GAMEOPT: Optimal Real-time Multi-Agent Planning and Control for Dynamic Intersections

Nilesh Suriyarachchi[1], Rohan Chandra[2], John S. Baras[1] and Dinesh Manocha[1,2]

Abstract— We propose GAMEOPT: a novel hybrid approach to cooperative intersection control for dynamic, multi-lane, unsignalized intersections. Safely navigating these complex and accident prone intersections requires simultaneous trajectory planning and negotiation among drivers. GAMEOPT is a hybrid formulation that first uses an auction mechanism to generate a priority entrance sequence for every agent, followed by an optimization-based trajectory planner that computes velocity controls that satisfy the priority sequence. This coupling operates at real-time speeds of less than 10 milliseconds in high density traffic of more than 10,000 vehicles/hr, 100× faster than other fully optimization-based methods, while providing guarantees in terms of fairness, safety, and efficiency. Tested on the SUMO simulator, our algorithm improves throughput by at least 25%, time taken to reach the goal by 75%, and fuel consumption by 33% compared to auction-based approaches and signaled approaches using traffic-lights and stop signs.

I. INTRODUCTION

Effectively navigating unsignalized intersections often requires carefully planning due to low visibility and complex maneuvers such as unprotected left turns. 40% of all crashes, 50% of serious collisions, and 20% of fatalities occur at unsignalized intersections [1]. The notion of introducing vehicle-to-infrastructure (V2I) communication capable connected autonomous vehicles (CAVs), to provide additional sensing and actuation points in the traffic flow, allows for the development of new algorithms to handle these scenarios. This has been applied successfully in complex traffic bottlenecks such as highway merging and traffic shock waves [2], [3], [4]. Recent research into leveraging cooperation among CAVs to achieve safe, fair, and efficient intersection control has provided promising results with solutions from many fields, including game theory, auctions, optimization, and deep learning. While each domain has its own advantages, no single solution has been shown to be able to achieve all the four key considerations of safety, efficiency, fairness, and real-time operation. Here, safety involves the prevention of collisions, efficiency involves maximizing the capacity of the intersection, and fairness involves treating all agents equitably. Real-time computation is also essential in this scenario, as the system must react to dynamic changes in this fast-paced environment.

The general multi-agent intersection control problem can be broadly divided into two phases. The planning phase involves the selection of the optimal entrance sequence (order in which vehicles enter the intersection) and the control phase involves generating safe trajectories for all the vehicles in

[1]Electrical and Computer Engineering Department, [2]Computer Science Department, University of Maryland, College Park, Maryland, USA. Email: {nileshs,rchandr1,baras,dmanocha}@umd.edu
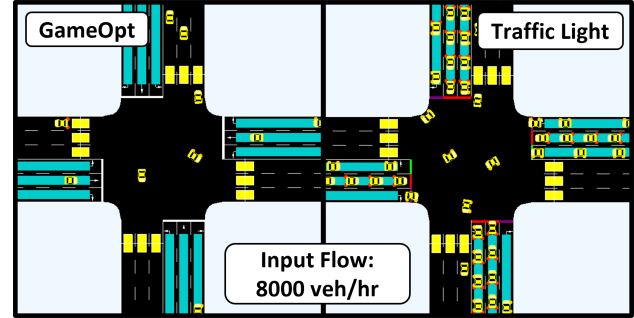GAMEOPT page with videos: https://gamma.umd.edu/gameopt

Fig. 1: **GAMEOPT vs. Traffic Lights:** We present a new approach for optimal real-time planning and control in dynamic, multi-agent unsignalized intersections. In this figure, the light blue regions denote the control zone. We show that at identical input traffic flow levels, our approach outperforms even traffic light infrastructure, resulting in less queues and enabling smooth traffic flow.

order to achieve the selected sequence. The system then transmits the output target trajectories to each vehicles' local controller for execution. Here, sequence selection can become very difficult as the number of possibilities grows exponentially with the number of vehicles. This is further complicated in multi-lane scenarios, which should allow multiple non-conflicting vehicles to enter the intersection at the same time to achieve optimal throughput. Additionally, practical intersection scenarios are *dynamic*, which means that the number of vehicles that request to cross the intersection changes with time.

A. Related Work

In Table I, we compare our approach with the current state-of-the-art in navigating unsignalized intersection scenarios on the basis of optimality guarantees, dynamic intersection handling and real-world applicability through real-time computation capabilities.

1) Auctions: Auction mechanisms have been used to navigate unsignalized intersections in real-time with fairness and feasibility guarantees [5], [6]. But they work only in *static* scenarios, where the number of participating agents is fixed, yielding poor efficiency and throughput in scenarios where the number of vehicles is variable. Incentive-compatible auctions such as Sayin et al. [7] propose a mechanism in which agents are assigned turns based on their distance from the intersection and the number of passengers in the vehicle. Carlino et al. [6] and Rey et al. [8] propose a similar mechanism but use a monetary-based bidding strategy. Buckman et al. [9] integrate a driver behavior model [10] within the first-in first-out framework to incorporate human social preference. More recently, Chandra et al. [5] proposed an auction based on the driving behavior of the agents.

Approach	Intersection	Optimality			Realtime
		Safety	Efficiency	Fairness	
Auctions	Static	✗	✗	✓	✓
Game Theory	Dynamic	✗	✗	✓	✗
Optimization	Dynamic	✓	✓	✗	✗
Deep Learning	Static	✗	✓	✗	✓
This work	Dynamic	✓	✓	✓	✓

TABLE I: **Comparison with prior work:** We highlight approaches for navigating unsignalized intersections based on the type of intersection, optimality guarantees, and real-time capability.

2) Game theory: Game-theoretic approaches have also been used for autonomous vehicle navigation [11], [12]. Li et al. [11] implement a stackelberg game in which one agent is the leader (modeled by first-out (FIFO) principle) and the other agent is a follower. Tian et al. [12], on the other hand, use a recursive $k-$level approach in which strategies at current levels are derived from previous levels. However, all agents except the ego-agent at the first level are assumed as static.

