UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

A Decentralized Approach for Connected and Autonomous Vehicles Traffic Negotiation and Conflict Resolution

Tiago Cesar dos Santos

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Tiago Cesar dos Santos

A Decentralized Approach for Connected and Autonomous Vehicles Traffic Negotiation and Conflict Resolution

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
June 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

```
        dos Santos, Tiago Cesar
dD722d     A Decentralized Approach for Connected and
        Autonomous Vehicles Traffic Negotiation and
        Conflict Resolution / Tiago Cesar dos Santos;
        orientador Denis Fernando Wolf. -- São Carlos, 2023.
           152 p.


           Tese (Doutorado - Programa de Pós-Graduação em
        Ciências de Computação e Matemática Computacional) --
         Instituto de Ciências Matemáticas e de Computação,
        Universidade de São Paulo, 2023.


           1. Decentralized negotiation. 2. Connected
        Autonomous Vehicles. 3. Multi-objective. 4.
        Collective Probabilities. 5. Autonomous Vehicles.
        I. Wolf, Denis Fernando, orient. II. Título.
```

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Tiago Cesar dos Santos

Uma Abordagem Descentralizada para Negociação de Tráfego de Veículos Conectados e Autônomos e Resolução de Conflitos

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
Junho de 2023

Dedicated to my family
Dora, Silas, Julio, Giseli, Pietro
and Marcela
In memoriam of Bacalhau

ACKNOWLEDGEMENTS

I would like to start by expressing my gratitude to my family, who have made this accomplishment possible.

I would like to thank my advisor Denis Wolf for persisting and giving the necessary support to complete this work. He also allows me to work on several important projects at the Mobile Robotics Laboratory (LRM).

I would like to express my deepest gratitude to the University of São Paulo community for their assistance in various situations. To LRM's professors for their cooperation in several works and for sharing knowledge all of these years.

Words cannot express my gratitude to my friends at the Laboratory for the exchange of experiences, support, conversations, and important moments that will always be remembered.

I am grateful to Dr. Charles Philippe and Dr. Ishaan R. Kale for their contributions in sharing knowledge about the method used in this work.

Finally, I could not have undertaken this journey without my girlfriend Marcela for her support over the last years, which was fundamental for the completion of this work.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

“Imagination is more important
than knowledge.”
(Albert Einstein)

RESUMO

SANTOS, T. C. **Uma Abordagem Descentralizada para Negociação de Tráfego de Veículos Conectados e Autônomos e Resolução de Conflitos**. 2023. 152 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

O crescimento populacional e a rápida expansão urbana no último século geraram uma aglomeração metropolitana que causou inúmeros impactos sociais e econômicos. A alta demanda por mobilidade resultou em um número crescente de veículos nas estradas, levando a tempos de viagem mais longos, o que tem causado congestionamentos, aumento dos índices de acidentes e aumento da poluição. Os Sistemas de Transporte Inteligente (ITS) têm o potencial de revolucionar o transporte e a mobilidade urbana ao reduzir acidentes de trânsito, mitigar congestionamentos e engarrafamentos.

A comunicação entre veículos permite negociação e cooperação que podem ser usadas em vários contextos, como cruzamentos, junções de faixa e mudança de faixa. A resolução de conflitos entre veículos é uma tarefa desafiadora devido à maior densidade de veículos e espaço de manobra limitado, muitas vezes apenas mudanças de velocidade é permitido. Um número significativo de estudos focam em soluções centralizadas que possuem uma forte dependência da infraestrutura. Além disso, há poucas pesquisas sobre métodos de negociação descentralizada que levem em consideração sistemas multiobjetivos onde cada veículo tem sua própria função de custo privada e também existem poucos trabalhos focados em soluções generalizadas que abordam diferentes cenários dinâmicos com fluxo de tráfego usando a mesma abordagem.

Esta tese propõe um método para resolução de conflitos entre veículos que possuem a capacidade de se comunicar de forma descentralizada e que pode ser utilizado em múltiplos cenários dinâmicos, tais como: cruzamentos, junções de faixa e mudanças de faixa. Também consideramos veículos com diferentes características, dimensões e funções de custo privadas. A abordagem proposta visa melhorar o fluxo de veículos e reduzir o tempo de viagem que pode ser traduzido na redução de emissão de poluentes.

Para avaliar a proposta, utilizamos uma ferramenta de simulação para criar diversos cenários de teste e comparamos os resultados com métodos existentes utilizando métricas comumente utilizadas na literatura. Além disso, também foi necessário adaptar a ferramenta de simulação para incluir recursos como cálculo de trajetória, detecção de colisão e gerenciamento de veículos em relação ao processo de tomada de decisão.

Em todos os cenários, melhorias significativas foram alcançadas em termos de fluxo de veículos, resultando em diminuição do tempo de viagem e aumento da velocidade média em relação aos benchmarks utilizados.

Palavras-chave: Negociação descentralizada, Veículos Autônomos Conectados, Multi-objetivo, Probabilidades Coletivas.

ABSTRACT

SANTOS, T. C. **A Decentralized Approach for Connected and Autonomous Vehicles Traffic Negotiation and Conflict Resolution**. 2023. 152 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Population growth and fast urban expansion in the last century generated a metropolitan agglomeration that caused numerous social and economics impacts. The high demand for mobility has resulted in a growing number of vehicles on the roads, leading to longer travel times, which has caused traffic congestion, increased accident rates, and increased pollution. Intelligent Transportation Systems (ITS) have the potential to revolutionize the transportation and urban mobility to reduce traffic accidents, mitigate congestion and traffic jams.

The communication between vehicles allows negotiation and cooperation that can be used in various contexts such as crossing intersections, on-ramp merge and lane change. Resolving conflicts between vehicles is a challenging task due to the higher density of vehicles and limited maneuverability, often limited to only speed changes. A significant number of studies focus on solutions rely on centralized methods that have a heavy dependency on the infrastructure. Additionally, there is limited research on decentralized negotiation methods that take into account multi-objective systems where each vehicle has its own private cost function and also there is a lack of work in generalized solutions that tackles different dynamic scenarios with traffic flow using the same approach.

This thesis proposes a method for resolving conflicts between vehicles that have the ability to communicate in a decentralized manner that can be used in multiple dynamic scenarios, such as: crossing intersections, on-ramp merging and changing lanes. We also considered vehicles with different characteristics, dimensions and private cost functions. The proposed approach aims to improve the flow of vehicles and reduce travel time that can be translated into a reduction in the emission of pollutants.

To evaluate our proposal, we use a simulation tool to create various test scenarios and compare our results with existing methods using metrics commonly used in the literature. In addition, it was also necessary to adapt the simulation tool to include features such as trajectory calculation, collision detection and vehicle management in relation to the decision-making process.

In every scenario, significant improvements were achieved in terms of vehicle flow, resulting in a decrease in travel time and an increase in average speed when compared to the benchmarks used.

Keywords: Decentralized negotiation, Connected Autonomous Vehicles, Multi-objective, Collective Probabilities.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ADAS	Advanced Driver Assistance Systems
AI	Artificial Intelligence systems
AIM	Autonomous Intersection Management
API	Application Programming Interface
ATC	Air Traffic Control
AV	Autonomous Vehicles
BCAS	Beacon Collision Avoidance System
CACC	Cooperative Adaptive Cruise Control
CALM	Communications Access for Land Mobiles
CAVs	Connected and Autonomous Vehicles
CMP	Collaborative Maneuver Protocol
DARPA	Defense Advanced Research Projects Agency
DSRC	Dedicated Short-Range Communications
FAA	Federal Aviation Administration
FCFS	First Come First Serve
IEEE	Institute of Electrical and Electronics Enginners
IPPCA	Iterative Peer-to-Peer Collision Avoidance
ISO	International Organization for Standardization
ITS	Intelligent Transportation Systems
MAS	Multi-Agent Systems
MILP	Mixed Integer Linear Programming
NSGA-II	Non-dominated Sorting Genetic Algorithm
PC	Probability Collectives
RSU	Road Side Unit
SAE	Society of Automotive Engineers
SAT	Separating Axis Theorem
SUMO	Simulation of Urban MObility
TCAS	Traffic Collision Avoidance System
TLS	Traffic Light System
TraCI	Traffic Control Interface
V2I	Vehicle-To-Infrastructure

V2V	Vehicle-To-Vehicle
V2X	Vehicle-To-Everything
VANETs	Vehicular Ad hoc NETworks
WAVE	Wireless Access in Vehicular Environments

CONTENTS

CHAPTER 1

INTRODUCTION

Population growth and fast urban expansion in the last century generated a metropolitan agglomeration that caused numerous social and economic impacts. The high demand for mobility has resulted in a growing number of vehicles on the roads, leading to longer travel times, which has caused traffic congestion, increased accident rates, and increased pollution (ENGLUND *et al.*, 2015).

Intelligent Transportation Systems (ITS) have the potential to revolutionize the transportation system. Their main goal is to use the knowledge of Engineering, Computing, and Information Technologies to reduce traffic accidents, mitigate congestion and traffic jams, reduce travel time, and decrease pollution emissions, resulting in environmental sustainability, economic efficiency, and social welfare (MALIK *et al.*, 2020) (KHAYATIAN *et al.*, 2020).

Many of these transformations have occurred since the 1950s with the advent of seat belts and airbags. Over the past few decades, the world has seen a stunning transformation in the car industry. The manufacturers, in collaboration with the academy, invested in Advanced Driver Assistance Systems (ADAS) such as blind spot detection, lane keeping, cruise control and parking assistance (KHAYATIAN *et al.*, 2020) (MAKSIMOVSKI; FESTAG; FACCHI, 2021).

Most of these ADAS technologies emerged from initiatives in the 1980s with projects related to Autonomous Vehicles (AV) (THORPE *et al.*, 1988) (CRISMAN; THORPE, 1993) (BROGGI *et al.*, 1999). Two decades later, in 2004 and 2005, the Defense Advanced Research Projects Agency (DARPA) promoted the first competition of AV, known as the DARPA Challenge, where vehicles had to perform a fully autonomous route in the desert (THRUN *et al.*, 2006). In 2007 the same challenge was proposed, but in an urban setting (URMSON *et al.*, 2008). These competitions showed the technical viability of an AV in a controlled environment. Some companies started initiatives to make AV commercially viable a few years later.

In 2011, the Netherlands hosted the Grand Cooperative Driving Challenge (GCDC), which aimed to showcase the capabilities of connected vehicles (NUNEN *et al.*, 2012). The main focus of the competition was on merging two platoons in the same lanes, with participants tasked with developing and implementing communication and Cooperative Adaptive Cruise Control (CACC) technologies to control the longitudinal velocity and ensure a smooth merger.

In 2016, the Interoperable GCDC AutoMation Experience (i-GAME) (ENGLUND *et al.*, 2016) (PLOEG *et al.*, 2018) introduced a series of additional challenges. The first challenge involved lane-merging problem, where two platoons had to safely merge into a single lane due to roadworks. The second challenge focused on automated T-junction interactions, and the final challenge involved a special vehicle requesting priority passage. Overall, the i-GAME competition aimed to test the capabilities of connected and automated vehicles in realistic scenarios.

AVs typically use radars, lidars, camera, and stereo cameras for sensing and Artificial Intelligence systems (AI), which allow the vehicle to perform a local scan of its surroundings. However, this perception is limited by the physical capabilities of the sensors (ENGLUND *et al.*, 2015). Vehicle-to-vehicle communication might be essential in improving traffic efficiency as it can overcome the sensors' range and occlusion limitations (MAKSIMOVSKI; FESTAG; FACCHI, 2021).

Communication also allows negotiation, cooperation, and organization of strategies together that can be used in various contexts, such as platooning, cooperative merging at ramps, automated roundabout management, cooperative lane changing, and intersection management (KHAYATIAN *et al.*, 2020). Instead of making predictions of a vehicle's planning or intentions, it is possible to exchange accurate information regarding driving intentions and planning algorithms (MAKSIMOVSKI; FESTAG; FACCHI, 2021).

The concept of communication and negotiation between vehicles and infrastructure is a recent development in the field of land vehicles. However, this subject has been researched for decades in the aviation field, and it is already used in the aviation industry. The 1956 Grand Canyon air disaster, caused by a communication failure between the airplanes and the Air Traffic Control (ATC), led to the development of a collision avoidance system in aviation.

In the early stages of development, it was noticed that non-cooperative methods did not have satisfactory results, leading to the development of cooperative systems. The Beacon Collision Avoidance System (BCAS) was developed in the mid-1970s. Using a transponder, it could detect other aircraft in the surrounding airspace, but it generated many false alarms. In 1978, an accident in San Diego forced the Federal Aviation Administration (FAA) to accelerate the development of collision avoidance systems. In 1981, the FAA decided to develop and implement the Traffic Collision Avoidance System (TCAS) using the basic design of the BCAS. The TCAS system utilizes a transponder to interrogate and track aircraft, providing Traffic Advisories for a visual search of other aircraft and Resolution Advisories that recommend maneuvers to separate aircraft. It is worth to mentioning that the TCAS system is an autonomous and decentralized system that communicates with the surrounding aircraft. Based on a system of rules, it finds a set of maneuvers informed and executed by the pilots. Therefore, the final stage is not autonomous.

The latest version of TCAS II (7.1) was designed to operate in traffic densities of up to 0.3 aircraft per square nautical mile (nmi), which means 24 aircraft within a 5 nmi radius (9.26 km), which was expected to be the highest traffic density envisioned over the next 20 years (FAA, 2011).

Unlike conflict resolution between aircraft, resolving conflicts between ground vehicles is more challenging due to the higher density of vehicles and limited maneuverability, often limited to only speed changes.

In the ground vehicles field, many studies focus on crossing intersections. Many of these solutions rely on centralized methods, as demonstrated by the surveys by Chen and Englund (2016) and Khayatian *et al.* (2020). The problem with centralized approaches is that they heavily depend on the infrastructure (LIU *et al.*, 2018).

Maksimovski, Festag and Facchi (2021) surveyed decentralized solutions and showed that decentralized is a research trend that can solve problems only using communication between vehicles without the need for infrastructure. Decentralized approaches are still an unexplored area in the context of connected vehicles. It is also possible to notice a lack of work related to a generalized solution that tackles different and dynamic scenarios with traffic flow using the same approach.

1.1 Objectives

The main goal of this thesis is to develop a method for resolving conflicts between vehicles that have the ability to communicate in a decentralized manner that can be used in multiple dynamic scenarios, such as: crossing intersections, on-ramp merging and changing lanes. Therefore, our proposal is generalized and addresses different problems. It also considers that vehicles are agents with different private cost functions, different dimensions and random speeds, resulting in a multi-objective system.

To evaluate the approach proposed, it was necessary to use a simulation tool to create various test scenarios and compare our results with existing methods using metrics commonly used in the literature, such as average travel time, average speed, average waiting time, number of arrived vehicles, and flow. In addition, it was also necessary to adapt the simulation tool to include features such as trajectory calculation, collision detection and vehicle management in relation to the decision-making process.

The proposed method aims to improve the flow of vehicles and reduce travel time that can be translated into a reduction in the emission of pollutants. The method will be evaluated using metrics commonly used in similar studies and will be compared to methods that represent a benchmark.

1.2 Thesis Structure

Chapter 1 provides an overview of the research topic, background information, and a statement of the research problem and its significance to the ITS field.

Chapter 2 presents a comprehensive review of the existing literature related to the research topic. It highlights the current state of knowledge in the field, the gap in the literature that the research aims to fill, and the theoretical framework used in the study.

Chapter 3 presents the Probability Collectives method in its original format and our modification to work with multi-objective systems. In Chapter 3.2, we present the adaptations of the method necessary for the context of connected vehicles, the mathematical formulations, and the algorithm for resolving conflicts between vehicles. We also present some support functions for the method to work with vehicles of different sizes and private cost functions.

Chapter 4 presents the simulation tool, provides definitions for vehicles and parameters, and the development of support tools for our experiments. It also presents the definition of our scenarios.

