engineering proceedings

MDPI

Proceeding Paper

Game-Theoretic Framework for Coordinating Mixed Traffic for Emergency Vehicle Passage †

Wei-Xiang Li [1], I-Hsien Liu [2,*], Kuan-Ting Lee [1] and Chu-Fen Li [3]

[1] Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan 701401, Taiwan; wxli@cans.ee.ncku.edu.tw (W.-X.L.); ktlee@cans.ee.ncku.edu.tw (K.-T.L.)
[2] Department of Electrical Engineering, National Cheng Kung University, Tainan 701401, Taiwan
[3] Department of Finance, College of Management, National Formosa University, Yunlin 632301, Taiwan; chufenli@gmail.com
* Correspondence: ihliu@cysec.ee.ncku.edu.tw
† Presented at the 7th Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability 2025 (ECBIOS 2025), Kaohsiung, Taiwan, 23–25 October 2025.

Abstract

We address the challenge of coordinating traffic at unsignalized intersections, particularly with an emergency vehicle, by proposing a game-theoretic decision model. In this non-cooperative game, each vehicle selects a Go or Yield strategy to maximize a utility function based on efficiency, collision risk, and driving style. We use the Iterated Best Response algorithm to find the pure strategy Nash equilibrium. A MATLAB–SUMO co-simulation validates that our model significantly enhances safety and efficiency while substantially reducing travel time.

Keywords: emergency vehicle preemption; vehicle priority; Nash equilibrium

1. Introduction

Coordinating autonomous vehicles (AVs) at unsignalized intersections, especially with high-priority emergency vehicles, is a critical challenge. Existing methods, including game-theoretic models, are often hampered by high computational costs from simulation-based payoffs and assumptions of vehicle homogeneity. To address this, we propose a computationally efficient, static game-theoretic framework that explicitly incorporates vehicle heterogeneity. The method developed in this study features a personalized utility function to model diverse driving styles and an analytical payoff calculation based on geometric conflicts, which bypasses simulation to rapidly construct a complete payoff matrix. A pure strategy Nash equilibrium is then swiftly computed from this matrix using the Iterated Best Response (IBR) algorithm. The framework's effectiveness in ensuring EV priority and traffic stability is validated via MATLAB–SUMO co-simulation.

2. Related Work

Existing game-theoretic models for vehicle coordination are computationally inefficient, requiring iterative simulations, and often ignore diverse driving styles [1]. The developed analytical static game framework addresses this by using a pre-calculated payoff matrix to decouple simulation from equilibrium-solving [2], ensuring rapid decision-making. This framework also quantifies different driving personalities in the utility function, making it both efficient and behaviorally aware [3].

Check for updates

Academic Editors: Teen-Hang Meen, Cheng-Yi Chen and Cheng-Fu Yang

Published: 24 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

3. Coordinating Mixed Traffic Methodology

The game-theoretic decision-making framework was designed to address the coordination problem at unsignalized intersections. The overarching goal of this method is to find a stable traffic strategy that ensures safety and accommodates different driving styles, all while circumventing the high computational costs of traditional dynamic simulations.

3.1. Game Model

To analyze the complex interactions at the unsignalized intersection, we formulate this decision-making problem as a static, non-cooperative, complete information game. This game, denoted by G, can be represented by a triplet as follows.

$$\mathrm{G} =< \mathrm{P}, \mathrm{A}, \mathrm{U} > \tag{1}$$

where P is the set of players, A is the action space, and U is the set of utility functions for all players.

The game-theoretic framework models the interaction at an unsignalized intersection as a static game involving a fixed set of players, where each player corresponds to a traffic participant. Each player chooses from a discrete set of actions, specifically Go or Yield, representing the intention to either proceed through the intersection without deceleration or to slow down and cede the right-of-way. The combination of actions selected by all players forms an action profile, which determines the outcome of the game. A utility function then evaluates each outcome based on factors such as efficiency, collision risk, and driving style, assigning a payoff that reflects the player's satisfaction with the result.

3.2. Payoff Function

The utility functions set $U = \{u_1, u_2, \ldots, u_N\}$ forms the model's core, mapping each action profile a to a real number representing player satisfaction. We define three base payoffs [1].

- P_{go}: The base payoff for successfully traversing the intersection.
- P_{Yield}: The time cost incurred for choosing to yield.
- $P_{collision}$: This signifies the significant penalty assigned to any outcome involving a collision.