3) Optimization-based methods: These methods compute optimal trajectories and provide safety guarantees, however, they are unable to perform in real-time due to high computational costs. Existing methods circumvent this issue by using unrealistic assumptions. Rios-Torres *et al.* [2], [13] solve the optimization problem in real time using Hamiltonian analysis, which does not extend to the fully constrained problem. Computation is also restricted to a fewer vehicles in single lane roads, and no turning at intersections, to allow for close to real-time computing [14], [15], [16]. In Suriyarachchi *et al.* [3], we applied a hybrid rule and optimization based control to the highway merging scenario to achieve real-time performance. However, the selection of sequences based on rules only provides a sub-optimal solution. Most of these methods do not scale well into multi-lane scenarios.

4) Deep learning: Deep learning methods involving reinforcement learning [17] and recurrent neural networks [18] learn a planning and control policy, to be used by agents approaching an intersection. In practice, these policies do not generalize well to different environments and often do not provide guarantees in terms of safety and fairness. Furthermore, most learning methods do not easily extend to dynamic multi-agent planning and control scenarios.

B. Main Contributions:

In the field of cooperative unsignalized multi-lane intersection control, existing work is unable to combine optimality guarantees in safety, efficiency, and fairness, with real-time performance. In order to bridge this gap, we develop a novel hybrid planning and control algorithm for navigating unsignalized dynamic intersections by combining auctions with optimal control. This is the first such framework, with implications that extend beyond intersection control to more general multi-agent systems, due to the challenging nature of dynamic mechanism design. Our main contributions are:

- *Game-theoretic optimality:* The priority order sequence is fair, efficient, and tractable (Section III-A).
- *Safety guarantees:* The optimal trajectories satisfy the priority order with safety guarantees (Section III-C).

- *Real-time computation:* GAMEOPT operates on average at 1.16ms with more than 10000 vehicles per hour.
- *Efficiency:* We outperform state-of-the-art by improving throughput, time-to goal, and fuel consumption on a realistic traffic simulator (Figure 5).
- *Dynamic flow:* GAMEOPT handles varying traffic flow in different arms of the intersection (Figure 6).
- *Multi-lane capability:* Increased performance by allowing many vehicles to enter intersection simultaneously.

II. PROBLEM FORMULATION

In this section, we formally define the fully automated unsignalized intersection planning and control problem as a large optimal control problem consisting of a hierarchy of two simpler decoupled optimization problems: mechanism design followed by vehicle trajectory planning. In a dynamic scenario consisting of n vehicles, our goal is to compute velocities v_i for each vehicle such that they autonomously navigate an unsignalized multi-lane intersection safely, fairly, and efficiently. We observe the following goals/constraints:

1) *Fairness*: The computed velocities, v_i, must maximize utility across all agents and be incentive compatible.
2) *Safety*: The trajectories generated by the optimizer must not result in collisions.
3) *Efficiency*: The overall algorithm must optimize throughput, time-to-goal, and fuel efficiency.
4) The overall planning and control must not assume access to the objective or utility functions of other agents.
5) The overall planning and control must not assume a dynamics model; rather, it should compute the dynamics for each agent assuming a simple forward motion model.

The first step in this process involves defining the physical characteristics of an unsignalized intersection as well as the dynamics and control of the CAVs involved.

A. Modeling the Physical Intersection

We define an unsignalized intersection as a single-lane or multi-lane four-way crossing that is not regulated either by traffic signals or right-of-way rules. In Fig. 2, we show an example of a single-lane intersection used in our work. We define a control zone of length L_c along each arm of the intersection; vehicles in the control zone can share state information and receive actuation commands using V2I communication protocols. In this research we assume zero transmission delay and that this V2I communication channel has perfect conditions. The center of the intersection is referred to as the conflict zone, where collisions are likely to occur. We denote the number of vehicles in the control zone in roads $0, 1, 2, 3$ as n_1, n_2, n_3, n_4, respectively, with the total number of vehicles in the control zone being $n = n_1 + n_2 + n_3 + n_4$.

Vehicles are autonomous and can freely choose between turning left, turning right, and going straight. The input flow rate of vehicles in each control zone of the intersection can be adjusted along with the ratio of vehicles that turn left, turn right, or go straight. We vary the input flow rate between $2,000$ vehicles per hour to $10,000$ vehicles per hour, where vehicles are generated according to a Poisson distribution.
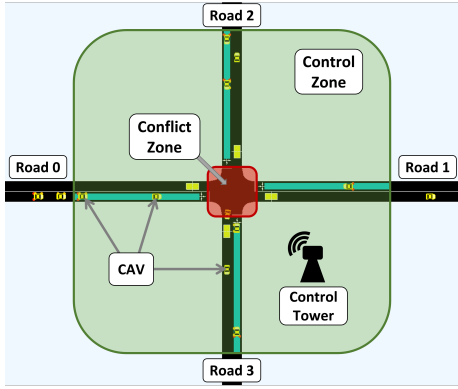
Fig. 2: **Regions of interest:** We highlight the control zone in which vehicles communicate with the control tower and the conflict zone in which vehicles cross over to their desired target road.

B. Vehicle Dynamics and Control Variables

In real-world scenarios, vehicle dynamics are hard to model accurately due to their non-linearity. Standard practice in control theory suggests the use of a local controller, z_i, that produces low-level controls for the longitudinal and lateral motion of a vehicle according to the following equation,

$$\frac{\delta s_i}{\delta t} = \mathcal{F}(t, s_i, z_i). \tag{1}$$

Here, we represent $\mathcal{F}$ as the unknown non-linear dynamic function. s_i denotes the distance of the agent i from the intersection measured in the Frenet coordinate system.

Dynamics: Since $\mathcal{F}$ is generally unknown, we define the high-level dynamics by the following velocity control scheme:

$$\begin{aligned} \frac{\delta s_i}{\delta t} &= v_i \\ v_i(t) &= \boldsymbol{u}_i(t) \end{aligned} \tag{2}$$

for $i \in \{1, \dots, n\}$.