Chapters 5 and 6 present results for static and dynamic scenarios, respectively. Chapter 7 summarizes the key findings and contributions of the research and provides recommendations for future work in the field. Finally, Appendix A includes additional experiments with multi-objective benchmark functions, which demonstrate the general applicability of our proposal and can be used in other contexts.

CHAPTER 2

RELATED WORKS

2.1 Connected and Autonomous Vehicles

Recent advance in the fields of communication and networking have led to the development of new methods and systems related to Vehicular Ad hoc NETworks (VANETs). According to Yousefi, Mousavi and Fathy (2006), VANETs are ad-hoc networks that do not rely on a fixed infrastructure and in which there are several mobile elements such as vehicles and pedestrians and also fixed elements such as Road Side Unit (RSU). The concept of VANETs allows bidirectional communications between vehicles that is generally classified as Vehicle-To-Vehicle (V2V) and Vehicle-To-Infrastructure (V2I) communication. Some authors have also defined Vehicle-To-Everything (V2X) as bidirectional communication between vehicles and any other road element. This technology represents a promising solution for the transportation system to reduce accidents, congestion, and pollution.

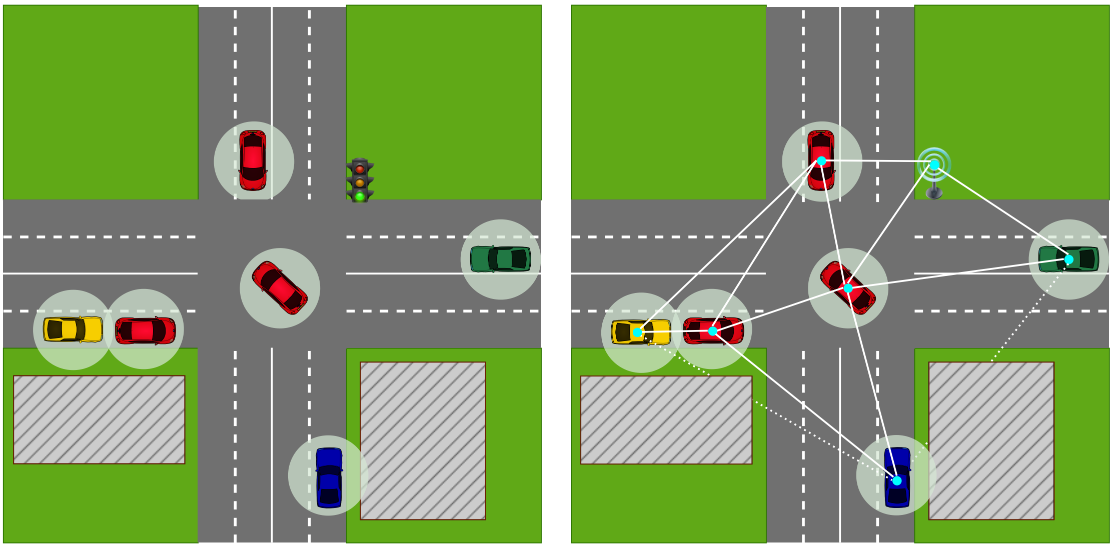
In practice, there are already some protocols to make possible the implementation of a VANET. The Institute of Electrical and Electronics Enginners (IEEE) published the protocol Wireless Access in Vehicular Environments (WAVE) which uses IEEE 802.11p based on Dedicated Short-Range Communications (DSRC), which are short-range wireless communication channels designed for vehicles, as shown by Kenney (2011). The International Organization for Standardization (ISO) also proposed the Communications Access for Land Mobiles (CALM). The Society of Automotive Engineers (SAE) specified standards to be used in communication such as the SAE J2735 DSRC, which has a series of customized messages for vehicles such as the BasicSafetyMessage exclusive for V2V. This message contains vehicle identification, size, latitude, longitude, elevation, acceleration, and speed information.

Connected and Autonomous Vehicles (CAVs) are vehicles equipped with advanced technologies, such as on-board sensors (Lidars, Radars, and cameras) and wireless communication capabilities, to enable them to operate not only autonomously, but also communicate with external agents such as other vehicles and RSUs, making them an element of VANETs. Therefore, in addition to use the information in its surroundings, the vehicle can communicate and cooperate with distant agents to enhance overall traffic efficiency. In this work, we use the terms CAV and agent interchangeably to refer to entities (intelligent vehicles) that have the ability to perceive and act on their environment, as defined by Russell and Norvig (2010).

Figure 1 illustrates the difference in functionality between AVs and CAVs. The circle surrounding the vehicle represents the range of the sensors (not in scale). In Figure 1a, we can see that the yellow vehicle is close to the red vehicle ahead, resulting in a limited view of the intersection due to occlusion. In this scenario, the traffic light coordinates the intersection, but the vehicles must use machine learning techniques to detect and interpret the traffic light's status. Finally, AVs must also detect and infer the actions of other AVs to perform decision-making.

In the case illustrated in Figure 1b, we present a cooperative scenario using communication and AVs, in which this combination represents CAVs. The traffic light was replaced by a V2I system to transmit infrastructure information, and the vehicles maintain communication by informing data about their positions, speeds, and actions through V2V communication. We can notice a specific case where the buildings (grey hatched structure) can attenuate or prevent communication between yellow and blue and blue, and green vehicles. This limitation is overcome by the network itself, which can perform a broadcast and re-transmit information from other vehicles.

Figure 1 – Autonomous Vehicles and Connected Autonomous Vehicles



(a) AVs.

(b) CAVs

Source: Elaborated by the author.

According to the study by Tientrakool, Ho and Maxemchuk (2011), AVs can increase traffic efficiency by 43%, while CAVs can increase traffic efficiency by 273% by increasing the capacity of vehicles on the highway. Research by Arem, Driel and Visser (2006) and Talebpour and Mahmassani (2016) show that CAVs can improve traffic efficiency by decreasing string stability using an approach known as Cooperative Adaptive Cruise Control (CACC). String stability is a propagation effect of an action performed by the platoon leader.

VANETs enable vehicles to share information such as georeferenced position, speed, trajectories, and actions, which makes it possible to implement several applications such as collision avoidance systems for intersections, warning of risks of lateral collisions during lane changes and merging, as stated by Karagiannis *et al.* (2011). Additionally, VANETs can support logistics and general internet services, as cited by Karagiannis *et al.* (2011).

2.2 Applications and use cases of VANETs

This section provides a review of relevant works on applications and uses cases of VANETs discussed in this thesis. We explore various approaches employed by authors to address the associated challenges Intersection Management, Ramp Merge, and Lane Change.

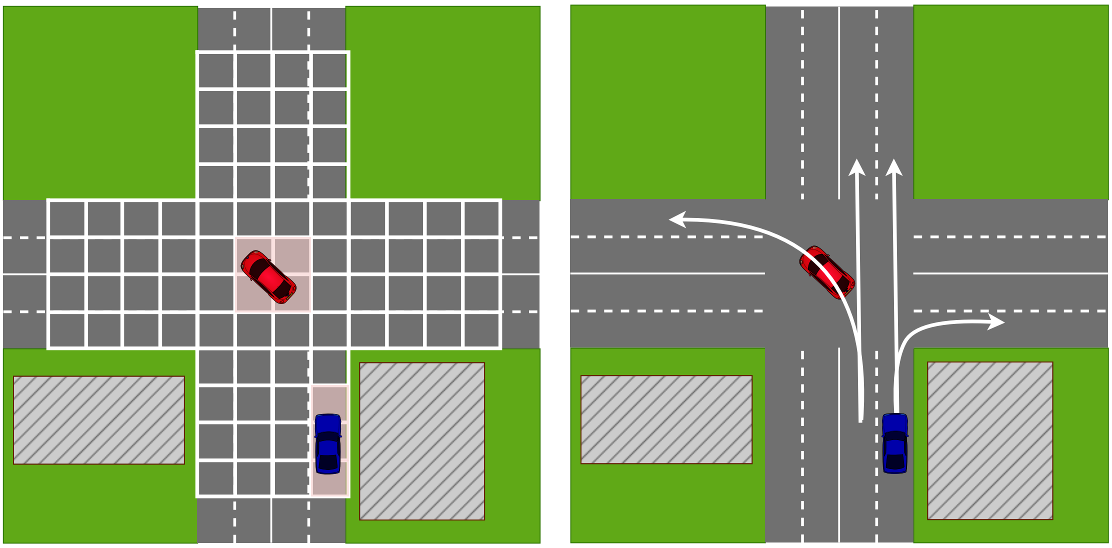
2.2.1 Intersection Management

Intersection Management is one of the most studied applications in ITS because a significant portion of traffic accidents occurs in intersections and it is also a conflict zone that can cause traffic jams and congestions. Chen and Englund (2016) defined the intersection as a shared resource that require scheduling for passage. They also classified the intersections in two types: signalized and non-signalized. The first one are managed with traffic lights, while the second are managed by drivers using road signs information.

Intersection Management has been extensively surveyed by several studies, such as (CHEN; ENGLUND, 2016), (KHAYATIAN *et al.*, 2020), and (GHOLAMHOSSEINIAN; SEITZ, 2022), which focus mainly on modeling and approaches for non-signalized intersections. The two main approaches to modeling an intersection are: space discretization and trajectory modeling. In the first approach, the intersection is implemented as an occupancy grid in which vehicles allocate space resources at specific times. In the second approach, vehicles are modeled as following pre-defined trajectories and an overlap test is performed over time. In addition, Chen and Englund (2016) mentions a third way that would be a combination of space discretization and trajectory modeling, this would be collision region modeling where only the conflict region should be evaluated.

Figure 2a represents the space discretization that determines an occupancy grid, spaces are also called cells or tiles. Vehicles allocate spaces according to their dimensions respecting the size of the cells. In Figure 2b the trajectory model is presented in which the vehicle must follow a predefined path during a period of time. Generally, in these regions, lane changes are not defined since the paths must be followed.

Figure 2 – Intersection modelling



(a) Space discretization (b) Vehicles' trajectory

Source: Elaborated by the author.

Chen and Englund (2016) classified the methods in Resource Allocation and Trajectory Modeling considering centralized and distributed methods. Khayatian *et al.* (2020) divided the methods in centralized and distributed, and also organized the scheduling policy in First Come First Serve (FCFS) approaches, optimizations and heuristics methods. Gholamhosseinian and Seitz (2022) categorized the methods in a more conceptual way from the point of view of safety, efficiency and environment.

Dresner and Stone (2004) proposed one of the first methods related to Resource Allocation, a reservation-based intersection control policy in the context of Multi-Agent Systems (MAS) using discretized intersection modeling in a centralized manner. It compares three different intersection control mechanisms: overpass, traffic light, and reservation-based.

In the overpass method vehicles traveling in orthogonal directions can pass through one another without stopping, which can help reduce delays and improve traffic flow. The traffic light uses signals to control when vehicles can pass through the intersection, it is an approximation of the actual traffic signals used in real scenarios. The reservation-based allows vehicles to reserve time slots at intersections, which can help improve traffic flow and reduce delays. The authors showed that the reservation-based system outperforms the traffic light method and approaches the efficiency of the overpass method.

The limitations found concern the inability of vehicles to turn and the constraint that vehicles may not change their speed while in the intersection. To overcome these limitations, Dresner and Stone (2005) proposed an intersection control policy protocol to handle complex situations such as turns and intersection accelerations. Heuristics manages the reservation protocol to calculate the parameters. This protocol enhanced the system by allowing vehicles to turn, allowing acceleration within intersections, and improving their interaction capabilities through a detailed protocol.

Azimi *et al.* (2014) introduces Spatio-Temporal Intersection Protocols for autonomous vehicles, which are designed to increase safety and reduce traffic delays at intersections. The intersection area is modeled as a grid that is divided into small cells. Each cell in the grid is associated with a unique identifier. The protocols enable cooperative driving among approaching vehicles to ensure their safe passage through the intersection and describes the FCFS method as a ticket scheduling system in which vehicles are ordered by arrival time at the intersection, with the shortest arrival time having the highest priority.

The authors divided the approach into two protocols, Minimal Concurrency Protocols, and High Concurrency Protocols. The first protocol enables higher-priority vehicles to cross the intersection without slowing down, while lower-priority vehicles come to a complete stop and wait for an exit message from the higher-priority vehicle before proceeding. The second protocol aims to increase parallelism in the intersection by allowing more vehicles to cross simultaneously. It allows conflicting vehicles to navigate through the intersection by having lower-priority vehicles stop before entering conflicting areas until higher-priority vehicles have safely passed.

Levin and Rey (2017) proposed an Autonomous Intersection Management (AIM) based on reservation timings and use a conflict point approach to vehicle separation to formulate a Mixed Integer Linear Programming (MILP). The AIM protocol can only choose from the reservations from the vehicles' requests. It is used to specify intersection-optimized vehicle trajectories and has been developed as a prerequisite to deploying the conflict point MILP. The protocol has a limited ability to admit arbitrary vehicle trajectories, which led to the development of a new protocol called AIM*. In AIM*, unlike the previous one, the intersection manager specifies reservations for each vehicle using a conflict point approach to vehicle separation. The authors conclude that AIM* performs much better than AIM with FCFS policy.

Dongxin *et al.* (2021) proposed the coordination of passage at the non-signalized intersection using predefined trajectories coordinated through a priority tree based on a Dynamic Scheduling Algorithm and on an online formed priority tree. They also discretized the intersection center into sections used to calculate conflicts. In this approach, the tree is built according to the priority of the vehicles, waiting vehicles and based on the traffic regulations and their estimated arrival time.

2.2.2 Ramp Merge

The on-ramp scenario is a conflict zone similar to an intersection in which vehicles move in approximately the same direction as depicted in Figure 3. The objective is to execute a safe lane change maneuver, effectively merging the vehicles into a single lane.

A study by Zhao *et al.* (2019) surveyed and categorized various works related to the ramp merge scenario into two categories: centralized and distributed. In centralized systems, control is handled by a roadside unit or transportation management center. In contrast, in distributed systems, vehicles have V2V capacity used to find a solution managed by the vehicles themselves. According to the study, distributed systems have better scalability.

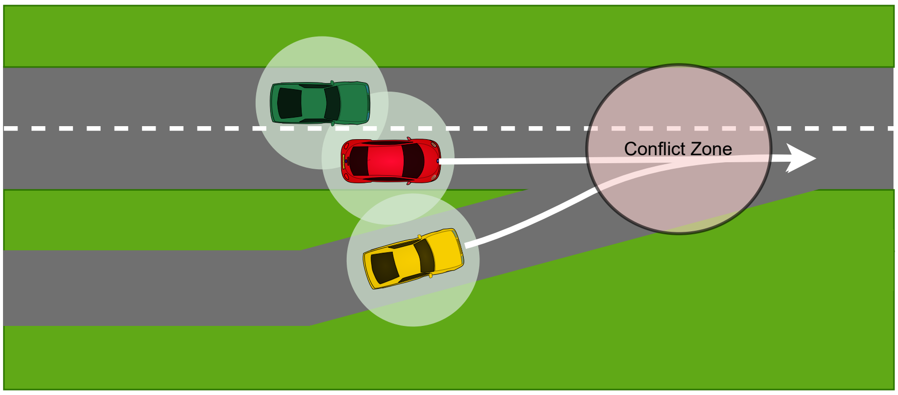
The hierarchical architecture is designed to coordinate ramp control across a corridor with mixed traffic conditions, incorporating both infrastructure-based surveillance and CAVs technology to estimate traffic states, model driving behavior, and optimize ramp metering.

Rios-Torres and Malikopoulos (2016) addressed the problem of on-ramp merge as an unconstrained optimal control problem and applied Hamiltonian analysis to derive an analytical, closed-form solution. The solution allows for online coordination of vehicles at the merging zone, while ensuring collision avoidance and a smooth traffic flow without stop-and-go driving.

Eiermann *et al.* (2020) proposed a Collaborative Maneuver Protocol (CMP) for medium-term planning before the actual merge maneuver in challenging on-ramp scenarios. The authors defined the protocol in three steps, the planning, cooperation, and execution levels.