For any action profile a, the outcome function $\mathrm{R_i(a)}$ assigns player $\mathrm{p_i}$ one of $\{P_{go}, P_{Yield}, P_{collision}\}$ based on conflict rules.

To reflect the behavioral preferences of different driving styles, we introduce a set of personalized weight parameters, w, to adjust these base payoffs, thereby deriving the final utility function, $\mathrm{U_i(a)}$. The driving personality model is established as follows.

- Normal: The utility for a standard player is simply the base payoff, without any personality-based adjustments.

$$\mathrm{U_i(a)} \mid_{\mathrm{type=Normal}} = \mathrm{R_i(a)} \tag{2}$$

- Aggressive: An aggressive player is characterized by a strong aversion to waiting. To model this, we define an aversion weight, $\omega_{\mathrm{agg}} > 1$. When the outcome for such a player is 'Yield', the associated negative utility is magnified by this weight.

$$U_i(a) \mid_{type=Agressive} = \begin{cases} R_i(a) \cdot w_{agg}, \ if \ R_i(a) = P_{yield} \\ R_i(a), \ otherwise \end{cases} \tag{3}$$

- Conservative: A conservative player is more willing to yield for the sake of safety but has an extreme aversion to collisions. We model this behavior using two distinct weights:

https://doi.org/10.3390/engproc2026129002

$$U_i(a) \mid_{type=Conservative} = \begin{cases} R_i(a) \cdot w_{con,yield}, \ if \ R_i(a) = P_{yield} \\ R_i(a) \cdot w_{con,coll}, \ if \ R_i(a) = P_{collision} \\ R_i(a) \end{cases} \quad (4)$$

A conservative agent is characterized by a high tolerance for yielding, modeled by a weight $0 < w_{con,yield} < 1$ that reduces the associated cost, and a strong aversion to collisions, captured by a weight $w_{con,coll} > 1$ that amplifies the penalty.

- Emergency vehicles have the highest mission priority and are far more sensitive to time than ordinary vehicles. To model this, we define a significantly large mission priority weight, $w_{ev} \gg 1$, to amplify the cost associated with yielding.

$$U_{ev}(a) = \begin{cases} R_{ev}(a) \cdot w_{ev}, \ if \ R_{ev}(a) = P_{yield} \\ R_{ev}(a), \ otherwise \end{cases} \quad (5)$$

Through the design of this utility function, we have successfully quantified abstract driving styles into a concrete mathematical model, providing a solid foundation for the subsequent computation of the Nash equilibrium.

3.3. Analytical Payoff Matrix

To achieve an efficient solution for the static game, we depart from the conventional method of repeatedly invoking a microscopic traffic simulator to evaluate payoffs within an iterative loop. Instead, we propose an analytical method to generate the payoff matrix. Using the method, the payoff outcomes are pre-computed and stored for all possible action profiles before the game-solving process begins, thereby constructing a complete lookup table of payoffs (i.e., the payoff matrix). This process consists of the following three steps.

1. Collision Pair analysis (Algorithm 1)

The analytical payoff calculation is based on the a priori determination of geometric conflicts. By analyzing predefined vehicle routes, we identify all pairs of players (p_i, p_j), whose paths physically intersect. These relationships are formalized in a static Collision Pairs set, denoted as C. Consequently, a collision outcome occurs if two players simultaneously select the go action and their pair is an element of C. This approach effectively transforms a complex spatio-temporal judgment into a computationally efficient set-membership query for the payoff algorithm.

Algorithm 1: Calculate Analytical Payoff

Input: Action profile *Profile*, Player types *Types*
Output: A map of payoffs for all players *Payoffs*
Initialization: Define base payoff values: P_{go}, P_{wait}, $P_{collision}$
for each player $p_i \in Profile$ **do**
  $a_i \leftarrow Profile[p_i]$
  if $a_i = Wait$ **then** $u_i \leftarrow P_{wait}$
  else if $\exists p_j \in Profile \setminus \{p_i\}$ s.t. $Profile[p_j] = Go$ **and** $(p_i, p_j) \in$ CollisionPairs **then**
    $u_i \leftarrow P_{collision}$
  else $u_i \leftarrow P_{go}$
  $Payoffs[p_i] \leftarrow$ Adjust u_i based on $Types[p_i]$
end for
return *Payoffs*

https://doi.org/10.3390/engproc2026129002

2. Payoff calculation algorithm (Algorithm 2)

The calculate-analytical-payoff algorithm, presented in Algorithm 1, was developed to compute the final utilities for all players under a given action profile.

