero for these algorithms, as these approaches are not scalable to a large coalition size of 15. This shows CoCoA algorithm's reliability in achieving higher collision avoidance success rate for larger number of CAVs than current single coalition algorithms. Our previous work[6] shows comparison of non-zero success rates of our single-coalition approach with the state-of-the-art single-coalition algorithms shown here for a smaller number of CAVs.

D. Scalability Evaluation

We define scalability for multi-coalition decision making as the maximum number of coalitions for which the algorithm can complete its execution in a reasonable amount of time. In the scalability experiments, we fix the number of CAVs in each coalition to 5 and vary number of coalitions on the road from one to up to five coalitions. We chose the acceleration misbehavior type to test the scalability of our multi-coalition approach. We expect the break misbehavior type to achieve the similar scalability results as the scalability of our approach is not scenario-dependent and that s what we have plotted in Figure 4. We increase the allowed computation time to 10 seconds as the purpose is to evaluate the computational efficiency of our algorithm. Since the multi-coalition decision-making algorithm is sequential in nature, each coalition can start its execution of the decision-making algorithm only after the previous coalition has finished. For this reason, each coalition gets the allowed computation time in seconds that equals to 10 divided by the number of coalitions $|\mathcal{N}|$ in the coalition sequence. All CAVs in a given coalition perform decision-making in parallel and independent of each other for full $10/|\mathcal{N}|$ seconds. For single coalition algorithms, full 10 seconds are used by each single coalition decision-making algorithm. Experimental results show that our multi-coalition algorithm found non-colliding coalition action plans for up to 4 coalitions in the coalition sequence, where each coalition consisted of five CAVs. It means that a total number of 20 CAVs can perform collision avoidance decision-making in 10 seconds of processing time. It is a significant improvement over the highest limit of the single coalition algorithms, which support up to 14 CAVs (see Figure 4).

E. Trade-off Analysis

We perform the trade-off analysis between the number of coalitions and the individual coalition size for a fixed total number of CAVs on the road. For a fixed total number of CAVs, we can reduce each individual coalition size to increase the number of coalitions and vice versa. Our objective is to derive optimal values for the individual coalition size and the number of coalitions for a fixed number of CAVs and argue that there should be a balance between the coalition size and the number of coalitions. We fix the total number of CAVs to 12 (See Figure 5). We generate several different coalition configurations including 12 coalitions of 1 CAV each, 6 coalitions of 2 CAVs each, 4 coalitions of 3 CAVs each, 3 coalitions of 4 CAVs each, 2 coalitions of 6 CAVs each, and a single coalition of 12 CAVs. For each of these coalition configurations, we perform 10 simulations experiments and note the collision avoidance success rate. We fix the allowed computation time to 3.6 seconds. For the sequential decision-making algorithm used for the multi-coalitions, 3.6 seconds computation time gets divided among the coalitions. For 12 coalitions, each coalition gets 0.3 seconds of the computation time. Whereas for 2 coalitions, each coalition gets 1.8 seconds of the computation time.

We can see that 4 coalitions with 3 CAVs each configuration yields the highest success rate (see Figure 6). 3 coalitions with 4 CAVs each and 2 coalitions with 6 CAVs each configurations also have very good success rate values. However, very small coalition sizes of 1 and 2, as well as

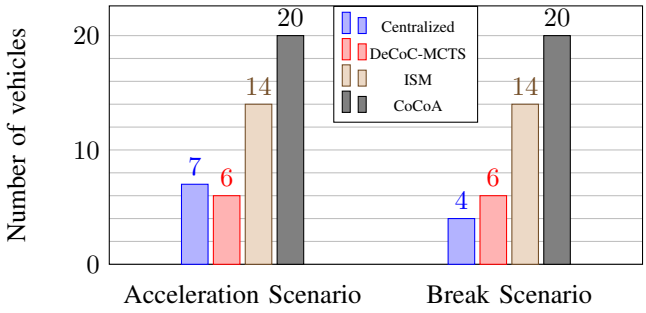


Fig. 4: Scalability results

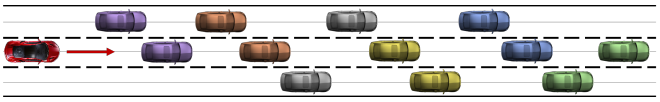


Fig. 5: Trade-off analysis: 6 coalitions of 2 CAVs each (total 12 CAVs)

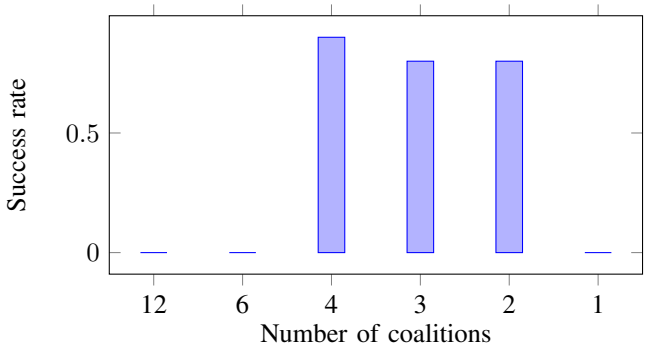


Fig. 6: Success rate for different number of coalitions constructed out of 12 CAVs

very large coalition size of 12, do not yield good results at all. This confirms our initial argument that the balance between the number of coalitions and the individual coalition size must be struck to yield the best results for collision avoidance.