At the planning level, the function uses the CMP protocol for medium-term planning well before the actual merge maneuver. This involves exchanging information about vehicle positions, speeds, and intentions to determine a safe and efficient merging strategy. The cooperation level uses role-based cooperation to ensure that each vehicle knows its own responsibilities and those of other vehicles involved in the merging process. This helps to avoid conflicts and ensure a smooth merging process. The execution level function uses trajectory planning and control to execute the merging maneuver safely and efficiently. This involves adjusting vehicle speeds and positions based on real-time feedback from sensors and communication with other vehicles.

Figure 3 – Ramp Merge

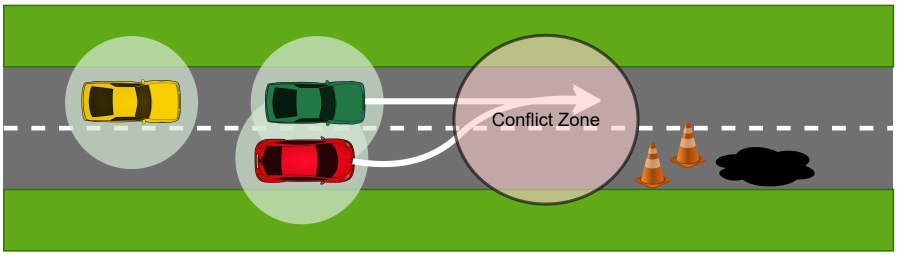


Source: Elaborated by the author.

2.2.3 Lane Change and Lane Merge

Lane-changing is a common maneuver drivers use to transition from one lane to another on a roadway. These maneuvers are often performed dynamically and are influenced by the current traffic conditions. Several motivations can cause a lane change, such as the need to follow a route, overtaking, or unexpected traffic problems. Figure 4 illustrates the dynamic lane changing and the conflict region that is dynamically generated when there is a need to perform the maneuver.

Figure 4 – Lane-changing



Source: Elaborated by the author.

Lane-changing maneuvers play a significant role in traffic dynamics and can cause speed oscillations, which influence the traffic flow, as noted by Moridpour, Sarvi and Rose (2010). Besides isolated lane change maneuvers, we can also consider lane-merging maneuvers, a common feature in platooning scenarios, as presented by Ploeg *et al.* (2018). This paper discusses the development of a layered control architecture for cooperative automated maneuvering in driving. The goal is to extend cooperative adaptive cruise control and platooning to accommodate common highway maneuvers, such as merging, and enable urban applications. The authors present two scenarios, a highway lane-reduction scenario and an urban intersection-crossing scenario, to demonstrate the effectiveness of their approach.

Poli *et al.* (2021) proposed a negotiation based on Cooperative Lane Merging called Cellular-Vehicle-to-Everything, which is sidelink connectivity for Cooperative Lane Merging services in a specific cross-border highway context. The negotiation protocol involves three vehicles: the initiator, the second vehicle in charge of creating the gap, and the merging vehicle. The negotiation procedure includes four messages: request message, recommendation message, safe-to-merge message, and denial message.

Our analysis reveals a notable lack of unified approaches to address multiple scenarios with the same method.

2.3 Negotiation, Protocols and Optimization

In this section, we present a review of research that applies negotiation, protocols, and optimization methods to address the problems previously discussed scenarios. The approaches will be categorized into peer-to-peer, centralized, and decentralized.

As previously mentioned, the resolution of conflicts between vehicles extends to several areas, especially in the field of aviation. The resolution of conflicts between aircraft has been well-established, and solutions such as the TCAS has become a standard in the aviation industry.

Šišlák, Volf and Pěchouček (2011) has developed a method called Iterative Peer-to-Peer Collision Avoidance (IPPCA) to resolve conflicts between aircraft. The method is based on pairwise negotiation and no decision is taken centrally. In conflict situations with more than three aircraft involved, they can negotiate in a pairwise manner and if new conflicts arise from their decisions, new negotiation rounds are initiated. During the negotiation process alternative trajectories are proposed and evaluated. As the negotiation evolves, new higher cost proposals must be evaluated for conflict resolution.

The previous method is similar to the bargaining proposal studied by Nash (1950), a bargaining game is played by two rational agents that have similar bargaining skills, both are aware of each other's preferences and can compare their desires. This was also used by Pritchett and Genton (2017) who used Game Theory bargaining modeling to carry out negotiations between aircraft without the need for a centralized air traffic controller. A bargaining process between aircraft was used to resolve the conflict between them in a decentralized manner and in a way that agents do not need to exchange private information. In this scenario, two aircraft with communication capability detect that they are on a collision course and initiate a negotiation to resolve the conflict. The aircraft could propose possible strategies in three ways to generate the trajectories: right/left, up/down and faster/slower. The aircraft start the process at zero cost and each one proposes maneuvers to change their trajectories. These trajectories are intercommunicated and each aircraft calculates a response route to resolve the conflict at the lowest cost.

The negotiation method converged in all experiments and, on average, resolved the conflict at a cost 29% lower than the lowest cost resolution by any aircraft alone. It was verified in the experiments that the resolution of the conflict in 10.1% generated a new conflict and in 1.5% generated more than one conflict. The authors mention that, among other factors, the density of vehicles is one of the factors for the generation of new conflicts.

Santos and Wolf (2019b) conducted experiments with CAVs using a similar bargaining approach. Before the negotiation process, an agent decides to make a lane change and initially communicates its maneuver to its neighboring vehicles. The neighbors check if the maneuver may cause a collision and send a response to the first agent. If the neighbors detect that the maneuver may cause a collision, the agent initiates a negotiation with the nearest agent that has a higher likelihood of collision.

After the initial conflict detection and the decision of the agents to start the negotiation, the bargaining process begins. Initially, the two Agents 1 and 2 assign zero value to the cost and calculate proposals that represent changes in speed, and these changes increase the cost. The vehicles then communicate their proposals, in which Agent 1 and 2 communicate the proposals to each other. If the proposals solve the conflict then an agreement has been found and the agents undertake to execute the proposal. If the proposals do not solve the conflict then the cost is increased according to the cost function of each agent. Also if the speed exceeds the highway speed limit the cost is attenuated to avoid that the new proposals exceed the limit of speed in an excessive way.

Through experiments, it was observed that the density of vehicles and the number of types of strategies are also limiting since only speed changes were used. To address the issue of traffic density, the proposed method prevents other vehicles from initiating negotiation maneuvers within a 100 meters range when two vehicles begin negotiating to prevent simultaneous maneuvers that can result in infeasible solutions.

One of the challenges presented at the GCDC 2016 was the lane merging scenario, where two platoons (A and B), each traveling in separate lanes, were required to merge into a single lane due to the roadworks. Each platoon had its own leader. The standard interaction protocol used in GCDC 2016 is described and discussed by Bengtsson *et al.* (2015), Englund *et al.* (2016) and Ploeg *et al.* (2018) and it consists of four phases. This method is a combination of a one-to-many and one-to-one approach.

Leaders are responsible for adjusting the speed of vehicles within the platoon in a process called Pace Making. This ensures that there are no major differences in speeds between vehicles. Once they are aligned the leaders sends a message to request the merge. Then they start do make a series of pair-up called Simultaneous Pair-up and Sequential Pair-up, the first one all vehicles of the platoon B pairs up with the closest vehicle in front of it in the platoon A. The vehicles from platoon B start to make gaps simultaneously. Next, the vehicles from platoon A pairs up with the closest vehicle in front of it in the platoon B and the vehicles in platoon A start to make gaps in sequential manner. After this alignment the CAVs start to send a Safe-To-Merge message and once the gaps are ready, which means that they have a safe distance to complete the maneuver, each vehicle in platoon B sends a STOM message to its pair in platoon A and starts a merging maneuver in a sequential manner.

Gaciarz, Aknine and Bhouri (2015) proposed a negotiation method to control intersections in real time using a solution based on the Constraint Satisfaction Problem. They represented the intersection in a discretized way and divided it into cells. Two policies were proposed to manage the negotiation: Iterated Policy, in which agents enter negotiation in waves and decisions cannot be revised, and Continuous Policy, agents can continuously enter negotiations and receive information from other vehicles. If the strategy fails an FCFS policy can be applied.

Meng *et al.* (2018) proposed a comparison of an ad hoc negotiation based method which is mainly based on FCFS and planning based. The results show that for small amounts of vehicles, the two methods have similar results, but in traffic situations with greater demand, the planning based method presents better results. We need to take into account that the negotiation approach implemented by the authors is a greedy approach to find the passing order of the vehicles.

One of the few works that present dynamic scenarios is presented by Liu *et al.* (2018) proposed a graph based approach to conflict resolution for the scenario of crossing intersections with flow of vehicles, but all vehicles are homogeneous, that is, they have the same physical characteristics and cost functions. Regarding multiple scenarios, Nichting *et al.* (2020) carried out a space time reservation work to resolve the conflict between vehicles for intersections and lane change, but up to 3 vehicles were tried.

Philippe *et al.* (2019) proposed an approach based on Probability Collectives (PC) for crossing intersections and roundabouts scenarios with 4 vehicles and later Philippe (2020) used the same proposal for crossing intersections with vehicle flow. However more dynamic scenarios must be evaluated and with vehicles with different characteristics and dimensions. It is also necessary to evaluate multi-objective systems where vehicles have different private cost functions. Finally, it is also important to investigate scenarios that do not have specific intersections, such as lane changes.

2.4 Decentralized negotiation

The study by Sislak *et al.* (2011) presents a method for resolving conflicts between aircraft that aims to minimize a common global objective, specifically, the safe distance between aircraft. This is achieved by finding control inputs that optimize the trajectory of the aircraft. The objective function penalizes changes to the original control inputs calculated by the aircraft that are given as an optimal trajectory. The objective function also has constraints that impose aircraft limitations and collisions are highly penalized. The proposed method is limited to changes in heading as the only form of action, but the authors mention that other types of actions can be used.

In each round of the algorithm, the costs of all the aircraft are added to compose a global cost, making the process multi-objective.The approach is compared with the IPPCA, a one-to-one solution also proposed by the same author, and it was found that the decentralized PC method results in trajectories that are 10 times smaller than the IPPCA, leading to an estimated 46% reduction in fuel consumption. However, it comes with the added cost of increased communication flow, which is 6 times more than the semi-centralized approach.

The study by Philippe *et al.* (2019) applied the decentralized PC method to coordinate vehicles at an crossroad and a roundabout. The intersections was modeled using predefined paths, and the only degree of freedom available to vehicles was the ability to change speed. Similarly, in the work of Santos and Wolf (2019a), the decentralized PC method was used for lane changing, with speed again being the only degree of freedom available. The results of these studies showed close approximation to those of a centralized method, indicating the algorithm's effectiveness, since the centralized method has all the information for optimization gathered in just one node.

2.5 Final Remarks

Unlike conflict resolution between aircraft, resolving conflicts between ground vehicles is more challenging. The main difference from the aviation area is the higher density of vehicles and limited maneuverability, often limited to only speed changes.

Much of the research in the field has centered on resolving conflicts at intersections using centralized methods. While one-to-one methods can be effectively applied to aviation problems that offer more degrees of freedom for conflict resolution, they are more challenging for ground vehicles, which are limited by their maneuverability and space restrictions. Furthermore, decentralized one-to-one methods tend to create new conflicts, mainly when multiple conflicts are being addressed simultaneously, which poses challenges to the scalability of the approach.

One-to-many methods are widely used mainly in platooning scenarios, but they also fall into the problem of scalability and difficult use in intersection scenarios. Another problem with one-to-many solutions is that management is performed by only one node or agent, which generates problems if the main node fails or loses connection.

Many-to-many solutions are a promising approach for conflict resolution in the context of CAVs as they can be treated as multi-agent systems where each vehicle has its own private cost functions and objectives. However, there are limited studies that tackle the problem from a multi-objective perspective, while considering dynamic scenarios with traffic flow and a generalized approach that can be applied to various scenarios. The use of infrastructure in many-to-many methods is also a consideration, as it can aid in managing data traffic and serve as a signal repeater in intersections with occlusion.

This thesis focuses on developing a generalized method for conflict resolution in dynamic traffic scenarios at crossing intersections with one and two lanes, on-ramp merging, and lane changing. We also evaluate the performance of our proposal with vehicles varying characteristics and dimensions. Finally, we consider agents with different private cost functions turning the problem into a multi-objective system.

CHAPTER 3

PROBABILITY COLLECTIVES

In this chapter, we introduce the PC method developed by Kulkarni, Tai and Abraham (2015). We choose this approach due to its formulation that allows each agent to determine its set of individual strategies from a range of possible strategies (called samples). Additionally, Kulkarni, Tai and Abraham (2015) have shown that their PC method is able to converge with a relatively small number of samples and iterations, which is a desirable feature for our problem as it allows the vehicles to solve the optimization quickly, which is a significant improvement over the original proposal by Wolpert, Strauss and Rajnarayan (2006).

While fast optimization is generally desirable in any context where optimization problems are being addressed, we should note that computational performance is beyond the scope of this thesis. Instead, our focus is on exploring the ability of the PC method to effectively organize strategies and achieve convergence in the context of negotiation and conflict resolution. All of the following formulations described in this chapter are based on the work of Kulkarni and Tai (2010), Kulkarni and Tai (2011), Kulkarni and Tai (2013) and Kulkarni, Tai and Abraham (2015).

Let's first consider the case where the PC method is used to solve general optimization problems of functions with N number of variables. The PC method treats each variable as an agent or a player defined as i. Each agent i randomly sets a strategy X from a predefined sampling interval space referred to $\Psi_i \in [\Psi_i^{lower}, \Psi_i^{upper}]$. As a general case, the interval can also be referred to as the sampling space and can be any real value from an interval.

Therefore, each agent i randomly samples strategy $X_i^{[r]}, \;\; r = 1, 2, ..., m$, wherein m is the maximum number of strategies within the sampling interval Ψ_i that forms a strategy set X_i as

$$X_i = \{X_i^{[1]}, X_i^{[2]}, ..., X_i^{[m]}\}, \;\; i \in \{1, 2, ..., N\} \tag{3.1}$$

In order to begin the optimization process, each agent i selects its own set of strategies X_i from the sampling interval Ψ_i. After selecting their individual strategies, the agents then form combined strategies Y_i by sampling randomly from the strategies of other agents in the system. Each agent i is able to make random guesses about the strategies that have been chosen by the other agents in the system. We define one combined strategy for every m strategies according to Equation 3.2.

$$\begin{array}{ll} Y_i^{[1]} & = \{X_1^{[?]}, X_2^{[?]}, \ldots, X_i^{[1]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \\ Y_i^{[2]} & = \{X_1^{[?]}, X_2^{[?]}, \ldots, X_i^{[2]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \\ & \vdots \\ Y_i^{[r]} & = \{X_1^{[?]}, X_2^{[?]}, \ldots, X_i^{[r]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \\ & \vdots \\ Y_i^{[m_i]} & = \{X_1^{[?]}, X_2^{[?]}, \ldots, X_i^{[m_i]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \end{array} \tag{3.2}$$

The superscript [?] indicates that it is a "random guess" and not known in advance. It is important to note that this process is carried out by all agents in the system.