Algorithm 2: Build Payoff Matrix

Input: Players P, Types T, Available actions A
Output: $PayoffMatrix$
Initialize an empty map $PayoffMatrix$
$\mathcal{S}$ Generate all action profile combinations from A^P
foreach $s \in \mathcal{S}$ **do**
  $key \leftarrow$ String representation of s
  $PayoffMatrix[key] \leftarrow$ CalculateAnalyticalPayoff(s, T)
end for
return $PayoffMatrix$

3. Payoff matrix construction (Algorithm 3)

Before solving for the Nash equilibrium, a complete payoff matrix is pre-computed using the build-payoff-matrix function, detailed in Algorithm 2. This procedure exhaustively enumerates all 24 = 16 possible action profiles and invokes the calculate-analytical-payoff algorithm for each to determine the corresponding utility vector, $U(a) = \{u_1(a), u_2(a), \ldots, u_N(a)\}$.

Algorithm 3: Main Control Flow for Static Game Analysis

Input: Players $\mathcal{P}$, Player types $\mathcal{T}$, Available actions $\mathcal{A}$
Output: Nash Equilibrium profile NE
Part I: Offline Solving
$PM \leftarrow$ BuildPayoffMatrix$(\mathcal{P}, \mathcal{T}, \mathcal{A})$
$NE \leftarrow$ FindNashEquilibrium$(\mathcal{P}, \mathcal{A}, PM)$
Part II: Online Simulation
Connect to SUMO environment; Initialize controlled set $\mathcal{C} \leftarrow \varnothing$
while simulation is active **do**
  $\mathcal{V}$ Update and retrieve active vehicles from SUMO
  foreach player $p_i \in (\mathcal{P} \cap \mathcal{V}) \setminus \mathcal{C}$ **do**
    Execute $NE[p_i]$ in SUMO (Set v_{max} if "Go", else $v = 0$)
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{p_i\}$
  end for
end while
Close SUMO connection and **return** NE

3.4. Nash Equilibrium Solution

The pure strategy Nash equilibrium is identified using the IBR algorithm, an iterative procedure that emulates rational players sequentially maximizing their utility. In each iteration, a player determines their best response to the action that yields the highest payoff given the current strategies of all other players.

Formally, for player p_i, given a fixed action profile of all other players, $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N)$, the best response $BR_i(a_{-i})$ is an action $a^* \in A_i$.

$$a_i^* = \arg \max_{a^* \in A_i} u_i(a_i, a_{-i}) \tag{6}$$

https://doi.org/10.3390/engproc2026129002

Here, a_i^* represents the optimal action chosen from the feasible action space A_i. The term $u_i(a_i, a_{-i})$ denotes the utility function of player p_i, which quantifies the payoff derived from taking action a_i while the other players maintain their strategies a_{-i}.

4. Experiment

To validate the proposed static game decision algorithm, its performance was evaluated in a standard unsignalized intersection using a MATLAB–SUMO co-simulation via the TraCI interface. The evaluation was conducted to compare the algorithm against a baseline model across various combinations of driving styles.

4.1. Simulation Environment and Tools

MATLAB R2023a was used for the game model, Nash equilibrium computation, and aster simulation control. The traffic simulator was SUMO v1.22.0 for microscopic simulation. TraCI4Matlab was employed as an interface for MATLAB–SUMO communication.

4.2. Scenario Setup

The simulated environment is configured as a standard signal-free intersection comprising four two-way roads, each featuring one inbound lane and one outbound lane. Right-of-way is entirely determined by vehicles' decentralized decision-making mechanisms. At simulation start, four vehicles with predefined roles and routes depart simultaneously to generate traffic flow. To model behavioral heterogeneity, SUMO's vType parameter was configured to represent distinct driving personalities, as shown in Table 1. These parameters, which directly influence vehicle dynamics and decision-making, were calibrated to align with the utility functions of the proposed game theory model [4].

Table 1. SUMO vehicle type parameters for different driving styles.