Control variables: The control input, $\boldsymbol{u}_i(t)$, represents the command velocity for vehicle i. We compute the optimal value for $\boldsymbol{u}_i(t)$ with respect to safety, efficiency and reachability guarantees described in the following sections.

State vector: In addition to $s_i(t)$ and $v_i(t)$, we assign an indicator variable $l_c^i(t) \in \{0, 1, \dots, L_n - 1\}$, which signifies which lane the vehicle is currently in, with L_n representing the number of lanes on each road. We also assign a bidding value $b_i(t)$ to each vehicle according to III-A, which allows for the computation of the sequence order in which vehicles would cross the intersection. We also obtain the length l_i, and maximum acceleration a_i^{max} and deceleration a_i^{min} capabilities of each CAV i. The CAVs are thereby completely defined by the state vectors:

$$\boldsymbol{\Phi}_i(t) = [s_i(t), v_i(t), l_c^i(t), b_i(t), l_i, a_i^{max}, a_i^{min}]^\top \tag{3}$$

for $i \in \{1, \dots, n\}$. We also define a parameter tuple $M_s = \{M_{sr}, M_{sl}\}$, which represent the safety margins needed to prevent rear-end and lateral collisions.

C. Dynamic Traffic Intersection Planning & Control Problem

The dynamic traffic intersection problem involves computing suitable, collision-free, continuous trajectories for all vehicles approaching an intersection to achieve efficient and fair traffic flow. This problem can be formulated as an optimal control problem, with the objective of computing the optimal command velocities $\boldsymbol{u}_i$ for each CAV $i \in \{1, \dots, n\}$, resulting in the minimization of the maximum time taken by each CAV to cross the intersection:

$$\begin{aligned} &\min_{\{\boldsymbol{u}_i\}} \quad \max_i \ t_f^i \\ &s.t. \quad \mathcal{C}(\{\boldsymbol{\Phi}_i\}, \bar{v}, \mathbb{S}(M_s)) \end{aligned} \tag{4}$$

where t_f^i is the intersection crossing time of vehicle i. $\mathcal{C}$ is a set of constraints, discussed in Section III-C, that depend on the state vectors $\{\boldsymbol{\Phi}_i\}$ of the CAVs, and safety requirements $\mathbb{S}(M_s)$ which prevent collisions between vehicles.

Equation 4 is, however, intractable to optimize directly because it implicitly assumes that agents are navigating the intersection according to the optimal priority order, q^*. Equation 4 can be equivalently solved by optimizing jointly over sequences, q, and the control velocities, $\boldsymbol{u}_i$, needed to implement the sequence.

$$\begin{aligned} &\min_{\{\boldsymbol{u}_i, q\}} \quad \sum_{i=1}^n \lambda(\boldsymbol{u}_i - \bar{v})^2 + (1 - \lambda)(\boldsymbol{u}_i - v_i)^2 \\ &s.t. \quad \mathcal{C}(\{\boldsymbol{\Phi}_i\}, \bar{v}, \mathbb{S}(M_s), q) \end{aligned} \tag{5}$$

where $\bar{v}$ is the speed limit. We solve this optimization problem in Section III-C.

Intractability of Equation (5): The optimal control velocities, $\boldsymbol{u}_i$, depend on the sequence order q of the vehicles entering the conflict zone. The naive solution is to solve a mixed-integer optimization problem for every possible entrance sequences q by performing an exhaustive search. This process involves two sub-problems—generating a sequence q followed by computing the optimal velocity commands $\boldsymbol{u}_i$ for that sequence. The total number of possible sequences q grows exponentially with the number of vehicles in the control zone which would take the form $\mathcal{O}\left(\frac{(n_1 + n_2 + n_3 + n_4)!}{n_1! n_2! n_3! n_4!}\right)$, where $n = n_1 + n_2 + n_3 + n_4$ is the number of vehicles in the control zone. Therefore, current methods to solve Equation 5 are intractable.

We propose the use of auctions to obtain an optimal priority sequence order q^* in $\mathcal{O}(n \log n)$ time, significantly reducing the combinatorial runtime complexity of the mixed-integer formulation. We run the optimization over $\boldsymbol{u}_i$ for the selected optimal sequence q^* which can be solved in real time.

D. Auction Theory

Auctions are a central theme in mechanism design. An auction allocates k items among n agents. Each agent i has a private valuation $\boldsymbol{\zeta}_i$ and submits a bid $\boldsymbol{b}_i$ to receive at most one item of value $\boldsymbol{\alpha}_i$. In any auction, we have $\boldsymbol{b}_1 > \boldsymbol{b}_2 > \dots > \boldsymbol{b}_K$ and $\boldsymbol{\alpha}_1 > \boldsymbol{\alpha}_2 > \dots > \boldsymbol{\alpha}_K$. Each auction has an
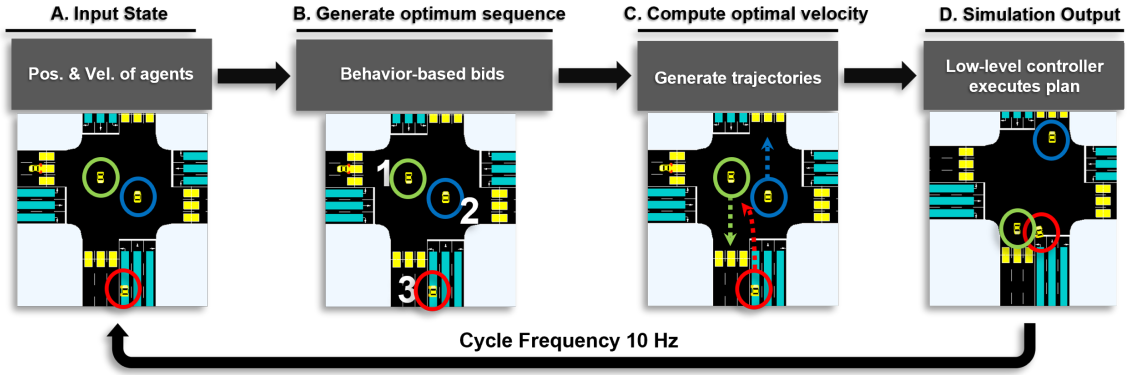
Fig. 3: **GameOpt overview:** Our approach begins by reading in the positions and velocities of all agents in the control zone (blue). Our approach is hybrid; in the planning phase, GameOpt collect the bids from every agent and generate an optimal priority sequence (Section III-A). In the controls phase, we use an optimization-based trajectory planner to compute the optimal velocity for each agent that satisfies the priority order while simultaneously guaranteeing safety and real-time performance (Section III-C).