After generating the combined strategy, each agent calculates the objective function (cost) $G(Y_i^{[r]})$, this step is done for all m strategies. The main objective of each agent i is to find a strategy that minimizes the sum of all costs $\sum_{r=1}^{m_i} G(Y_i^{[r]})$. The Equations 3.3, 3.4 and 3.5 present the calculation of the objective function costs followed by the sum of all these costs. The equations exemplify the calculation for agent 1, i and N.

$$\left.\begin{array}{ll} Y_1^{[1]} & = \{X_1^{[1]}, X_2^{[?]}, \ldots, X_i^{[?]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_1^{[1]}) \\ Y_1^{[2]} & = \{X_1^{[2]}, X_2^{[?]}, \ldots, X_i^{[?]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_1^{[2]}) \\ & \vdots \\ Y_1^{[r]} & = \{X_1^{[r]}, X_2^{[?]}, \ldots, X_i^{[?]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_1^{[r]}) \\ & \vdots \\ Y_1^{[m_i]} & = \{X_1^{[m_i]}, X_2^{[?]}, \ldots, X_i^{[?]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_1^{[m_i]}) \end{array}\right\} \Rightarrow \sum_{r=1}^{m_i} G(Y_1^{[r]}) \tag{3.3}$$

$$\left.\begin{array}{ll} Y_i^{[1]} & = \{X_1^{[1]}, X_2^{[?]}, \ldots, X_i^{[?]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_i^{[1]}) \\ Y_i^{[2]} & = \{X_1^{[2]}, X_2^{[?]}, \ldots, X_i^{[?]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_i^{[2]}) \\ & \vdots \\ Y_i^{[r]} & = \{X_1^{[r]}, X_2^{[?]}, \ldots, X_i^{[?]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_i^{[r]}) \\ & \vdots \\ Y_i^{[m_i]} & = \{X_1^{[m_i]}, X_2^{[?]}, \ldots, X_i^{[?]}, \ldots, X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_N^{[m_i]}) \end{array}\right\} \Rightarrow \sum_{r=1}^{m_i} G(Y_i^{[r]}) \tag{3.4}$$

$$\left.\begin{array}{ll} Y_N^{[1]} & = \{X_1^{[1]}, X_2^{[?]}, ..., X_i^{[?]}, ..., X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_N^{[1]}) \\ Y_N^{[2]} & = \{X_1^{[2]}, X_2^{[?]}, ..., X_i^{[?]}, ..., X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_N^{[2]}) \\ & \vdots \\ Y_N^{[r]} & = \{X_1^{[r]}, X_2^{[?]}, ..., X_i^{[?]}, ..., X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_N^{[r]}) \\ & \vdots \\ Y_N^{[m_i]} & = \{X_1^{[m_i]}, X_2^{[?]}, ..., X_i^{[?]}, ..., X_{N-1}^{[?]}, X_N^{[?]}\} \Rightarrow G(Y_N^{[m_i]}) \end{array}\right\} \Rightarrow \sum_{r=1}^{m_i} G(Y_N^{[r]}) \tag{3.5}$$

Minimizing $\sum_{r=1}^{m_i} G(Y_i^{[r]})$, which determines the optimal strategy, can be very hard to solve and computationally expensive task (KULKARNI; TAI; ABRAHAM, 2015). To address this problem, Wolpert, Strauss and Rajnarayan (2006) and Kulkarni, Tai and Abraham (2015) proposed the use of Homotopy function, a method that transforms the function into another topological space that is easier to minimize. The Homotopy function is defined by Equation 3.6 and is parameterized by T called temperature.

$$J_i(q(X_i),T) = \sum_{r=1}^{m_i} G(Y_i^{[r]}) - TE, \;\; T \in [0,\infty) \tag{3.6}$$

We denoted the "easier" function as E in the Equation 3.6. Wolpert, Strauss and Rajnarayan (2006) and Kulkarni, Tai and Abraham (2015) suggested to use the entropy function presented in the Equation 3.7.

$$S_i(q) = -\sum_{r=1}^{m_i} q(X_i^{[r]}) \cdot ln(q(X_i^{[r]})) \tag{3.7}$$

One of the main advantages of the PC method is its ability to transform the domain value $G(Y_i^{[r]})$ into the probability domain $E(G(Y_i^{[r]}))$ (ELLIOTT; TALLANT; DOGAN, 2017). This transformation is achieved through the use of the expectation operation (denoted by Equation 3.8), which changes the goal from finding the optimal function value to searching for the optimal probability distribution. The subscript (i) means every agent other than i.

$$E(G(Y_i^{[r]})) = G(Y_i^{[r]})q(X_i^{[r]})\prod_{(i)} q(X_{(i)}^{[?]}) \tag{3.8}$$

At the beginning of the optimization the probability of each strategy $q(X_i^{[?]})$ is initialized equally distributed with $1/m_i$. Equation 3.9 presents the initial probability distribution.

$$q(X_i^{[r]}) = 1/m_i, \;\; r = 1,2,...,m_i \tag{3.9}$$

Every agent calculates the expected collection of system objectives $\sum_{r=1}^{m_i} E(G(Y_i^{[r]}))$ which is the sum of all expected system objective $E(G(Y_i^{[r]}))$. This process is carried out for all

N agents following the Equations 3.10, 3.11 and 3.12. The equations exemplify the calculation for agent 1, i and N.

$$\left.\begin{array}{l} G(Y_1^{[1]})q(X_1^{[1]})\prod_{(1)} q(X_{(1)}^{[?]}) = E(G(Y_1^{[1]})) \\ \vdots \\ G(Y_1^{[r]})q(X_1^{[r]})\prod_{(1)} q(X_{(1)}^{[?]}) = E(G(Y_1^{[r]})) \\ \vdots \\ G(Y_1^{[m_i]})q(X_1^{[m_i]})\prod_{(1)} q(X_{(1)}^{[?]}) = E(G(Y_1^{[m_i]})) \end{array}\right\} \Rightarrow \sum_{r=1}^{m_i} E(G(Y_1^{[r]})) \tag{3.10}$$

$$\left.\begin{array}{l} G(Y_i^{[1]})q(X_i^{[1]})\prod_{(i)} q(X_{(i)}^{[?]}) = E(G(Y_i^{[1]})) \\ \vdots \\ G(Y_i^{[r]})q(X_i^{[r]})\prod_{(i)} q(X_{(i)}^{[?]}) = E(G(Y_i^{[r]})) \\ \vdots \\ G(Y_i^{[m_i]})q(X_i^{[m_i]})\prod_{(i)} q(X_{(i)}^{[?]}) = E(G(Y_i^{[m_i]})) \end{array}\right\} \Rightarrow \sum_{r=1}^{m_i} E(G(Y_i^{[r]})) \tag{3.11}$$

$$\left.\begin{array}{l} G(Y_N^{[1]})q(X_N^{[1]})\prod_{(N)} q(X_{(N)}^{[?]}) = E(G(Y_N^{[1]})) \\ \vdots \\ G(Y_N^{[r]})q(X_N^{[r]})\prod_{(N)} q(X_{(N)}^{[?]}) = E(G(Y_N^{[r]})) \\ \vdots \\ G(Y_N^{[m_i]})q(X_N^{[m_i]})\prod_{(N)} q(X_{(N)}^{[?]}) = E(G(Y_N^{[m_i]})) \end{array}\right\} \Rightarrow \sum_{r=1}^{m_i} E(G(Y_N^{[r]})) \tag{3.12}$$

As the expected collection of system objectives still hard to minimize we use it to compose the Homotopy function and, finally, the Homotopy function to be minimized by each agent i is presented in the Equation 3.13.

$$\begin{aligned} J_i(q(X_i),T) &= \sum_{r=1}^{m_i} E(G(Y_i^{[r]})) - TS_i \\ &= \sum_{r=1}^{m_i} \left(G(Y_i^{[r]}) q(X_i^{[r]}) \prod_{(i)} q(X_{(i)}^{[?]}) \right) - T \left(- \sum_{r=1}^{m_i} q(X_i^{[r]}) \cdot ln(q(X_i^{[r]})) \right) \end{aligned} \tag{3.13}$$

The Homotopy function can be solved using an appropriate technique such as the Nearest Newton Descent Scheme as suggested by Wolpert, Strauss and Rajnarayan (2006) and Kulkarni, Tai and Abraham (2015). Thus, we can minimize the Homotopy function and solve the problem using the iterative method described in Equations 3.14, 3.15 and 3.16, in which the parameter k is initialized at 1 and updated up to the maximum number of iterations k_{final}. The step size is represented by the parameter α_{step}, which is initialized equally for all agents and kept fixed throughout the optimization.

$$q(X_i^{[r]})^{k+1} \leftarrow q(X_i^{[r]})^k - \alpha_{step} \cdot q(X_i^{[r]})^k \cdot k_r update \tag{3.14}$$

where

$$k_r update = \frac{(\text{Contribution of } X_i^{[r]})^k}{T} + S_i^k + log_2(q(X_i^{[r]})^k) \tag{3.15}$$

and

$$(\text{Contribution of } X_i^{[r]})^k = E(G(Y_i^{[r]}))^k - \left(\sum_{r=1}^{m_i} E(G(Y_i^{[r]}))^k \right) \tag{3.16}$$

By applying this iterative method, we can find the optimal solution for that agents considering the chosen strategies. This will result in a strategy called favorable strategy $X_i^{[fav]}$, which will be the best strategy so far. After this convergence process, Kulkarni, Tai and Abraham (2015) propose to continue the optimization by decreasing the search space Ψ_i around the favorable strategy $X_i^{[fav]}$ so that each agent composes a new set of X_i strategies and restart the optimization.

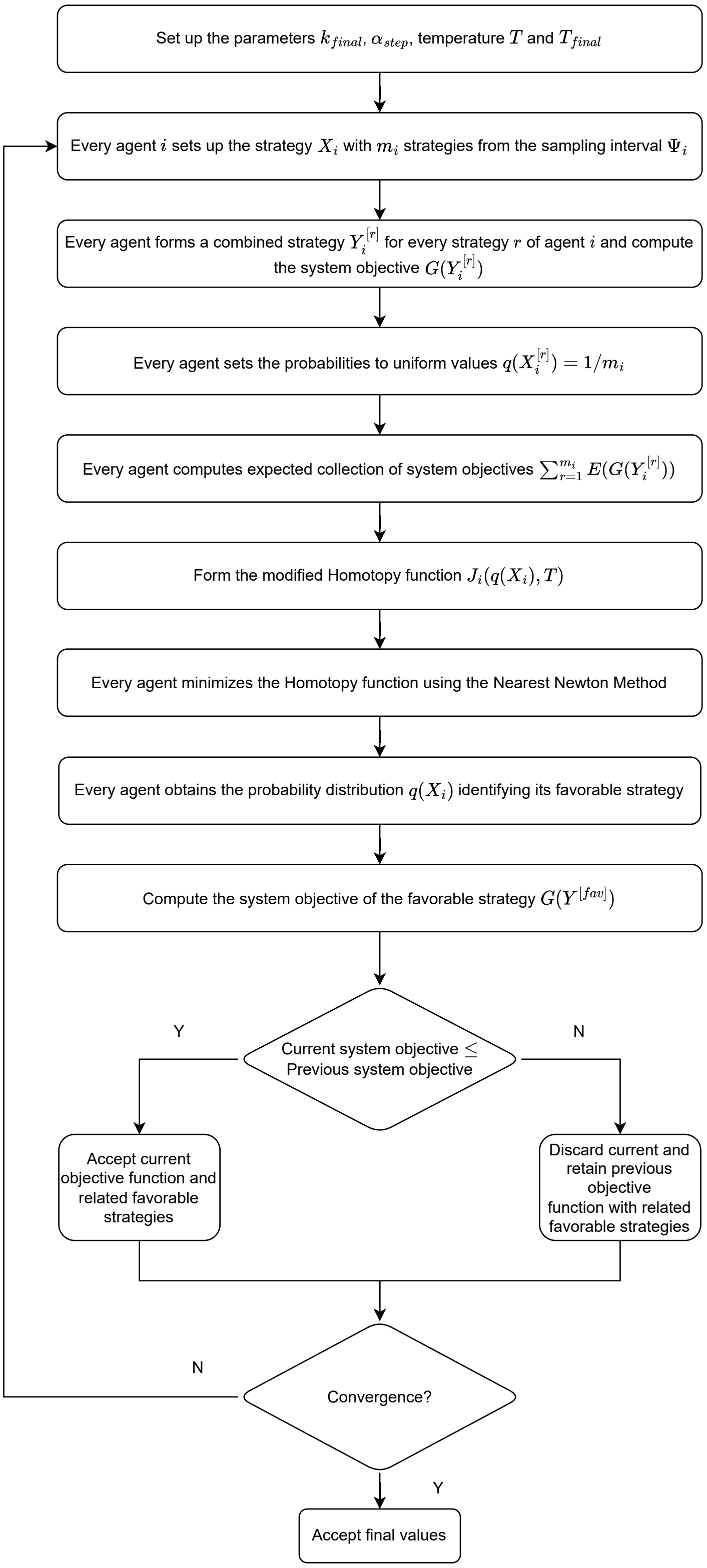
We then compute the system objective favorable $G(Y^{[fav]})$ where $Y^{[fav]}$ is given by the Equation 3.17.

$$Y^{[fav]} = \left\{ X_1^{[fav]}, X_2^{[fav]}, \ldots, X_{N-1}^{[fav]}, X_N^{[fav]} \right\} \tag{3.17}$$

Upon completing this process, we have completed the first iteration of the optimization. The optimization process ends when the stopping conditions is met, which is determined by the following condition: $T = T_{final}$ or $T \to 0$ and if $G(X_i^{[fav]})^n - G(X_i^{[fav]})^{n-1} \leq \varepsilon$ occur several times in a row.

Figure 5 summarizes the optimization algorithm, highlighting the steps involved in the process. It is worth noting that the minimization step includes an iterative loop to apply the Nearest Newton Descent Scheme to calculate the favorable probabilities.

Figure 5 – Probability Collectives Algorithm


Set up the parameters k_{final}, α_{step}, temperature T and T_{final}
Every agent i sets up the strategy X_i with m_i strategies from the sampling interval Ψ_i
Every agent forms a combined strategy $Y_i^{[r]}$ for every strategy r of agent i and compute the system objective $G(Y_i^{[r]})$
Every agent sets the probabilities to uniform values $q(X_i^{[r]}) = 1/m_i$
Every agent computes expected collection of system objectives $\sum_{r=1}^{m_i} E(G(Y_i^{[r]}))$
Form the modified Homotopy function $J_i(q(X_i), T)$
Every agent minimizes the Homotopy function using the Nearest Newton Method
Every agent obtains the probability distribution $q(X_i)$ identifying its favorable strategy
Compute the system objective of the favorable strategy $G(Y^{[fav]})$
Y
Current system objective $\leq$ Previous system objective
N
Accept current objective function and related favorable strategies
Discard current and retain previous objective function with related favorable strategies
N
Convergence?
Y
Accept final values

Source: Adapted from Kulkarni and Tai (2010), Kulkarni, Tai and Abraham (2015).

3.1 Decentralized and Multi-objective Probability Collectives

Based upon the studies Sislak *et al.* (2011) and Waldock and Corne (2010), our proposal incorporated two additional layers into the algorithm previously proposed by Kulkarni, Tai and Abraham (2015). The algorithm is illustrated in Figure 6.