	Acceleration (m/s^2)	Deceleration (m/s^2)	Sigma	Tau	Maximum Speed (m/s)
Aggressive	4.0	5.5	0.7	1.0	60
Conservative	2.2	4.0	0.3	2.0	40
Normal	2.9	2.9	0.5	1.5	50
Emergency	5.0	6.0	0.2	0.8	70

4.3. Game Model Configuration

The static game model is defined by the following components: The content below represents only one scenario within a randomized experiment. The player set comprises four vehicles in the scenario: $P = \{EV, NV_\{NE\}, NV_\{SW\}, NV_\{ES\}\}$ (Figure 1), with each participant's action space defined as $\mathrm{A_i} = \{\mathrm{Go}, \mathrm{Yield}\}$ The participant reward function is determined by Algorithm 1, which assigns values for successful passage (P_{go}): +10, Yielding (P_{yield}): −1, collision ($P_{collision}$): −1000. Based on prior path analysis, the set of conflicting vehicle pairs is predefined as $\mathrm{C} = (\mathrm{EV}, \mathrm{NV_{NE}}), (\mathrm{NV_{NE}}, \mathrm{NV_{ES}})$. Finally, the base reward values are adjusted using style-dependent weights (Table 2) to incorporate player heterogeneity factors.

Table 2. Driving style parameters.

	Yield Cost Multiplier ($\omega_{\mathbf{yield}}$)	Collision Penalty Multiplier ($\omega_{\mathbf{collision}}$)
Aggressive	2.0	1.0
Conservative	0.5	2.0
Emergency	10.0	1.0

https://doi.org/10.3390/engproc2026129002

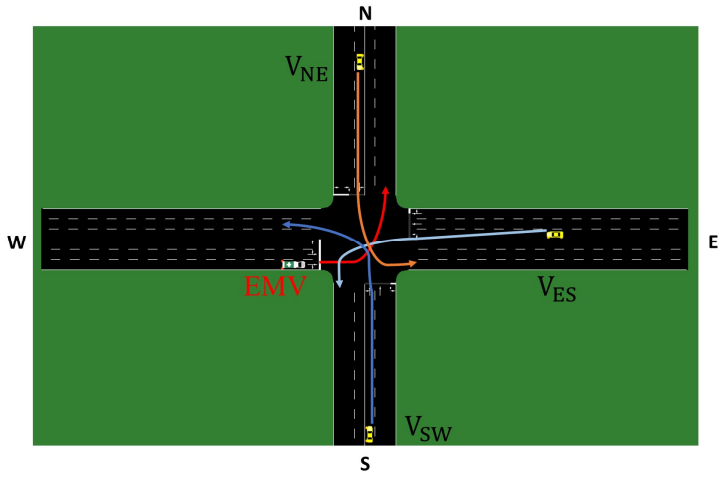

Figure 1. The unsignalized intersection scenario with predefined vehicle trajectories.

Experimental cases include the following.

Case 1 (mixed): Baseline performance verification in a typical mixed-style setting.
Case 2 (homogeneous): All non-EV vehicles share one style: (a) aggressive, (b) normal, (c) conservative—used for robustness/equilibrium stress tests (Table 3).

Table 3. Driving personality composition for each experimental case.

	$\mathbf{NV_{NE}}$	NV_{SW}	NV_{ES}
Case 1.	Normal	Aggressive	Conservative
Case 2a.	Aggressive	Aggressive	Aggressive
Case 2b.	Normal	Normal	Normal
Case 2c.	Conservative	Conservative	Conservative

Across metrics, the algorithm reduces average travel time in all scenarios except Case 2a, where a slight increase is traded for markedly better EV safety distance (Figure 2). Speed profiles show the comparison of all four vehicles under the Game-Theoretic Algorithm vs. Default SUMO (Figure 3) [3].

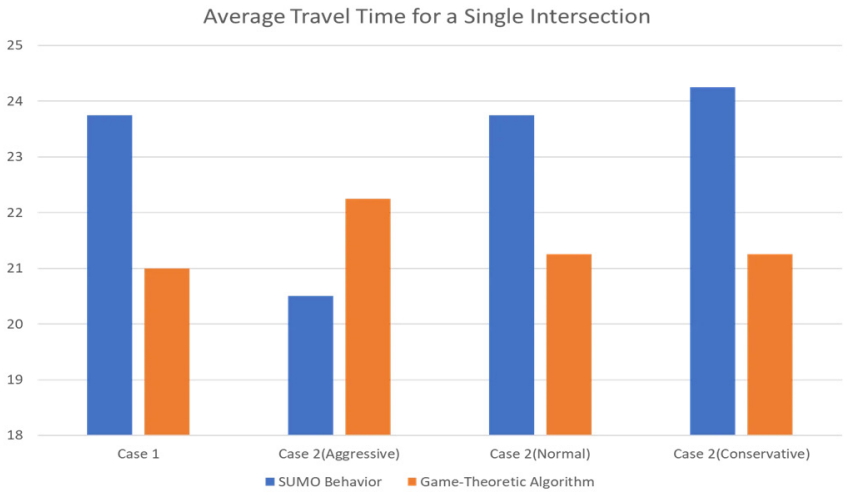

Figure 2. Average travel time for a single intersection.