allocation rule (who gets what), a payment rule (who pays what), and a utility function (how much value an agent gets based on their bid). In a sponsored search auction, which is the model we are using in this work, the allocation rule is that the agent with the jth highest bid is allocated the jth most valuable item, $\boldsymbol{\alpha}_i$. The quasi-linear utility $\boldsymbol{\theta}_i$ [19] incurred by i is as follows,

$$\boldsymbol{\theta}_i(\boldsymbol{b}_i) = \boldsymbol{\zeta}_i \boldsymbol{\alpha}_i - \sum_{j=i}^{k} \boldsymbol{b}_{j+1} (\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_{j+1}). \qquad (6)$$

The term $\boldsymbol{\zeta}_i \boldsymbol{\alpha}_i$ denotes the time reward gained by i by moving on the ith turn. The payment term, $\sum_{j=i}^{k} \boldsymbol{b}_{j+1} (\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_{j+1})$, represents the risk [20] associated with moving on that turn. It follows that an allocation of a conservative agent to a later turn (smaller $\boldsymbol{\alpha}$) also presents the lowest risk and vice-versa.

III. GameOpt: Methodology

We present our overall approach in Figure 3. Computation begins by receiving the positions, velocities and initial bids from all agents in the control zone via V2I communication (Section II). In the following planning phase, this information along with the processed bids are used to generate an optimal priority order or sequence, which determines the order in which agents will enter the intersection (Section III-A). The control phase of our approach then involves using an optimization-based trajectory planner to compute the optimal velocity (Section III-C) for each agent, which satisfies the priority order while simultaneously guaranteeing safety and real-time performance.

A. Game: *Selecting the Priority Order, q^**

Choosing an optimal ordering for agents to navigate unsignalized traffic scenarios is equivalent to allocating each agent a turn in which they would cross the intersection. Such an allocation depends on the incentives of the agents which, in many cases, are not known apriori. Auctions model the incentives of agents in unsignalized traffic scenarios using an optimal combination of bidding, allocation, and payment strategies. In the rest of this section, we present the auction framework for generating an optimal priority order, followed by an analysis of its fairness.

The Sponsored Search Auction (SSA), described in Section II-D, is an ideal mechanism to generate a priority order for the agents. *The agent with the highest priority bid is allowed to navigate the scenario first, followed by the agent with the next highest priority, and so on.* This algorithm runs in polynomial time, since the main computation at this stage is dominated by sorting the agents' bids [21]. The priority value of an agent is its true private valuation, $\boldsymbol{\zeta}_i$, and our approach accommodates many different metrics that can be used for computing the priority value for an agent. For instance, our approach can work with both driver behavior-based [5], distance-based [9], and monetary-based [6] bidding strategies. We compute the priority value based on its velocity and its distance from the intersection. More specifically, agents that are closer to the intersection and have higher velocities have higher priority to cross the intersection.

More formally, recall that $s_i(t)$ measures the distance of agent i from the intersection, and we use τ_i to denote the time in which i shall reach the intersection, based on its current velocity and $s_i(t)$. Then, the priority value for agent i is,

$$\boldsymbol{\zeta}_i = s_i(t) \times (c - \tau_i), \qquad (7)$$

where c is a constant. To avoid a single lane dominating the intersection (due to a constant flow of fast moving vehicles) and to prevent congestion, we reward agents that have been waiting in queue for a long time. Thus, $\boldsymbol{\zeta}_i \leftarrow w_i \boldsymbol{\zeta}_i$, where w_i represents the waiting time reward value. Such a strategy of mitigating congestion has been commonly used in auction-based approaches [6]. Finally, to deal with multi-agent dynamic traffic, we implement an overflow strategy where, if an agent with a higher priority is behind an agent with a lower priority, then the former transfers some amount of their bid to the latter. This phenomenon is commonly observed in agent-based motion models such as the Social Forces model.

In [5], the authors show that the sponsored search auction is game-theoretically optimal for static intersections consisting of 1 vehicle on each arm of the intersection. In this work, we extend the proof for dynamic traffic intersections where we must take into account three factors–(i) multiple

vehicles on each arm of the intersection and (ii) lower priority vehicles blocking higher priority vehicles, and (iii) providing waiting time rewards to agents.

Fairness Analysis of q^*: From a game-theoretic perspective, fairness implies that no agent is incentivized to "cheat" or, in other words, this occurs when the dominant strategy for each agent is to bid their true valuation ζ_i. Incentivizing traffic-agents to bid their true value ($b_i = \zeta_i$) as a dominant strategy is known as incentive compatibility [19]. Chandra et al. [5] showed that SSAs implemented at static intersections with a single agent on each arm are incentive-compatible. We extend the theoretical framework of static SSAs to dynamic intersections with multiple vehicles, taking into account *overflow*, which is when a lower priority vehicle blocks a higher priority one.

Theorem III.1. *Incentive compatibility for dynamic intersections: For each agent i for $i = 1, 2, \ldots, n$ and $n = n_1 + n_2 + n_3 + n_4$ where n_j represents the number of vehicles on the jth arm, bidding $b_i = \zeta_i$ is the dominant strategy.*

The next desired property in a fair auction is welfare maximization [19] which maximizes the total time reward (Section II-D) earned by every agent. We show that SSAs maximize social welfare for dynamic intersections as well.