VII. Conclusion

In this paper, we present a cooperative action planning algorithm for multiple interacting coalitions of CAVs to find non-colliding action plans in collision situations. The primary coalition affected by the misbehaving vehicle uses Monte Carlo Tree Search based approach to derive three action plans and sends them to the neighboring secondary coalition leader. The secondary coalition leader picks the least impact plan as the final plan for the primary coalition. The secondary coalition CAVs take into account the primary coalition plan when deriving top three secondary coalition action plans using MCTS. This sequential decision making approach is carried out until the other end of the coalition sequence. We implemented CoCoA in the multi-agent-based simulator along with state-of-the-art centralized and decentralized single coalition algorithms. Experimental results show that only CoCoA improves upon the single coalition algorithms in both reliability and scalability metrics. Additionally we presented an analysis to select the optimal coalition size and the optimal number of coalitions given a total number of CAVs.

The limitations of the current version of CoCoA algorithm include its inability to tackle a misbehaving vehicle in middle of the coalition. It also does not tackle continuous changes in the behaviour of the misbehaving vehicle such as lane changes. Additionally, it requires coalitions to be in a non-overlapping sequence. The future work includes extending our approach to allow multiple coalitions navigating laterally together and also allowing changes in misbehaviour by the misbehaving vehicle.

References

[1] F. Mohseni and L. Nielsen, "Decoupled sampling-based velocity tuning and motion planning method for multiple autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1–6.

[2] A. Nakamura, Y.-C. Liu, and B. Kim, "Short-term multi-vehicle trajectory planning for collision avoidance," *IEEE Transactions on Vehicular Technology*, 2020.

[3] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of guidance, control, and dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[4] D. Lenz, T. Kessler, and A. Knoll, "Tactical cooperative planning for autonomous highway driving using monte-carlo tree search," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 447–453.

[5] K. Kurzer, C. Zhou, and J. M. Zöllner, "Decentralized cooperative planning for automated vehicles with hierarchical monte carlo tree search," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018a, pp. 529–536.

[6] D. Patel and R. Zalila-Wenkstern, "Scalable monte carlo tree search for cavs action planning in colliding scenarios," in *2021 IEEE 32nd Intelligent Vehicles Symposium (IV21)*. IEEE, 2021, pp. 1–8 MC–MPS.14.

[7] ——, "Adaptive reward for cav action planning using monte carlo tree search," in *2021 IEEE 24th International Conference on Intelligent Transportation (ITSC)*. IEEE, 2021, pp. 1–7 MoA7.8.

[8] ——, "Collaborative collision avoidance for cavs in unpredictable scenarios," in *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)*. IEEE, 2020, pp. 1–6.

[9] J. Eilbrecht and O. Stursberg, "Cooperative driving using a hierarchy of mixed-integer programming and tracking control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 673–678.

[10] C. Burger and M. Lauer, "Cooperative multiple vehicle trajectory planning using miqp," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 602–607.

[11] Z. Wang, Y. Zheng, S. E. Li, K. You, and K. Li, "Parallel optimal control for cooperative automation of large-scale connected vehicles via admm," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1633–1639.

[12] L. Gao, D. Chu, Y. Cao, L. Lu, and C. Wu, "Multi-lane convoy control for autonomous vehicles based on distributed graph and potential field," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2463–2469.

[13] D. Bellan and C. Wartnaby, "Decentralized cooperative collision avoidance for automated vehicles: a real-world implementation," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 487–494.

[14] Y. Yuan, R. Tasik, S. S. Adhatarao, Y. Yuan, Z. Liu, and X. Fu, "Race: Reinforced cooperative autonomous vehicle collision avoidance," *IEEE transactions on vehicular technology*, vol. 69, no. 9, pp. 9279–9291, 2020.

[15] K. Kurzer, F. Engelhorn, and J. M. Zöllner, "Decentralized cooperative planning for automated vehicles with continuous monte carlo tree search," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018b, pp. 452–459.

[16] K. Kurzer, M. Fechner, and J. M. Zöllner, "Accelerating cooperative planning for automated vehicles with learned heuristics and monte carlo tree search," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1726–1733.

[17] H. Manoochehri and R. Wenkstern, "Dynamic coalition structure generation for autonomous connected vehicles," in *2017 IEEE International Conference on Agents (ICA)*. IEEE, 2017, pp. 21–26.

[18] C. Boutilier, "Sequential optimality and coordination in multiagent systems," in *IJCAI*, vol. 99, 1999, pp. 478–485.