Each agent i randomly samples strategy $X_i^{[r]},\ \ r = 1,2,...,m$ within the sampling interval Ψ_i that forms a strategy set X_i. They also randomize the strategy set that will be used to form a combined strategy, this randomization is presented in Equation 3.18, where the superscript [?] means a random strategy.

$$X_i = \{X_i^{[?]}, X_i^{[?]}, ..., X_i^{[?]}\},\ \ i \in \{1,2,...,N\} \tag{3.18}$$

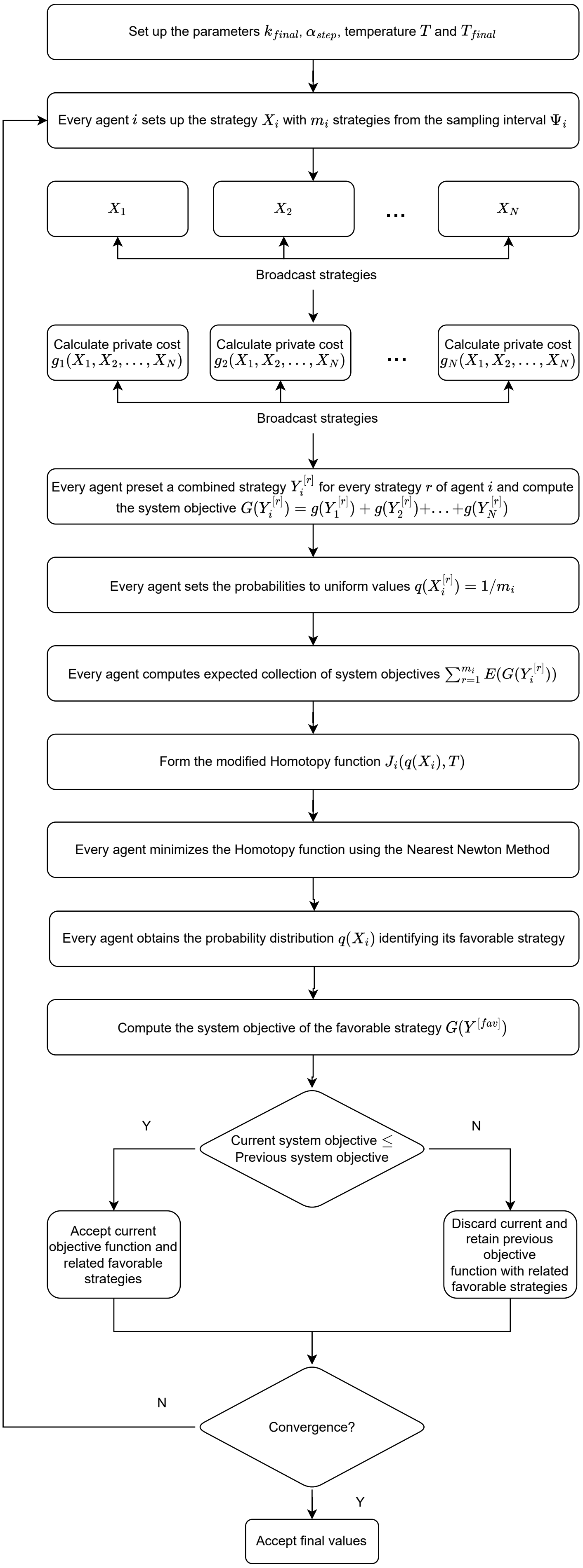
Unlike the algorithm shown in Figure 5, the combined strategy and system objective are not calculated directly. There are two intermediate layers regarding the communication between agents. The agents broadcast their sets of strategy to each other and then they can form the combined strategies Y_i. At this point, each combined strategy has the same value for all agents, for example, $Y_1^{[r]} = Y_2^{[r]} = Y_N^{[r]}$ Each agent calculates the cost using their private functions, represented in the Figure 6 by the g function.

Next, the agents broadcast again, now sharing the cost value g. They wait to receive all the information and sum all costs to calculate the system objective. Equation 3.19 presents the final calculation of the cost which is the sum of the costs of all agents. This scalarization process for solving multi-objective problems is known as the weighting method and was first proposed by Zadeh (1963).

$$\left.\begin{array}{ll} Y_1^{[r]} & = \{X_1^{[r]}, X_2^{[r]}, ..., X_i^{[r]}, ..., X_{N-1}^{[r]}, X_N^{[r]}\} \Rightarrow g_1(Y_1^{[r]}) \\ Y_2^{[r]} & = \{X_1^{[r]}, X_2^{[r]}, ..., X_i^{[r]}, ..., X_{N-1}^{[r]}, X_N^{[r]}\} \Rightarrow g_2(Y_2^{[r]}) \\ & \vdots \\ Y_i^{[r]} & = \{X_1^{[r]}, X_2^{[r]}, ..., X_i^{[r]}, ..., X_{N-1}^{[r]}, X_N^{[r]}\} \Rightarrow g_i(Y_i^{[r]}) \\ & \vdots \\ Y_N^{[r]} & = \{X_1^{[r]}, X_2^{[r]}, ..., X_i^{[r]}, ..., X_{N-1}^{[r]}, X_N^{[r]}\} \Rightarrow g_N(Y_N^{[r]}) \end{array}\right\} \Rightarrow G(Y^{[r]}) = \sum_{i=1}^{N} g_i(Y_i^{[r]}) \tag{3.19}$$

Utilizing the computed total cost $G(Y^{[r]})$, the optimization algorithm then proceeds as described in the method proposed by Kulkarni, Tai and Abraham (2015).

Figure 6 – Probability Collectives Algorithm


Set up the parameters k_{final}, α_{step}, temperature T and T_{final}
Every agent i sets up the strategy X_i with m_i strategies from the sampling interval Ψ_i
X_1
X_2
...
X_N
Broadcast strategies
Calculate private cost $g_1(X_1, X_2, \ldots, X_N)$
Calculate private cost $g_2(X_1, X_2, \ldots, X_N)$
...
Calculate private cost $g_N(X_1, X_2, \ldots, X_N)$
Broadcast strategies
Every agent preset a combined strategy $Y_i^{[r]}$ for every strategy r of agent i and compute the system objective $G(Y_i^{[r]}) = g(Y_1^{[r]}) + g(Y_2^{[r]}) + \ldots + g(Y_N^{[r]})$
Every agent sets the probabilities to uniform values $q(X_i^{[r]}) = 1/m_i$
Every agent computes expected collection of system objectives $\sum_{r=1}^{m_i} E(G(Y_i^{[r]}))$
Form the modified Homotopy function $J_i(q(X_i), T)$
Every agent minimizes the Homotopy function using the Nearest Newton Method
Every agent obtains the probability distribution $q(X_i)$ identifying its favorable strategy
Compute the system objective of the favorable strategy $G(Y^{[fav]})$
Y
N
Current system objective $\leq$ Previous system objective
Accept current objective function and related favorable strategies
Discard current and retain previous objective function with related favorable strategies
N
Convergence?
Y
Accept final values

Source: Elaborated by the author.

3.2 Problem Statement

Now we aim define the problems and scenarios that are the focus of this thesis. Using the Probability Collectives definitions outlined in Section 3.1, we formulate our approach for resolving conflicts of CAVs in the presented scenarios. Additionally, we examine how our solution can be adapted to accommodate dynamic scenarios.

Consider the three scenarios depicted in Figure 7. In these scenarios, we can observe different challenging situations where vehicles are in imminent conflict, requiring some interaction to resolve effectively. These situations represent critical moments on the road where quick decision-making and coordinated actions are essential to ensure safety and efficient traffic flow.

In the first scenario (Figure 7a), three vehicles approach an on-ramp simultaneously. We can see that the yellow vehicle is approaching an on-ramp junction, but there is concurrent traffic on the horizontal road. Without proper coordination, a collision becomes highly probable. The second scenario (Figure 7b) presents a non-signalized crossroad. This type of scenario typically involves a shared zone where vehicles from different directions may generate a conflict.

The third scenario (Figure 7c) depicts a challenging lane change situation. In this scenario, the red vehicle needs to change lanes either to overtake the yellow vehicle ahead or just to follow its intended route. However, the presence of other vehicles in adjacent lanes adds complexity and increases the risk of collision. As the red vehicle navigates the lane change maneuver, it must compete with the surrounding traffic for the same space. The proximity and movement of the neighboring vehicles create a competitive environment where merging into the desired lane becomes a critical task. The risk of side-swiping or rear-end collisions increases, demanding precise timing, accurate decision-making, and effective communication among drivers.

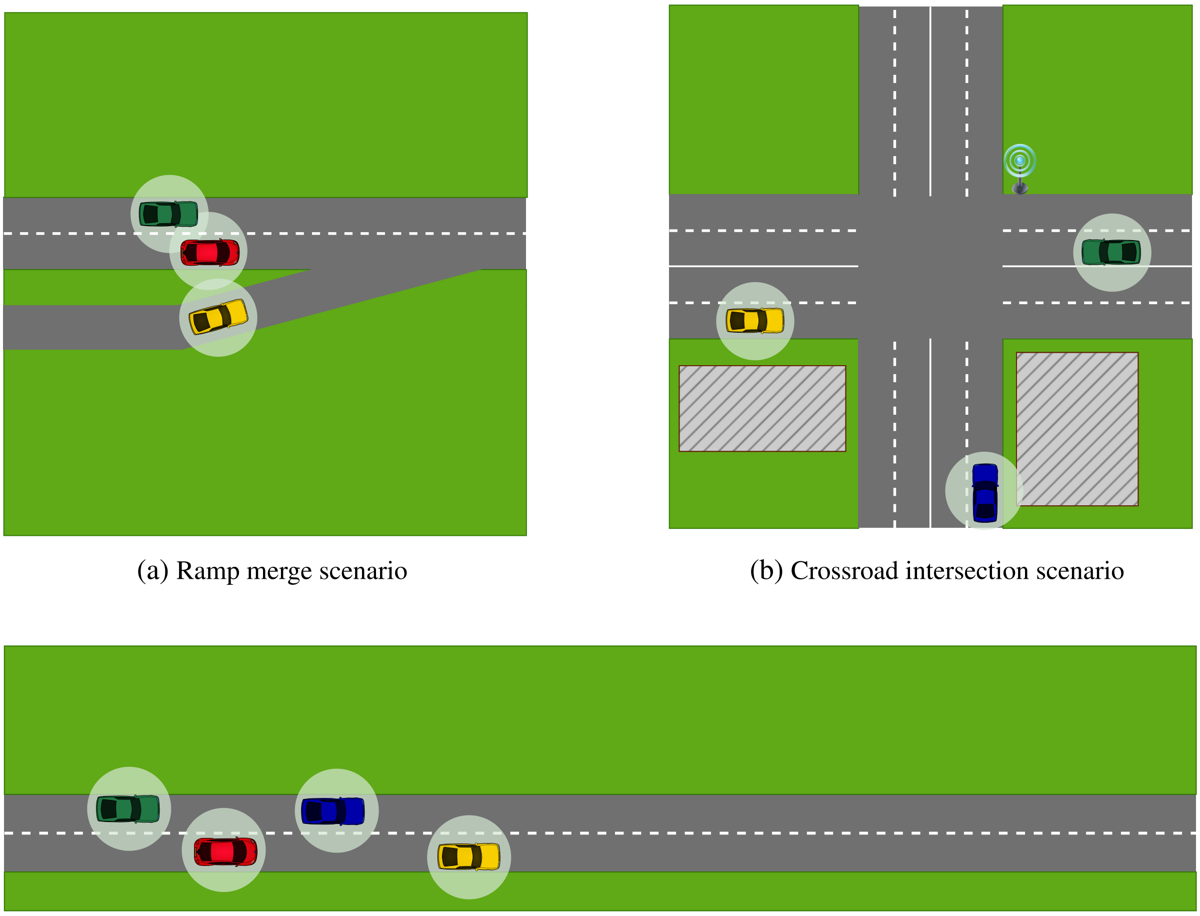
The three situations represent a state of imminent conflict that can generate an impasse for the decision-making system of the vehicles, and there is also the possibility of causing an accident if the maneuvers are not well coordinated. Resolving these conflicts demands effective negotiation and cooperation among drivers to ensure smooth traffic flow and prevent potential accidents.

In the intersection crossroad and on-ramp merge scenarios, fixed negotiation zones were defined. When CAVs enter these zones they are required to communicate and send their information to other vehicles. We assume that vehicles can be in conflict and they can send their strategies (trajectories) to other vehicles and also to the infrastructure that is in charge of relaying the communication.

In the lane change scenarios, before the CAVs initiate any maneuver, they request the maneuver by broadcasting their strategies and also asking the strategies of the other CAVs, in order to calculate and assess the probability of collision.

In our static scenarios, where the vehicles are previously positioned before starting the

Figure 7 – Representation of scenarios with imminent conflict



(a) Ramp merge scenario

(b) Crossroad intersection scenario

(c) Lane change scenario

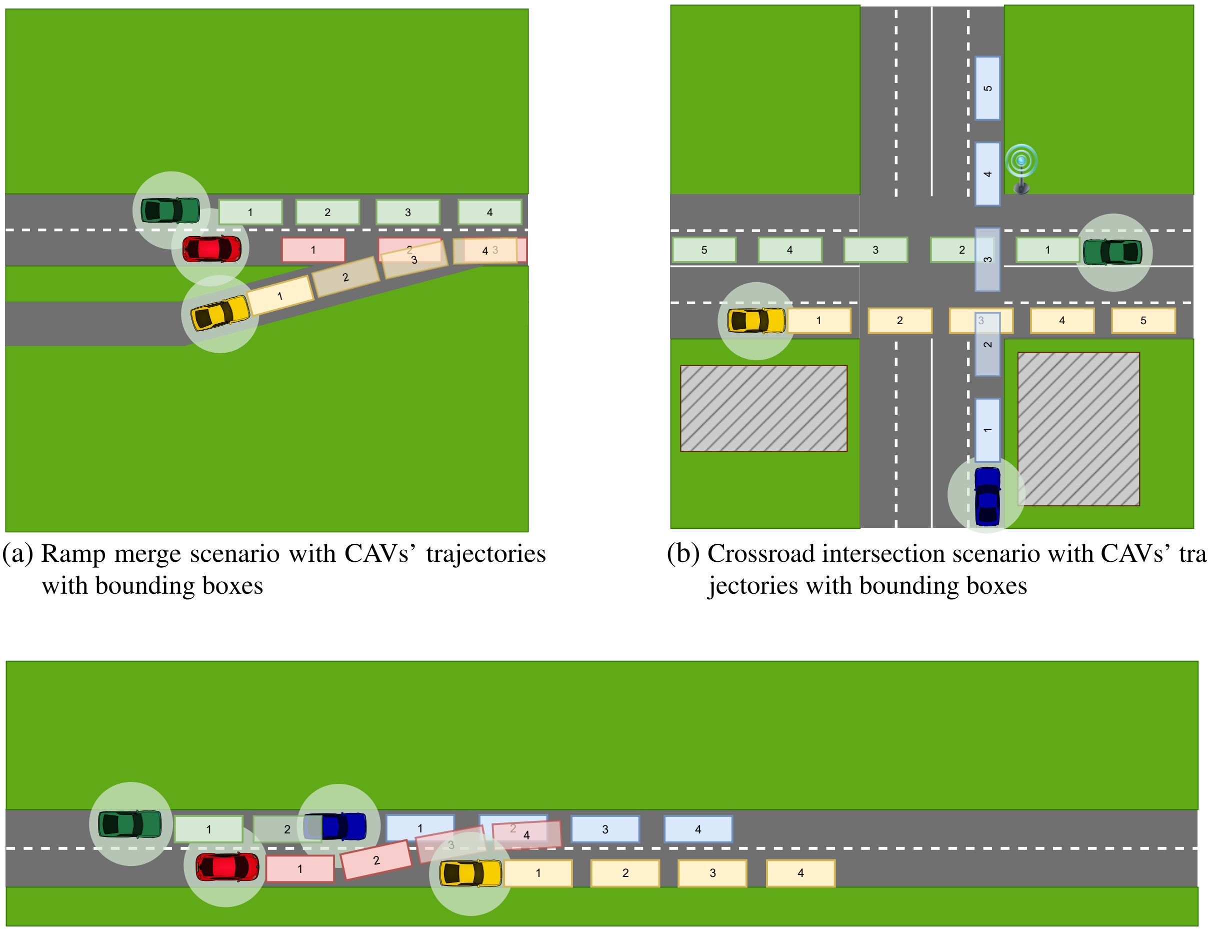
Source: Elaborated by the author.

simulation, we always positioned the vehicles to force the conflict and the vehicles must negotiate to avoid it. The objective of the CAVs are negotiate and generate strategies that are translated into trajectories to solve the conflict. In the dynamic scenarios the vehicles are generated from a Poisson distribution and thus a traffic flow is created (MENG *et al.*, 2018) (BUCKMAN *et al.*, 2019).

Figure 8 illustrates the strategies expressed as the trajectories of the CAVs. The bounding boxes of each vehicle's color are used to represent the trajectories, the value inside the bounding box represents the time. When the CAVs receive the trajectories from other vehicles, they can determine whether their chosen strategies will result in collisions by analyzing the intersection of their positions over time. Thus, we can notice in these situations that there are no collisions in the trajectories.

We assume that the CAVs communicate and receive strategies from each other without network delays or communication failures. By using these trajectories, it is possible to calculate the private costs and collision constraints. The CAVs then communicate again by sending their

Figure 8 – CAVs' discretized trajectories with bounding boxes



(a) Ramp merge scenario with CAVs' trajectories with bounding boxes

(b) Crossroad intersection scenario with CAVs' trajectories with bounding boxes

(c) Lane change scenario with CAVs' trajectories with bounding boxes

Source: Elaborated by the author.

partial costs. After this process, all vehicles have knowledge of the costs and the optimization process begins, which will occur for a predefined number of times. If the conflict is not resolved, new iterations will be initiated by modifying the strategies.