Theorem III.2. *Welfare maximization for dynamic intersections: Social welfare of an auction is defined as $\sum_i \zeta_i \alpha_i$ for each agent i for $i = 1, 2, \ldots, n$ and $n = n_1 + n_2 + n_3 + n_4$ where n_j represents the number of vehicles on the jth arm. Bidding $b_i = \zeta_i$ maximizes social welfare for every agent.*

Finally, we propose a novel strategy to address overflow by transferring a portion of the higher priority agent's bid to the lower priority agent i. We denote such a transfer by $(\hat{a}, \hat{b} = a \xrightarrow{c} b)$, where $\hat{a} = a - c, \hat{b} = b + c$. More formally,

Theorem III.3. *Overflow prevention: In a current SSA, suppose there exists $(i, j) \in [n_k] \times [n_k], k = 1, 2, 3, 4$, such that $\zeta_i > \zeta_j$ and $s_i[t] > s_j[t]$. Let $\hat{\zeta}_i, \hat{\zeta}_j = (\zeta_i \xrightarrow{q} \zeta_j)$. If*
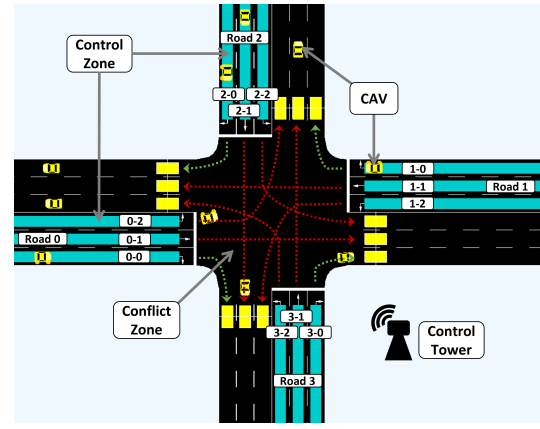
$$q < \zeta_i \left(1 - \frac{\alpha_{i+m}}{\alpha_i}\right) - \sum_{s=i}^{i+m-1} \zeta_{s+1} \left(\frac{\alpha_s - \alpha_{s+1}}{\alpha_i}\right),$$

the new SSA with $\hat{\zeta}_i, \hat{\zeta}_j$ as the new priority values for i, j is incentive compatible.

As a final remark, Carlino et al. [6] show that multiplying a bid by a waiting time reward does not change the incentive compatibility of the auction. We defer the proofs of Theorems III.1, III.2, and III.3 to [22].

B. Multi-lane Intersection Planning and Control

A key capability of our method is its ability to handle traffic effectively in multi-lane intersections. The extension from single lane to multi-lane is not straight forward. Many assumptions that can be made in single lanes (such as one vehicle at a time in the conflict zone) do not hold up in a multi-lane scenario. Therefore, we introduce a system to classify all vehicles in the control zone into groups based on their desired trajectory (origin road and intention) in order to handle conflicts among vehicles accurately. Group

(a) Multi-lane intersection structure with lane-based groups

Lane group	No-conflict group set
0-1	0-2, 1-1, 2-2
0-2	0-1, 1-2, 3-1
1-1	0-1, 1-2, 3-2
1-2	0-2, 1-1, 2-1
2-1	1-2, 2-2, 3-1
2-2	0-1, 2-1, 3-2
3-1	0-2, 2-1, 3-2
3-2	1-1, 2-2, 3-1

(b) Non-conflicting trajectory groups at intersection

Fig. 4: **Conflict handling at multi-lane intersections**. We propose a novel strategy for conflict handling using vehicle grouping based on non-colliding trajectories.

x-y: x=road_num, y=intention (Here intention values are: *0=right_turn, 1=go_straight, 2=left_turn*). For example, a vehicle from road 1 taking a left turn would be in group 1-2. Fig. 4a shows the trajectories that vehicles in each lane can take, along with the labeling used to allocate groups based on these trajectories, marked on each of the lanes.

Here, we notice that vehicles taking right turns at the intersection (groups 0-0, 1-0, 2-0, 3-0) do not conflict with any other vehicle groups. Therefore, they are allowed to enter into the intersection freely, and the constraints in the optimization in III-C, reflect this. For each of the remaining lane groups, we need to identify which other lane groups can be allowed to enter the intersection at the same time. We denote this collection as the non-conflicting group set for each corresponding main lane group, as shown in Table 4b. For example, consider the main lane group 0-1. The groups with trajectories that do not conflict with this main group are 0-2, 1-1, and 2-2 along with all the right turn groups (0-0, 1-0, 2-0, 3-0). What this effectively means is that any vehicles belonging to the non-conflicting group set of a main lane group will be allowed to enter the intersection at the same time along with a vehicle of the main lane group. Note that vehicles in the right turn at intersection groups (0-0, 1-0, 2-0, 3-0) are directly marked as non-conflicting and are included in the non-conflicting groups list for all vehicle groups by default. The non-conflicting groups are based on the assumption that the vehicles approaching the conflict zone have completed all strategic lane changes.

C. OPT: *Computing Optimal Velocities*

The Optimization Problem:

Based on the optimal priority sequence q^* from Section III-A, the optimal command velocities, $\boldsymbol{u}_i$, for each vehicle are computed by optimizing Equation 5 for a fixed sequence. The objective function restated:

$$\mathcal{J}(\boldsymbol{u}_i|q^*) = \min_{\{\boldsymbol{u}_i\}} \sum_{i=1}^{n} \lambda(\boldsymbol{u}_i - \bar{v})^2 + (1-\lambda)(\boldsymbol{u}_i - v_i)^2 \quad (8)$$

The first quadratic term of Equation 8 prevents $\boldsymbol{u}_i$ from deviating from the speed limit $\bar{v}$ (results in improved throughput). The second quadratic term minimizes the control effort applied to optimize fuel efficiency. Below we discuss constraints on safety and the influence of q^* on Equation (8).