https://doi.org/10.3390/engproc2026129002

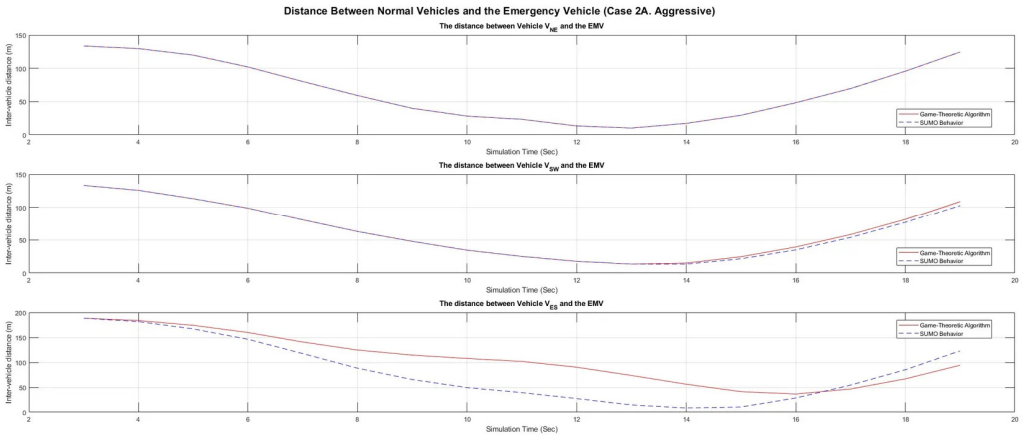

Figure 3. Comparison of safety distances to the emergency vehicle in Case 2a (all-aggressive scenario).

5. Conclusions

We presented a static game-theoretic decision algorithm for an unsignalized intersection in a MATLAB–SUMO co-simulation. Across evaluated cases, it reduces average travel time relative to SUMO's default behavior in all scenarios except Case 2a, where a slight increase is exchanged for improved EV safety distance. Speed profiles corroborate these trends across the four vehicles.

Author Contributions: Conceptualization, I.-H.L., C.-F.L. and W.-X.L.; methodology, I.-H.L. and W.-X.L.; software, W.-X.L. and K.-T.L.; validation, W.-X.L. and K.-T.L.; investigation, W.-X.L. and K.-T.L.; writing—original draft preparation, W.-X.L.; writing—review and editing, I.-H.L.; supervision, I.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science and Technology Council (NSTC) grant number NSTC 114-2634-F-006-001-MBK and NSTC 114-2222-E-006-005.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available within the article.

Acknowledgments: This research was supported by the National Science and Technology Council (NSTC) of Taiwan under grants NSTC 114-2634-F-006-001-MBK and NSTC 114-2222-E-006-005.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, N.; Kolmanovsky, I.; Girard, A.; Yildiz, Y. Game Theoretic Modeling of Vehicle Interactions at Unsignalized Intersections and Application to Autonomous Vehicle Control. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018.
2. Cao, Y.; Zeng, X.; Yin, Z. A Game Theoretic Decision-Making Framework With Conflict-Aware Nash Equilibrium Selection for Autonomous Vehicles at Uncontrolled Intersections. *IEEE Trans. Intell. Transp. Syst.* **2025**, *1*, 210–224. [CrossRef]
3. Asadi, M.; Fathy, M.; Mahini, H.; Rahmani, A.M. An Evolutionary Game Approach to Safety-Aware Speed Recommen-dation in Fog/Cloud-Based Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *7*, 7431–7440. [CrossRef]
4. Lu, X.; Zhao, H.; Li, C.; Gao, B.; Chen, H. A Game-Theoretic Approach on Conflict Resolution of Autonomous Vehicles at Unsignalized Intersections. *IEEE Trans. Intell. Transp. Syst.* **2023**, *11*, 12535–12548. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.