3.3 Approach Definition

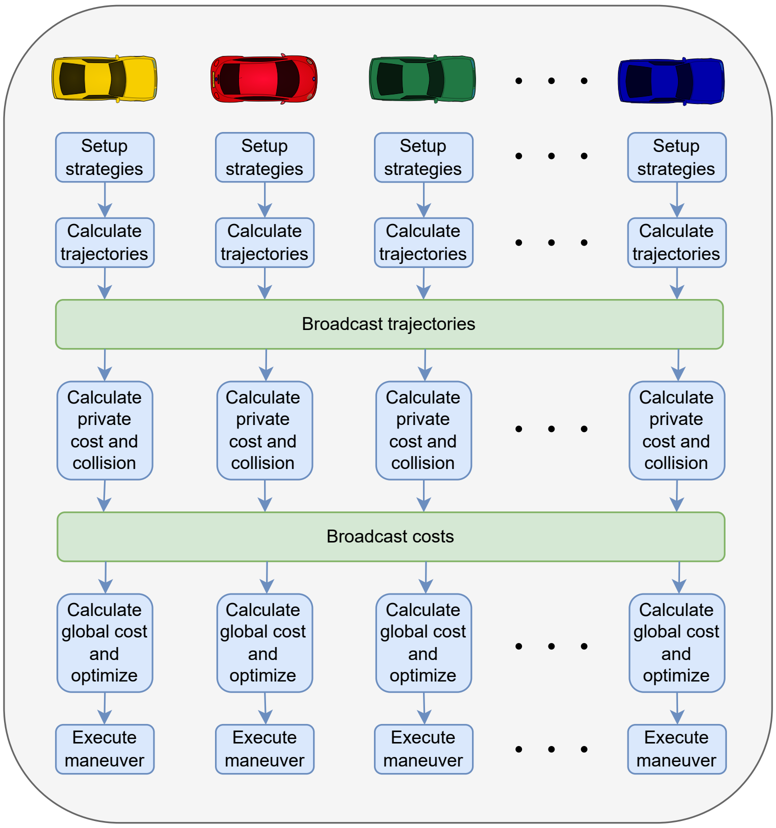
We will first present the Probability Collectives algorithm adapted for CAVs, as shown in Figure 9. The process begins with a vehicle initiating a negotiation request, either due to a maneuver such as a lane change or upon entering a fixed negotiation area, such as an intersection or lane junction.

CAVs initialize their strategies by choosing changes in speed. For each strategy, the trajectory is calculated for a given time interval and sampling time, and the trajectories represent "pre-defined" routes as pointed out by Chen and Englund (2016). As proposed by Dongxin *et al.* (2021) and Levin and Rey (2017), once a trajectory is chosen it is no longer changed and overtaking is not considered either. In this way, the positions are calculated in continuous space, discretized time and a decision making is not changed.

At this point the CAVs are prepared to broadcast the information, for better understanding we illustrate the process happening in a synchronized way but it doesn't necessarily have to be. The vehicles then send the trajectories to each of the strategies with the positioning information, orientation, length and width of the vehicle and also a unique id. From these data, the vehicles are able to calculate the bounding box oriented in each of the positions as shown in the illustrations in Figure 8.

With these data, it becomes possible to determine if a collision occurs and calculate the private cost function. The calculated cost values are then broadcasted. Upon receiving the data, all agents will have the complete information needed to perform the optimization and find their optimal strategy. Once the optimization is completed, vehicles start executing their strategies.

Figure 9 – Probability Collectives algorithm adapted for CAVs



Source: Elaborated by the author.

After a brief introduction of the method, we present the mathematical formulation and the steps of the algorithm. Consider an agent i from a set of N agents. Each agent i randomly sets a strategy X from a predefined sampling interval space referred to $\Psi \in [\Psi^{lower}, \Psi^{upper}]$. As a general case, the interval can also be referred to as the sampling space and can be any real value from an interval as we presented in Section 3. Here, we consider the sampling space to be a set of integers.

In our approach, the sampling space is represented by integer values that correspond to changes in the speed profile. This values represents a modification in the speed originally calculated by the planner of the CAVs (path planning and control). These speed values are randomly selected from the sampling speed interval $\Psi_{speed} \in \mathbb{Z}$. This allows adjustments in the CAV's speed profile and consequently in their trajectories.

For instance, if we set the sampling interval as $\Psi_{speed} = [0,1,2,3,4,5,6,7,8]$, each value represents a change in the CAV's speed. A value of 0 means no change, while a value of 1 corresponds to an increase of 1 unit in the current speed of the vehicle. Similarly, a value of 2 represents an increase of 2 units in the current speed, and so on. It is also possible to use the values to decrease the speed, suppose that the CAV's trajectory planner has set the speed at s for the next t seconds, and assuming that the speed limit cannot be exceeded, we can decrease the planned speed by 1, 2, 3, and so on by using these values from the sampling interval. Despite the possibility of using speed increases, in this work we only use speed decreases to ensure that the CAVs do not exceed the road's speed limit.

Unlike the sampling interval proposed by Kulkarni, Tai and Abraham (2015), we have set the same Ψ_{speed} for all agents and it remains unchanged throughout the optimization process, because the sample size is limited to speed units, which restricts the size of our set. We use a set of integers in a discretized manner to simplify the implementation and optimization process, but it is also possible to use a continuous set. Furthermore, it has to be taken into account to consider the number of decimal places used, because in real scenarios it may be difficult for a CAV's controllers to maintain speeds with high precision using many decimal places.

Therefore, each agent i randomly samples strategy $X_i^{[r]}, r = 1,2,...,m$, wherein m is the maximum number of strategies that forms a strategy set X_i as

$$X_i = \{X_i^{[1]}, X_i^{[2]}, ..., X_i^{[m]}\}, \ \ i \in \{1,2,...,N\} \tag{3.20}$$

Since the set Ψ_{speed} remains fixed throughout the optimization, we store in the set X_i the indices of the values of the set Ψ_{speed} that were randomly chosen instead of the value itself. We use the notation $[X_i^{[r]}]$ to refer to the index of the set Ψ_{speed} rather than the value.

$$X_i = \{[X_i^{[1]}], [X_i^{[2]}], ..., [X_i^{[m]}]\}, \ \ i \in \{1,2,...,N\} \tag{3.21}$$

Finally, we perform a randomization on the set X_i and we obtain the set X_{i_random}, as shown in Equation 3.22, where the superscript ? means a random sampling without replacement from the indexes set X_i.

$$X_{i\_random} = \{[X_i^{[?]}], [X_i^{[?]}], ..., [X_i^{[?]}\}], \ \ i \in \{1,2,...,N\} \tag{3.22}$$

The agents calculate the pose (position and orientation) estimation for each strategy. This estimation is represented as a list of positions that denote the vehicle's trajectory. The function

$\mathscr{P}(X_i^{[r]}, \Delta t, ts)$ calculates the position estimation for the next time interval Δt divided into ts time steps. This function returns the list of positions corresponding to the strategy $X_i^{[r]}$. The set of all position lists, each corresponding to a particular strategy, is shown in Equation 3.23.

$$P_i = \{\mathscr{P}(X_i^{[1]}, \Delta t, ts), \mathscr{P}(X_i^{[2]}, \Delta t, ts), ..., \mathscr{P}(X_i^{[m]}, \Delta t, ts)\}, \;\; i \in \{1, 2, ..., N\} \tag{3.23}$$

Each CAV broadcasts the positioning strategy information, in other words, each one sends its set P_i to the other agents. Once all CAVs have sent their strategies, they wait to receive the strategies of the other agents. After this sequence of communications, the agents form the combined strategy Y_i presented in Equation 3.2.

Consider the function $\mathscr{C}(\mathscr{P}(X_1^{[r]}, \Delta t, ts), \mathscr{P}(X_2^{[r]}, \Delta t, ts))$. It calculates whether there is a collision between the CAVs using the trajectory list. If there is a collision, the function returns a significant high value to prevent this strategy from being chosen and thus represent a constraint. On the other hand, if there is no collision then the function returns zero, described in Equation 3.24. Here we represent a significantly high value as being infinite, but in terms of implementation it should be avoided due to mathematical precision.

$$\mathscr{C}(\mathscr{P}(X_1^{[r]}, \Delta t, ts), \mathscr{P}(X_2^{[r]}, \Delta t, ts), ..., \mathscr{P}(X_N^{[r]}, \Delta t, ts)) = \begin{cases} \infty, & \text{if there is any collision.} \\ 0, & \text{otherwise.} \end{cases} \tag{3.24}$$

In order to evaluate the cost of each combined strategy, the agents consider both, the potential for collision and the private cost function which is the deviation from the planned speed multiplied by a weight. For each combined strategy $Y_i^{[r]}$, the agents calculate the cost $G(Y_i^{[r]})$ using Equation 3.25, which takes into account both the output of the collision function and the absolute value of the proposed strategy's deviation from the planned speed. This cost function is similar to the function proposed by Sislak *et al.* (2011) that penalizes the deviation from airplanes' optimal trajectories.

In this study, the weighting factor w plays a crucial role in determining the cost function. To distinguish between different vehicle classes, we employed a vehicle weight ratio as the value of w. This approach enabled us to use diverse cost functions in the simulations and turn the problem into multi-objective.

$$\begin{aligned} G(Y_i^{[r]}) = \mathscr{C}(\mathscr{P}(X_1^{[r]}, \Delta t, ts), \mathscr{P}(X_2^{[r]}, \Delta t, ts), ..., \mathscr{P}(X_N^{[r]}, \Delta t, ts)) \\ + w(PlannedSpeed + |\Psi_{speed}[X_i^{[r]}]|) \end{aligned} \tag{3.25}$$

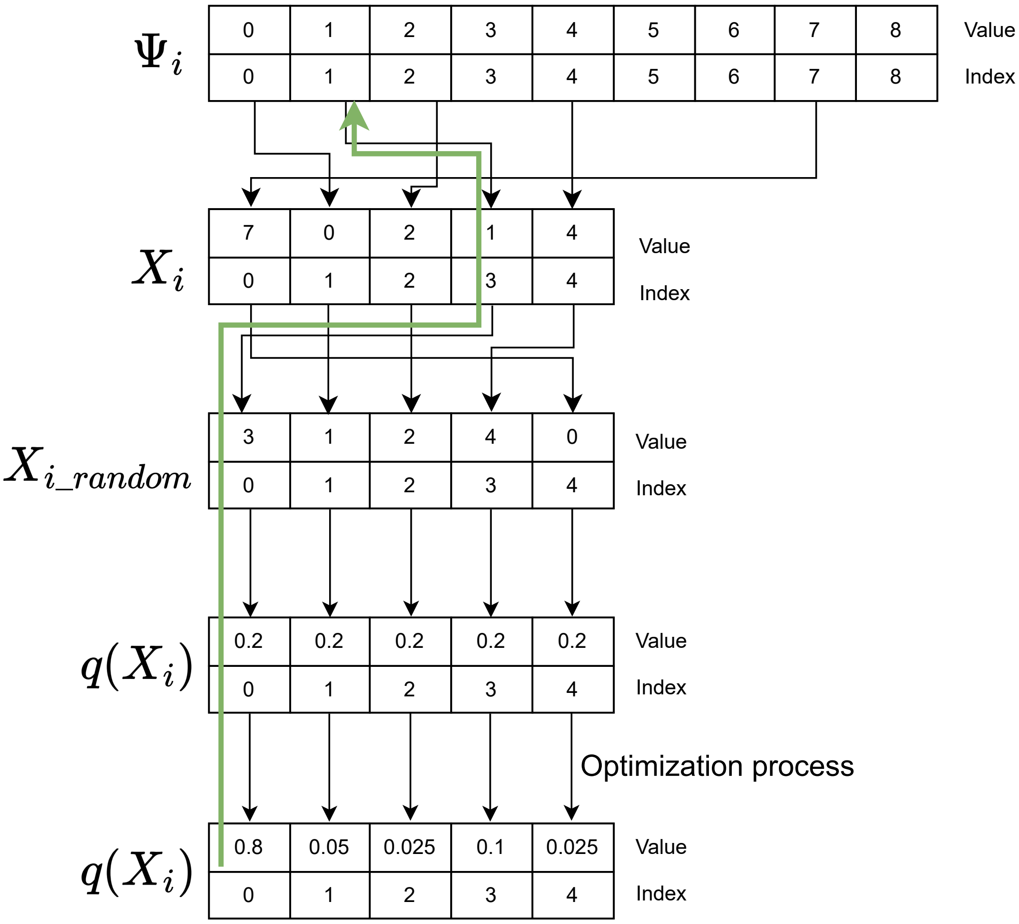
After calculating the costs of each strategy, the list of costs $[G(Y_i^{[1]}), G(Y_i^{[2]}), ..., G(Y_i^{[m]})]$ is broadcasted to the other agents. As before, they wait to receive the cost lists from the other agents. Once all cost lists have been received, we sum the costs of each agent's respective strategies to obtain the total costs to be minimized.

We calculate the expected system objective $E(G(Y_i^{[r]}))$ with random guesses to transform from the domain value into the probability domain.

Figure 10 illustrates the structure of the strategy set and how it is used throughout the algorithm's steps. Initially, we defined the sampling interval as Ψ_{speed}, where the strategies are represented as integers. Each agent i randomly selects m values from the sampling interval Ψ_{speed} and stores their indices in the set X_i. Each agent randomizes the strategies in the set X_{i_random}. As shown in Equation 3.9, the agents uniformly initialize the probabilities. After the optimization process, it is expected that the favorable strategy, which will have the highest probability value, will be selected.

Figure 10 also shows the inverse path (represented by a green arrow) used to find the strategy that was selected through the optimization process, which resulted in a strategy with the highest probability.

Figure 10 – Definition of the strategy set



Source: Elaborated by the author.

The Algorithm 1 outlines the procedure for the optimization process and consists of several steps as shown in the flowchart of the Figure 9 that we propose for solving the conflict. It begins by specifying the number of participating agents and determining the speed interval Ψ_{speed} through sampling.

As noted by Antoine *et al.* (2004), the temperature T in the optimization process can be fixed during the optimization. It is important to choose a value that has a good balance between exploration and exploitation. This can be done experimentally by observing the number of iterations required for convergence. In contrast to the generic PC algorithm presented in Section 3.1, where T is dynamically adjusted, in this specific application to CAVs, we chose to experimentally preset the value of T and maintain it fixed throughout the optimization and simulation. Therefore, the parameter is not updated at each step and this also eliminates the need for synchronizing the parameter with other agents, especially in dynamic scenarios.

The main loop of the algorithm starts by defining the set of strategies and their randomization process. Then a uniform distribution is applied on the set of strategies, because at the beginning of the optimization there is not enough information to increase or decrease the probability of a specific strategy.

Then the CAVs calculate the trajectories and send them by broadcast. The vehicles exchange pose information and calculate whether there is a collision and also calculate the private cost function. The agents share the information again, this time with the cost values. At this moment, all vehicles know the costs and start the optimization process to find the best strategy for each one that generates the lowest cost.

Algorithm 1 – PC for CAV Algorithm

```
$N \leftarrow$ number of agents
$\Psi_{speed} \leftarrow$ sampling speed interval
$T \leftarrow$ INITIALIZETEMPERATURE()
while not converged do
    $X \leftarrow$ GENERATESTRATEGIES($\Psi_{speed}$)
    $X_{random} \leftarrow$ RANDOMIZESTRATEGIES($X$)
    $q \leftarrow$ UNIFORMDISTRIBUTION()
    while max number of iteration not reached do
        $P_i \leftarrow$CALCULATETRAJECTORY($X_{random}$)
        SENDBROADCAST($P_i$)
        $P_{(i)} \leftarrow$ GETDATAFROMBROADCAST()
        $g \leftarrow$ PRIVATECOSTCALCULATION($P_i$, $P_{(i)}$)
        SENDBROADCAST($P_i$, $g$)
        $G \leftarrow$ GETDATAFROMBROADCAST()
        $q \leftarrow$ UPDATEDISTRIBUTION($X_{random}, q, T$)
        if CONVERGED($G$) then
            break
        end if
    end while
end while
```

3.4 Final Remarks

In this chapter, we introduce the Probability Collectives approach in its generalized form that can be used to optimize functions and benchmarks as shown in Appendix A.