Constraints on Safety and Compliance with q^*:

We consider both longitudinal (rear-end) collisions in the control zone and collisions in the conflict zone. Rear-end collisions along each of the lanes in the control zone are prevented by ensuring all vehicles in the same lane maintain a safe distance between each other. Formally,

$$\left| s_j^k(t+1) - s_{j\prime}^k(t+1) \right| \geq l_j + M_{sr} \quad (9)$$

for all k lanes in the control zone and for all agents $j, j\prime$ with $j \neq j\prime$ in the kth lane. l_j represents the length of the agent j and M_{sr} is a safety margin added to maintain a safe gap between vehicles and to account for imperfections in sensing and actuation. We compute the future position $s_i(t+1)$ using,

$$s_i(t+1) = s_i(t) - \Delta t \left(\frac{v_i(t) + \boldsymbol{u}_i(t)}{2} \right) \quad (10)$$

Here, Δt is defined as the planning time step of the controller.

Next, we also need to ensure that no collisions occur inside the conflict zone. This task reduces to ensuring no vehicles belonging to conflicting lane groups (III-B) enter the intersection simultaneously. Therefore, for each vehicle i in the sequence q^*, paired with a vehicle j in a conflicting group where i has higher priority over j, we add the constraint,

$$t_c^i + \frac{(l_i + M_{sl})}{\boldsymbol{u}_i} \leq t_c^j \quad (11)$$

where $t_c^i = \frac{s_i(t+1)}{\boldsymbol{u}_i}$ represents the time at which i would arrive at the intersection. The above constraint ensures that command velocities $\boldsymbol{u}_i$ are chosen such that vehicles proceed through the conflict zone in the order of the sequence under consideration q^*. Note that this constraint is applied to every conflicting pair of vehicles; non-conflicting vehicles are allowed to enter the intersection at the same time, increasing the efficiency of the system. The safety margin M_{sl} corresponds to the width of the intersection and prevents lateral collisions occurring when conflicting vehicles cross the intersection.

The safety constraints with respect to ordering q^* in Equations (9) and (11) result in an intractable mixed-integer quadratic programming (MIQP) optimization problem. This problem is further compounded due to lane indexing, which introduces a new variable indicating the agent's lane. We remove the dependency over the lane index by separating the constraints for each lane and using the multi-lane conflict resolution scheme introduced in III-B. This results in simplification of Equations (9) and (11), reformulating the original MIQP as a quadratic programming (QP) optimization problem; this QP can be solved in real-time. More formally, Equation (9) can be simplified as,

$$\boldsymbol{u}_j^k - \boldsymbol{u}_{j+1}^k \geq \left(v_{j+1}^k - v_j^k \right) + \frac{2}{\Delta t} \left(s_j^k - s_{j+1}^k + l_j + M_{sr} \right) \quad (12)$$

for all k lanes in the control zone and for all $j, j+1$ consecutive vehicles belonging to lane k. And Equation (11) can be simplified as,

$$\boldsymbol{u}_j \left(s_i - \frac{\Delta t}{2}.v_i + l_i + M_{sl} \right) \leq \boldsymbol{u}_i \left(s_j - \frac{\Delta t}{2}.v_j \right) \quad (13)$$

for all vehicles i in sequence q^* and for all vehicles j in a conflicting group to vehicle i.

Finally, we enforce reachability by bounding the command velocities by the speed limit, $\bar{v}$, as well as the acceleration and breaking capabilities of each individual vehicle.

$$0 \leq \boldsymbol{u}_i(t) \leq \bar{v} \quad (14)$$

$$a_i^{min}\Delta t \leq \boldsymbol{u}_i(t) - v_i(t) \leq a_i^{max}\Delta t \quad (15)$$

Solving OPT: *The constraints (12), (13), (14), and (15) are provided to* OPT *with objective function given by Equation (8), which is solved numerically using Gurobi (version 9.1.2) [23].*

The computed target $\boldsymbol{u}_i$ command velocities are transmitted to each of the CAVs in the control zone. Note that we do not consider delays in communication and assume data is transmitted instantaneously via V2I communication. The low-level controller on-board each CAV then computes the acceleration or deceleration needed, to achieve the desired $\boldsymbol{u}_i$ command velocity within the control time (Δt) duration.

Finally, a key feature of our approach is that it is robust to drift in the executed control. Our approach operates at a cycle frequency of 100 ms after which new command velocities are obtained, updating the previous controls. Therefore, any deviation between optimal control and the executed control is erased during every 100 ms update cycle.

IV. EXPERIMENTS AND RESULTS

In this section, we evaluate the capabilities of our approach, study the impact of varying environmental parameters, and compare against other real-time methods.

A. Experiment Setup

We evaluate the performance of GAMEOPT in a multi-lane four-arm intersection simulation using the SUMO platform [24] with the following parameters; *Control zone length (L_c) = 150m, Speed limit ($\bar{v}$) = 20m/s Objective trade-off (λ) = 0.7, Safety margins: M_{sr} = 2.0 and M_{sl} = 25.0. The high-level controller uses the TraCI traffic controller interface*

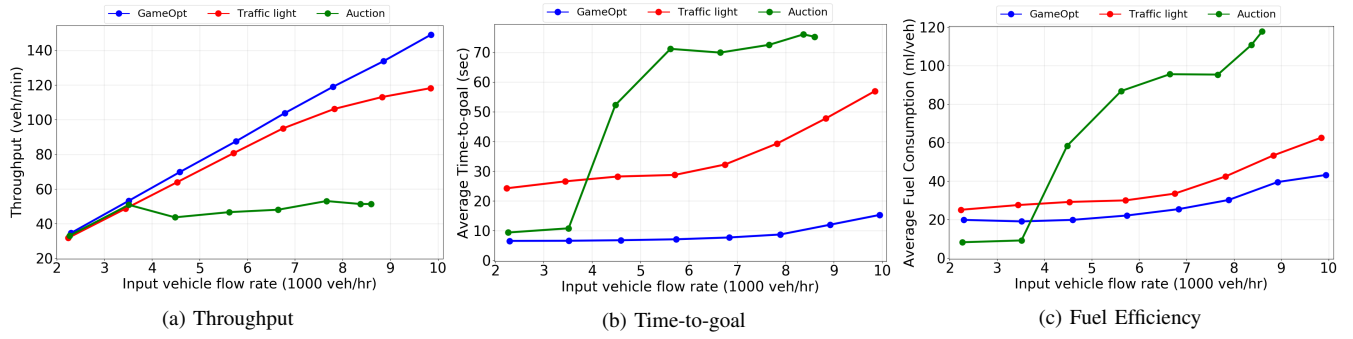(a) Throughput   (b) Time-to-goal   (c) Fuel Efficiency