According to Kulkarni, Tai and Abraham (2015), the variables of a function can be associated to individual agents, who can propose strategies to minimize the function. In the context of CAVs, each agent has an associated function that represents its private cost function. The goal is to sum these functions through the weighting method and minimize the total cost for all vehicles.

Therefore, this method can be adequately used for conflict resolution in a decentralized manner, since the CAVs have the ability to communicate, enabling the calculation of the weighting method.

We present the problem statement and define the scenarios used to conduct the experiments. We also demonstrate the adaptation of the method for CAVs through the mathematical formulation, as well as the main algorithm for conflict resolution.

CHAPTER

4

SIMULATION TOOLS AND EVALUATION METRICS

To evaluate the proposed method, we employ a microscopic traffic simulator that can simulate the behavior of multiple vehicles. We also compare the performance of our method with conventional methods already implemented in the simulator. To facilitate the development of our method, supplementary modules were necessary. The details of the simulator and its configuration are discussed in Section 4.1.

The simulator does not have an implementation of a feature for obtaining data on the vehicles' trajectories. That is, there is no function to relate the vehicle's position and time. It is crucial for the generating strategies step, it is worth remembering that the strategies are changes in speed in relation to the original plan of the vehicle, we need to calculate the positioning of the vehicles for each change in speed, which will be used for the collision detection. To overcome this, we developed a trajectory estimation method, described in Section 4.2.

The trajectory is represented by points in continuous space related to time, however, we also want to use information of the vehicles' dimensions to turn the problem more realistic. To this end, we employ a bounding box generation method to create a representation of the vehicle dimensions around the calculated points, as described in Section 4.3. With this information, we can then use an intersection detection algorithm to check for collisions between the proposed strategies. This process is outlined in Section 4.4.

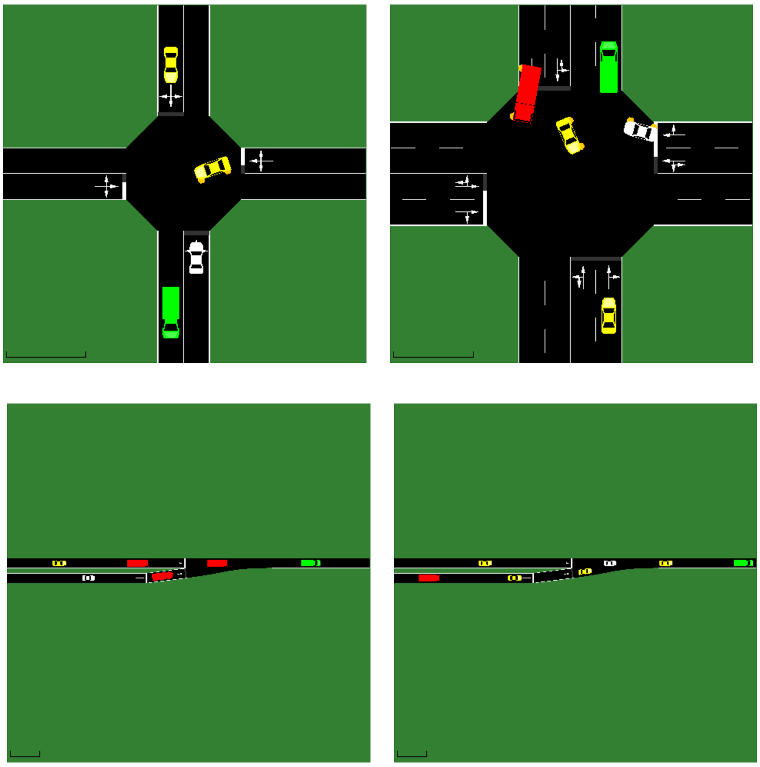
Finally, we developed an algorithm to manage strategies and CAVs within the simulation tool. This algorithm separates the maneuver into two phases, speed adjustment and maneuver. This process is presented in Section 4.5.

4.1 Simulation of Urban MObility

The Simulation of Urban MObility (SUMO) is an open-source microscopic simulator developed and maintained by the German Aerospace Center (BEHRISCH *et al.*, 2011) (LOPEZ *et al.*, 2018). SUMO has a very active and growing community and has been widely adopted by researchers in the field of ITS to investigate and propose alternatives for transportation systems and evaluate traffic control strategies. SUMO is also a multi-modal, discrete-time, and continuous-space simulator (KRAJZEWICZ *et al.*, 2012).

As a microscopic traffic simulator, SUMO assigns a unique identifier to each vehicle in the simulation and defines the vehicle's parameters, such as dimensions, maximum speed and acceleration, departure time, routes, and the car-following and lane-change models. Figure 11 presents some examples of simulations in the SUMO environment.

Figure 11 – Examples of simulations in SUMO



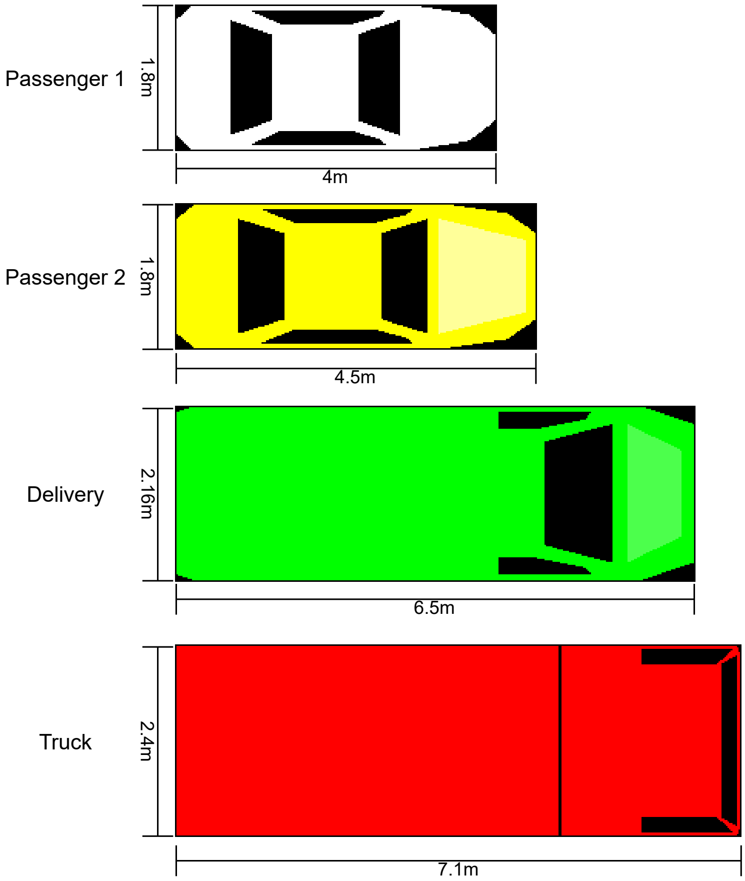
Source: Elaborated by the author.

To control the simulation in real-time, SUMO provides a tool called Traffic Control Interface (TraCI). It is an Application Programming Interface (API) for on-line access to the simulation data and allows getting and setting simulation parameters such as vehicles' speed, position, set speed, force a brake and request a lane-change maneuver.

In order to increase randomness in our experiments, we selected 4 different types of vehicles, as defined in SUMO with few modifications and we named it Passenger1, Passenger2, Delivery and Truck. In this study, we utilized the Federal Highway Administration's vehicle weight classification system to differentiate between different vehicle classes. This system has been used to create unique private cost functions for each vehicle that we defined. The classification determine Class 1 is any vehicle weighing less than 6000lbs, Class 2 between 6001lbs and 10000lbs and Class 3 between 10001lbs and 14000lbs. The assigned weights were Passenger1 with 6000lbs, Passenger2 with 8000lbs, Delivery with 10000lbs and Truck with 14000lbs. Passenger1 is the referral value with all others being a rounded percentage of that value. For example, Passenger1 is 1 and Passenger2 is 1.3 which is 1.3 times the reference weight.

Figure 12 illustrates the dimensions of the vehicles, while Table 1 provides information on their acceleration, deceleration, maximum deceleration, and probability of appearance during the simulation. The probability values were chosen empirically, with a focus on ensuring that passenger vehicles have the highest probability of appearing. All car-following models are based on Krauss, Wagner and Gawron (1997) and Krauss (1998). Table 2 lists all the parameters used for the vehicles during the simulations. While automatic lane change is disabled in our proposed approach, other comparison algorithms may utilize this feature.

Figure 12 – Vehicles dimensions



Source: Elaborated by the author.

Table 1 – Vehicles' parameters

Vehicle Type	Acceleration (m/s^2)	Deceleration (m/s^2)	Percentage (%)	Weight cost w
Passenger 1	2.6	4.5	35	1.0
Passenger 2	2.6	4.5	35	1.3
Delivery	2.6	4.5	20	1.6
Truck	1.3	4.0	10	2.6

Source: Research data.

Table 2 – SUMO Simulation paramaters

Parameter	Value
random	true
time to teleport	-1.0
collision action	warn
lane change duration	5.0
step length	0.1
tau	1.0
sigma	0.0
speedFactor	normc(1,0.025,0.925,1.075)
carFollowModel	Krauss
lcStrategic	0.5
lcCooperative	0.5

Source: Research data.

4.2 Position estimation

Position estimation is the process of determining the position of an object in some coordinate system. In this study, we use position estimation to predict the future positions of vehicles in order to detect potential collisions and validate strategies.

To estimate the future positioning of vehicles over time, we employ two methods: the Kinematic model and Absolute displacement. This was necessary because the SUMO simulator, which is based on car-following and lane-changing models, does not include information about the trajectory of vehicles during the simulation execution.

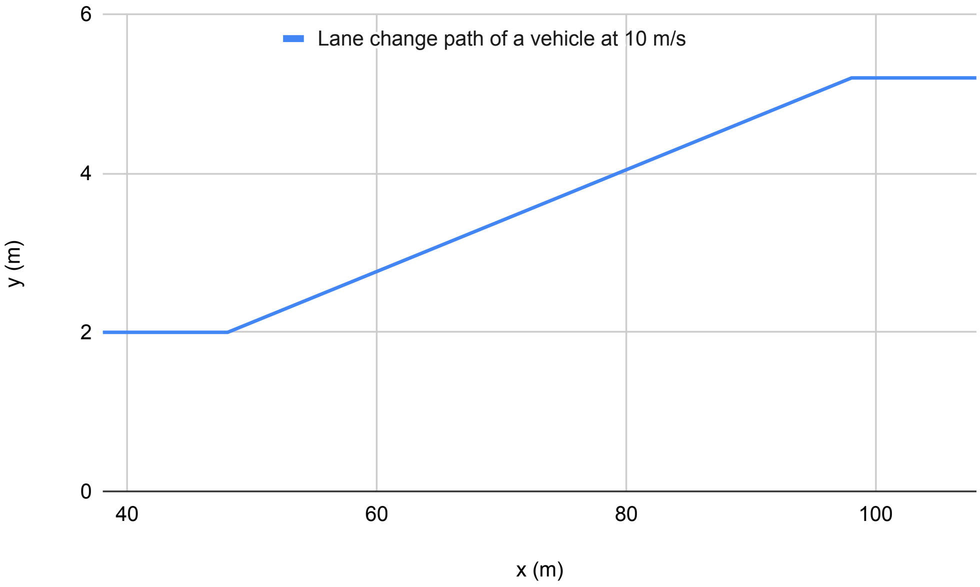
The Kinematic model uses information about the current position, velocity, and acceleration of a vehicle to predict its future position. The Absolute displacement method, on the other hand, estimates the future position of a vehicle based on its distance traveled using the SUMO road network information. By using these two methods, we were able to accurately estimate the future positioning of vehicles in the SUMO simulator, enabling us to detect potential collisions and validate strategies.

4.2.1 Kinematic model

Firstly, we use a kinematic model to describe the vehicle positioning concerning time. We used this approach in cases where the vehicle drove on a straight road or changed lanes.

With respect to lane-changing, the SUMO default lane change path can be approximated to a straight trajectory. The Figure 13 shows the lane-changing trajectory of a vehicle driving at 10 m/s. We can see that the path is a straight line, and we adopt that the vehicle always changes lanes with a fixed angle θ. Klischat *et al.* (2019) proposed an alternative method to lane-changing by coupling a motion planning in the simulator that can be used to generate a more realistic trajectory.

Figure 13 – Lane-changing trajectory.

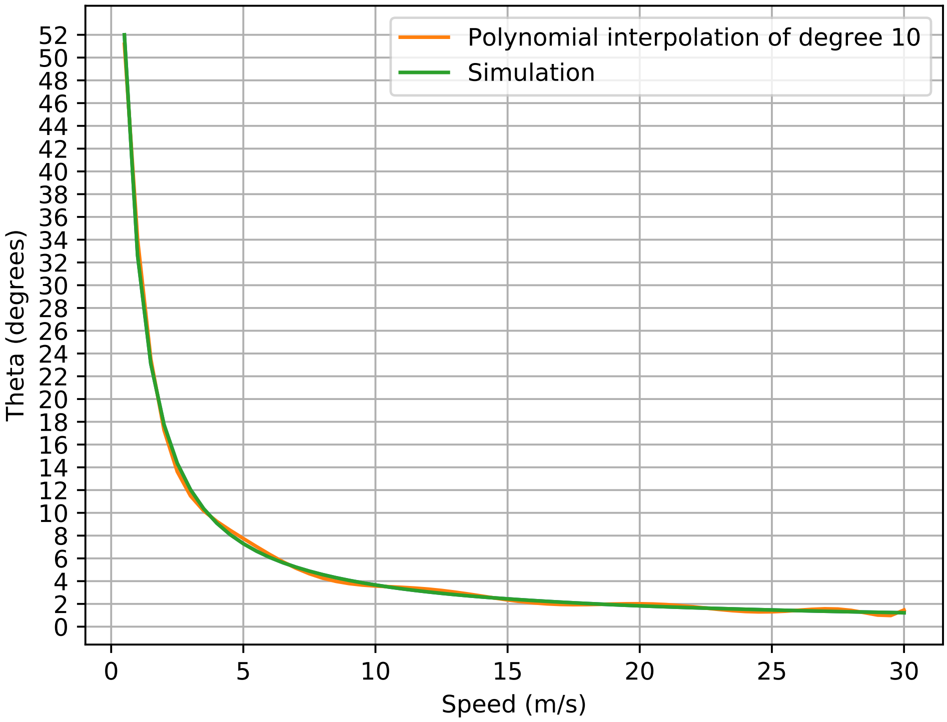


Source: Elaborated by the author.

As we use SUMO's car-following and lane-changing models, the simulator does not provide motion planning. To deal with this problem, we executed 60 simulations of lane changes on a straight road with speeds varying between 0.5 and 30 m/s with increments of 0.5 m/s for each simulation. We collected the angle data from each simulation to model a curve and make a polynomial interpolation fitting to calculate the angle θ for any speed within the closed interval [0.5, 30] m/s.

The graph in Figure 14 presents the data collected from the simulation in green and orange, the result of polynomial interpolation of degree 10. We opted for a high-degree polynomial to have higher accuracy in the positioning estimation and avoid errors generated by this process. Therefore, the R-squared (R2) of this interpolation is 0.998.

Figure 14 – Theta interpolation.



Source: Elaborated by the author.