Fig. 5: **Comparison with baselines:** GAMEOPT increases throughput by approximately 25% and 174% compared to TRAFFIC-LIGHT and AUCTION, respectively. Furthermore, our method reduces the time-to-goal by 75% and 81%, respectively, and improves fuel efficiency by 33% and 66%, respectively, compared with TRAFFIC-LIGHT and AUCTION.

to communicate with the simulation. All simulations and optimization algorithms run on a personal computer with an Intel i7-8750H CPU and 32GB of RAM.

B. Evaluation Metrics and Baselines

The key performance metrics used to evaluate merging algorithms are *throughput* (number of vehicles that can travel through the intersection per min), *time-to-goal* (average time vehicles spend in the control zone) and *fuel consumption*. To showcase the properties and appropriately assess the performance of GAMEOPT, we compare it with two other real-time capable baseline approaches using the same inputs. The first is an auction framework where the bidding strategy is based on the order in which agents arrive at the intersection, similar to the FIFO principle. To test the limits of our approach, we also compare against well-tuned traffic lights (light timing set to maximize performance), a well-established signaled intersection traffic management system. We refer to these baselines as TRAFFIC-LIGHT and AUCTION in the following sections. Note that the Krauss vehicle following model was used in these comparisons.

C. Efficiency Analysis

From the blue curves in Figure 5, we observe that GAMEOPT exhibits linearly increasing throughput with increasing input flow while the other two methods become capped after a point. Similarly, we see that GAMEOPT is capable of maintaining low temporal delays as well as low fuel consumption across all values of input flow. This highlights the consistent performance capabilities of the GAMEOPT method, even at very high input flow rates such as 10,000 vehicles per hour. Note that input flow rate is defined as the total number of vehicles that enter into the intersection simulation per hour on all four roads.

Results and Comparison with Baselines: In Figure 5, we show results of comparing GAMEOPT with TRAFFIC-LIGHT and AUCTION based on throughput, time-to-goal, and fuel efficiency. Perhaps the most significant result is that not only does GAMEOPT outperform the AUCTION approach, it also outperforms TRAFFIC-LIGHT, a well-established method optimized for signalized intersections. In Figure 5a, we observe that GAMEOPT linearly increases in throughput (equivalent to maximizing flow through the intersection) with increasing flow rate. At high flow rates, GAMEOPT performs approximately 25% and 174% better when compared to TRAFFIC-LIGHT and AUCTION, respectively. We also note that these

relative performance improvements become more apparent with the increase in flow rate.

This follows our intuition that with fewer vehicles, we do not require a complex optimization-based method; a simpler signal-based method suffices. However, as the traffic on the road increases, which often leads to increased congestion and delays, the benefit of GAMEOPT becomes apparent. Furthermore, from Figures 5b and 5c, we observe that at high flow rates our method reduces the time-to-goal by 75% and 81%, respectively, and reduces fuel consumption by 33% and 66%, respectively, compared with TRAFFIC-LIGHT and AUCTION. This shows that GAMEOPT results in less waiting in queues along with less acceleration and braking tasks, resulting in improved fuel efficiency.

(a) Impact of increased speed limit.

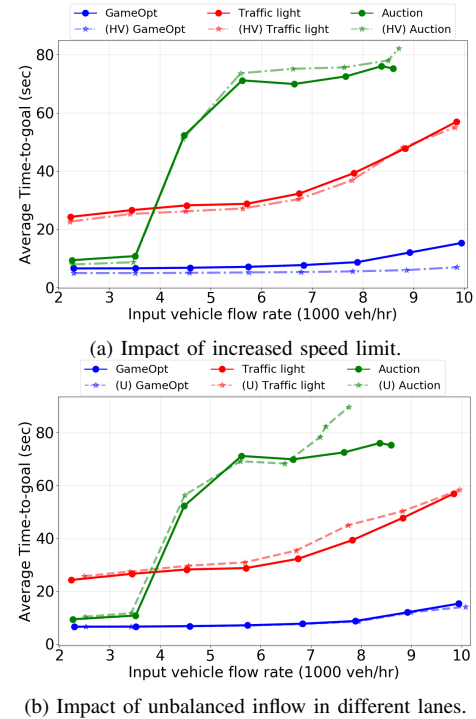
(b) Impact of unbalanced inflow in different lanes.

Fig. 6: Impact of increasing the speed limit and unbalanced inflow rates on the average time-to-goal. GAMEOPT handles both cases well and even performs better with a higher speed limit. TRAFFIC-LIGHT remains unchanged while AUCTION is worse in both cases.

We also analyze the impact of increasing road speed limits from 20m/s to 25m/s, as this is a possibility in fully autonomous systems. In Figure 6a, we observe that while

TRAFFIC-LIGHT and AUCTION both remain unchanged, GAMEOPT exhibits reduced time-to-goal. This shows that the overall capabilities of GAMEOPT can be increased further with improved road infrastructure. We also test the impact of creating unbalanced vehicle demand on different arms of the intersection by varying the inflow rate on each arm. Assuming we keep the total number of vehicles constant, some arms experience unusually high traffic density. As expected, from Figure 6b we observe that AUCTION's and TRAFFIC-LIGHT's performance worsens whereas GAMEOPT remains unchanged, demonstrating robustness to this configuration.

Real-time Operation: The algorithm in our optimization-based approach operates at a frequency of $f_s = 10$Hz (updates all control commands every 100ms). The time required for the optimizer to compute control outputs for a single sequence has a mean of 1.16ms, standard deviation of 1.73ms, and a worst-case maximum of 17.25ms. This is significantly lower than the controller update rate of 100ms, thus enabling real-time operation. The fastest optimization-based method in literature takes up to 2.4 seconds for 50 vehicles [16], [25] in a simple single lane environment. In comparison, GAMEOPT takes less than 18 milliseconds for 50+ vehicles in a multi-lane setting. This is a speed-up of around $130\times$ in considerably more complex environments.

D. Effect of Overbidding and Underbidding

According to Theorem III.1 in Section III-A, overbidding or underbidding yields a lower utility [5]. In simulation, when the auction is combined with a trajectory planner, we observe even more drastic consequences. In the case of overbidding, the agent submits a higher than expected bid to the trajectory planner. The planner accommodates this higher bid by reducing a conflicting agent's speed, causing the latter to slow down and create a dangerous situation with possible rear-end collisions. In the case of underbidding, the trajectory planner reduces the speed of the agent that is underbidding, resulting in possible rear-end collisions.

V. CONCLUSION, LIMITATIONS, AND FUTURE WORK

We propose GAMEOPT: a novel hybrid approach to cooperative intersection control for dynamic, multi-lane, unsignalized intersections. Our algorithm couples an auction mechanism that generates a priority entrance sequence with an optimization-based trajectory planner that computes the optimal velocity commands to achieve this sequence. This hybrid approach allowed us to achieve real-time computation capabilities in high-density multi-lane traffic, while providing guarantees in terms of efficiency, safety, and fairness. Our performance was verified using the SUMO platform, and we show that GAMEOPT improves throughput by around 25%, time spent by 75%, and fuel consumption by 33%, compared to auction-based approaches and signaled approaches using traffic-lights and stop signs. We also demonstrate that GAMEOPT operates at real-time speeds (< 10 milliseconds), which is at least $100\times$ faster than prior methods.

Future work in this area would include exploring different kinds of auctions for flexibility in intersection control management, the impact of imperfect communication on the performance of the algorithm, and the capability to handle the needs of heterogeneous vehicles and mixed traffic.

REFERENCES

[1] O. Grembek, A. A. Kurzhanskiy, A. Medury, P. Varaiya, and M. Yu, "Introducing an intelligent intersection," *Intelligent Transportation Systems Reports*, no. 13, 2018.

[2] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[3] N. Suriyarachchi, F. M. Tariq, C. Mavridis, and J. S. Baras, "Real-time priority-based cooperative highway merging for heterogeneous autonomous traffic," in *2021 IEEE International Transportation Systems Conference (ITSC)*, 2021, pp. 2019–2026.

[4] N. Suriyarachchi and J. S. Baras, "Shock wave mitigation in multi-lane highways using vehicle-to-vehicle communication," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021, pp. 1–7.

[5] R. Chandra and D. Manocha, "Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging," *IEEE Robotics and Automation Letters*, 2022.

[6] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2013, pp. 529–534.

[7] M. O. Sayin, C.-W. Lin, S. Shiraishi, J. Shen, and T. Başar, "Information-driven autonomous intersection control via incentive compatible mechanisms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 912–924, 2018.

[8] D. Rey, M. W. Levin, and V. V. Dixit, "Online incentive-compatible mechanisms for traffic intersection auctions," *European Journal of Operational Research*, 2021.

[9] N. Buckman, A. Pierson, W. Schwarting, S. Karaman, and D. L. Rus, "Sharing is caring: Socially-compliant autonomous intersection negotiation," 2020.

[10] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[11] N. Li, Y. Yao, I. Kolmanovsky, E. Atkins, and A. R. Girard, "Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections," *IEEE Trans. on Intelligent Transportation Systems*, 2020.

[12] R. Tian, N. Li, I. Kolmanovsky, Y. Yildiz, and A. R. Girard, "Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[13] A. A. Malikopoulos and L. Zhao, "Optimal path planning for connected and automated vehicles at urban intersections," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 1261–1266.

[14] Y. Bian, S. E. Li, W. Ren, J. Wang, K. Li, and H. X. Liu, "Co-operation of multiple connected vehicles at unsignalized intersections: Distributed observation, optimization, and control," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10 744–10 754, 2020.

[15] H. Pei, Y. Zhang, Y. Zhang, and S. Feng, "Optimal cooperative driving at signal-free intersections with polynomial-time complexity," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021.

[16] S. A. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Transactions on IV*, vol. 3, no. 3, pp. 287–299, 2018.

[17] A. P. Capasso, P. Maramotti, A. Dell'Eva, and A. Broggi, "End-to-end intersection handling using multi-agent deep reinforcement learning," *arXiv preprint arXiv:2104.13617*, 2021.

[18] J. Roh, C. Mavrogiannis, R. Madan, D. Fox, and S. S. Srinivasa, "Multimodal trajectory prediction via topological invariance for navigation at uncontrolled intersections," *arXiv preprint arXiv:2011.03894*, 2020.

[19] T. Roughgarden, *Twenty lectures on algorithmic game theory*. Cambridge University Press, 2016.

[20] M. Wang, N. Mehr, A. Gaidon, and M. Schwager, "Game-theoretic planning for risk-aware interactive agents," *IROS*, 2020.

[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 2009.

[22] GameOpt, "Gameopt project supplementary material," 2022. [Online]. Available: https://gamma.umd.edu/gameopt

[23] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: http://www.gurobi.com

[24] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[25] M. H. M. Nor and T. Namerikawa, "Optimal motion planning of connected and automated vehicles at signal-free intersections with state and control constraints," *SICE Journal of Control, Measurement, and System Integration*, vol. 13, no. 2, pp. 30–39, 2020.