The resulting solution was a polynomial with the following coefficients set: [5.35410503e-10, -8.73918338e-08, 6.16904742e-06, -2.46737090e-04, 6.15496682e-03, -9.93795987e-02, 1.04419273e+00, -7.00644016e+00, 2.87424579e+01, -6.68907190e+01, 7.82450261e+01], we present these results in scientific notation to preserve accuracy. Then, we can use Equation 4.1 to calculate every θ in the closed interval [0.5, 30], the $interpolation(s)$ function denotes the polynomial parameterized with s as speed.

$$\theta = interpolation(v) \tag{4.1}$$

To determine the vehicle position at time t, we first need to calculate acceleration/deceleration using the Equation 4.2 where v_0 is the initial speed and v is the final speed considering the accepted speed strategy.

$$v = v_0 + at \tag{4.2}$$

Then, we can calculate the vehicle positioning with the following set of kinematic Equations 4.3 and 4.4. The k denotes the step and x and y are the Cartesian coordinate positioning. The terms v_{0x} and a_x are the velocity's and acceleration's decomposition in the x direction, respectively. The v_{0y} and a_y are the velocity's and acceleration's decomposition in the y direction respectively. The t is the discrete amount of time.

$$x_{k+1} = x_k + v_{0x}t + \frac{1}{2}a_xt^2 \tag{4.3}$$

$$y_{k+1} = y_k + v_{0y}t + \frac{1}{2}a_yt^2 \tag{4.4}$$

The decomposition of velocity and acceleration in both x and y axes are given by Equations 4.5, 4.6, 4.7, 4.8.

$$v_x = vcos(\theta) \tag{4.5}$$

$$v_y = vsin(\theta) \tag{4.6}$$

$$a_x = acos(\theta) \tag{4.7}$$

$$a_y = asin(\theta) \tag{4.8}$$

4.2.2 Absolute displacement

In this approach, we use the route data provided by the simulator to calculate the vehicles' displacement. In SUMO, it is necessary to assign a route to every vehicle in the configuration files before the simulation. We can also change the vehicles' route during the simulation to perform a rerouting. In this way, it is possible to determine in advance the planning of the vehicle's route, i.e., we know which roads and junctions will be used by the vehicles. Here, we will not considering lane-changing neither rerouting, as the vehicles will follow the initial planning. Thus, it is possible to determine in advance which lanes the vehicles will drive during navigation.

The Traci API provides functions for accessing map data and vehicles' data as positioning and orientation information. With it, we can obtain information about which route, roads and lanes the vehicles will follow and vehicles' displacement in the current road. These map data are very useful to get information about roads' geometry as shape and length, and we used to calculate vehicles' displacement. Some Traci functions are shown generically in Algorithm 2.

To calculate the vehicle displacement, we again need to calculate acceleration/deceleration using the Equation 4.9 where v_0 is the initial speed and v is the final speed considering the accepted speed strategy.

$$v = v_0 + at \tag{4.9}$$

Then we calculate the displacement step by step according to Equation 4.10. The term k represents the step, x represents the absolute displacement and not a Cartesian position. The speed is denoted by v_k and t is the discrete amount of time.

$$x_{k+1} = x_k + v_k t + \frac{1}{2}at^2 \tag{4.10}$$

Equation 4.11 denotes the displacement with acceleration (deceleration) as x_{phase1} and Equation 4.12 denotes the displacement with constant speed as x_{phase2}, according to the strategies phases mentioned in Figure 16.

$$x_{phase1} = \sum_{0}^{k} x_k \quad \text{where} \quad x_{k+1} = x_k + v_k t + \frac{1}{2}at^2 \tag{4.11}$$

$$x_{phase2} = \sum_{0}^{k} x_k \quad \text{where} \quad x_{k+1} = x_k + v_k t \tag{4.12}$$

In the Equation 4.13, we denote Δx as the absolute displacement.

$$\Delta x = x_{phase1} + x_{phase2} \tag{4.13}$$

In Algorithm 2, we present how the position of a vehicle can be determined within a road using the displacement Δx calculated earlier. Initially, we attribute to d the sum of the displacement Δx with the vehicle's current position on the road. Notice that the latter is not the Cartesian position, but the vehicle's offset in the current road. We find vehicle's current road segment and assign it as the initial segment. Then, we calculate the difference from d and the road length. If the difference is greater than zero, we subtract the road length from d, otherwise we return d and the road segment index to calculate the Cartesian positioning of the vehicle using Traci.

Algorithm 2 – Find position on the segment list

```
procedure GETPOSITIONONSEGMENT($currentPosition$, $initialSegment$, $segmentList$, $\Delta x$)  ▷ currentPosition is the position in the initialSegment
    $d \leftarrow \Delta x + currentPosition$
    for $i \leftarrow$ FINDSEGMENT($initialSegment$), $roadList.size$ do  ▷ findSegment should be a function to find the index of the current segment and size is the total number of elements in roadList
        if ($d-$ ROADLENGTH($roadList[i]$)) $\geq 0$ then▷ roadLength should be a function that return the length of the segment in meters (or other unit)
            $d \leftarrow d-$ROADLENGTH($roadList[i]$)
        else
            return $d$, $segment\ i$
        end if
    end for
    return Vehicle out of simulation  ▷ Vehicle completed the route
end procedure
```

4.3 Oriented Bounding Box

The Oriented Bouding Box (GOTTSCHALK; LIN; MANOCHA, 1996) method is one of the more practical ways of describe the vehicles' dimensions with orientation in space. The geometric center of the vehicle was used as a reference point to calculate the positioning and orientation. We also considered the dimensions of the vehicles assuming a rectangular shape, and therefore it is necessary to know the width and length of the vehicles to define a rectangular bounding box around them.

To define the vertices of the rectangular bounding box, we employed a set of parametric Cartesian equations (presented as Equations 4.14 and 4.15) based on the parametric equations of the Lamé (1818) curve.

$$x = lx(|\cos p|\cos p + |\sin p|\sin p) \tag{4.14}$$

$$y = ly(|\cos p|\cos p - |\sin p|\sin p) \tag{4.15}$$

In Equations 4.14 and 4.15, we consider a rectangle with half-length and half-width denoted as lx and ly, respectively. This rectangle, referred to as the bounding box, can have the same size as the vehicles or be larger to account for positioning measurement errors. The parameter p represents all points of the edges of the the rectangle, but we are only concerned with the points that define the vertices of the rectangle. Therefore, the set of points of interest is $p = \{0, pi, pi/2, 3pi/2\}$.

Once the vertices of the bounding box have been determined, we can use the rotation matrix from Equation 4.16 and the translation matrix from Equation 4.17 to apply a rotation and translation to the points, positioning them correctly.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.16}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.17}$$

With this approach, we can also consider a safety margin in addition to the exact size of the vehicles. In this thesis we considered 1 m of length and 0.5 m width.

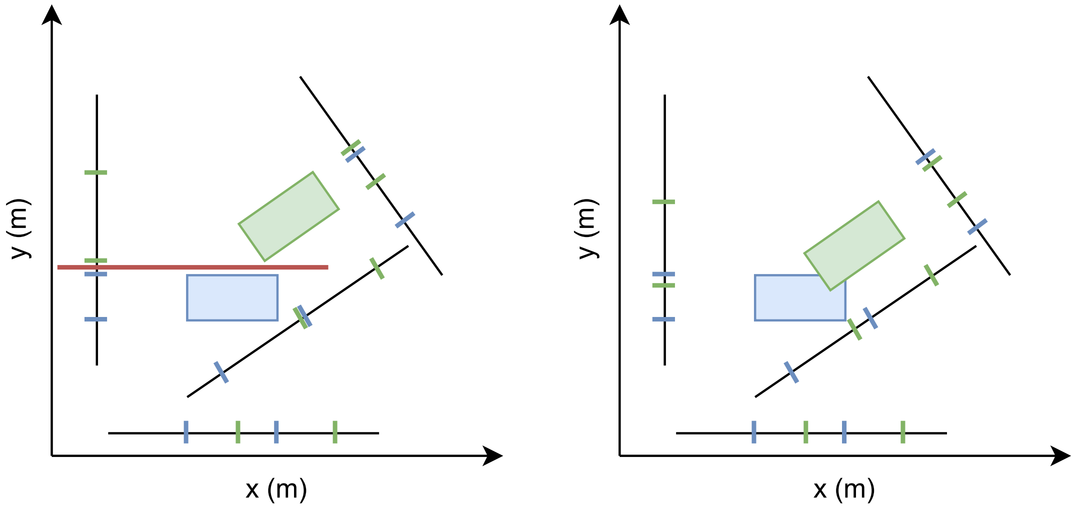
4.4 Collision Detection

Gottschalk (1996) proposed a method called Separating Axis Theorem (SAT) that is used to determine whether or not two convex shapes are intersecting. It is a useful tool in computer graphics and game programming for collision detection. SAT states that if it is possible to draw a line that separates the two shapes, then they are not intersecting. Thus, they will not be generating a collision.

The first step of the algorithm is to determine the edges and normals of the two shapes we want to test for intersection. In the Figure 15, the black lines in the graph represents the normal lines segments that we use for projection.

Once we have these lines, we can make a projection from the vertices of the rectangles to these lines. Figure 15 shows the projection of the vertices of the rectangles through dashes on the black lines. We can use the SAT to check for overlap along each of them. If there is at least one non-overlapping projection, then we can draw a line that separates the two shapes and they are not intersecting, as illustrated by the red line in Figure 15a. If there is an overlap along all projections, then it is not possible to draw a line that separates the two shapes, and they are intersecting, as shown in Figure 15b.

Figure 15 – Separating Axis Theorem



(a) No intersection.

(b) Intersection.

Source: Elaborated by the author.

4.5 Strategy Management

In order to manage the vehicle's strategies, we developed two algorithms and a data structure to support the execution during the SUMO simulation. The Strategy Management runs every simulation step, but when the agents finish the optimization and reach a consensus on which strategies each one will execute, they store the strategy information in a list and execute the strategies over time.

The data structure and the maneuver types are shown in Table 3. We simplify the maneuvers into two types: Slowdown and Lane change. Slowdown is used to assign a speed to the vehicle for a given time (duration) in a specific timestep, and only speed reductions are used. It is also used to maintain a constant speed. Lane change refers to changing lanes, where the lane index represents the desired lane. According to Toledo and Zohar (2007), the average lane change time is 4.6 seconds, with a range from 1.0 to 13.3 seconds. In the simulation, the lane change time for all vehicles is set to 5 seconds. It is worth to mention that both the Slowdown maneuver, to maintain a constant speed, and the Lane change maneuver can be executed simultaneously.

Table 3 – Data structure for strategy management

Type of maneuver	**Data**	**Duration**	**Timestep**
Slowdown	speed	duration	timestep
Lane change	lane index	duration	timestep

Source: Research data.

After consensus, each agent creates a maneuver queue with the data type of the Table 3 and these maneuvers are performed by each vehicle. In this structure we indicate the type of maneuver, the specification of how to execute, the duration time and the start timestamp. In general, this queue of maneuvers can be stored in the decision-making system of an autonomous vehicle, where each item is sent to the control or trajectory planning system at its appropriate timestamp. In this work, we perform the maneuvers using the functions provided by SUMO, example, Slowdown is a function that can smoothly decrease as well as increase the speed of vehicles, and Lane change is a function that performs the lane change maneuver for the lane index requested.

Agents schedule strategies and execute them according to the time and order they were added to the queue, as shown in Algorithm 3. For example, if an agent requests to slow down and change lanes, it will queue up a speed decrease maneuver, followed by a lane change with a simultaneous maneuver to maintain speed. After the execution, they remove the strategy from the list.

The Algorithm 4 is an example of how we use the Strategy Management inside the simulation main loop. At each simulation step the agents call the strategy manager. There are two ways to implement, the first considering that strategy manager runs for all agents, the second considering that each agent has its own manager (we selected the second for implementation). When there are no strategy in the list, the manager skips the execution, and the simulation continues.

We divide strategy management into two phases, speed adjustment and maneuver ex-

Algorithm 3 – Strategy Management

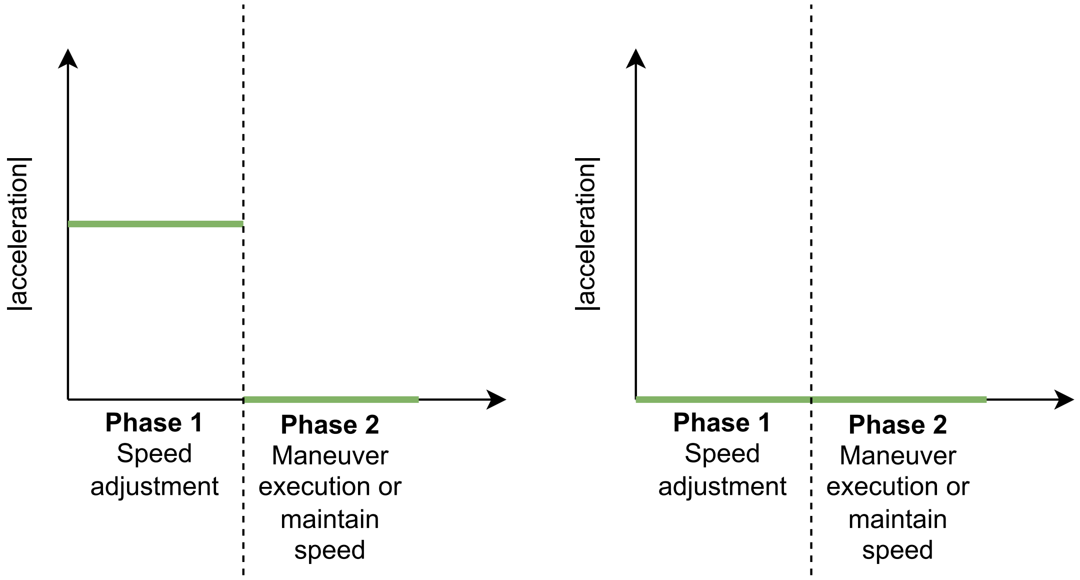
```
procedure EXECUTEMANEUVER(maneuverList, simulationStep) ▷ maneuverList is the list of maneuvers with the proposed data structure presented in Table 3
    for all $maneuver \in maneuverList$ do
        if $maneuver.timestamp = simulationStep$ then
            switch $maneuver.type$ do
                case $slowdown$
                    $speed \leftarrow maneuver.data$
                    SLOWDOWN($speed$, $maneuver.duration$)
                case $lanechange$
                    $laneIndex \leftarrow maneuver.data$
                    CHANGELANE($laneIndex$, $maneuver.duration$)
            REMOVEFROMLIST($maneuver$)                    ▷ Remove maneuver from the list
        end if
    end for
end procedure
```

Algorithm 4 – Strategy Management Execution

```
while the simulation is not finished do
    $step \leftarrow$ RUNSIMULATIONSTEP()                    ▷ run one simulation step
    for all $agent \in agentList$ do
        $maneuverList \leftarrow$ GETMANEUVERLIST($agent$)
        EXECUTEMANEUVER($maneuverList$, $step$)
    end for
end while
```

ecution. Figure 16 illustrates how the two phases work. We can assign different amounts of time to run Phase 1 and Phase 2, but the time for both phases must be the same for all agents involved in the negotiation. In Phase 1, the vehicles adjust their speeds and increase the acceleration/deceleration value (as shown in absolute terms in Figure 16a) if it is necessary to change the speed. Furthermore, if the vehicle is already driving at the proposed speed strategy, the speed will remain constant, as shown in Figure 16b. In phase 2, we assume that the vehicle remains at a constant speed and performs a lane change maneuver if requested.

Figure 16 – Phases to strategy execution



(a) Phase 1 with acceleration. Phase 2 maneuver execution or maintain speed.

(b) Phase 1 with constant speed. Phase 2 maneuver execution or maintain speed.

Source: Elaborated by the author.

4.6 Scenarios Design

This section describes the scenarios used to conduct our experiments and evaluate our method. To demonstrate the versatility of our approach, we have designed four distinct test scenarios. The first two scenarios are cross intersections, one with a single lane and the other with multiple lanes and more complex routes. An on-ramp merge scenario simulates the merging of CAVs from an on-ramp. A lane change scenario simulates the movement of vehicles between lanes. These test scenarios have been carefully designed to represent real-world situations and test our method's performance under different conditions.

These scenarios were used for both static and dynamic experiments. In the case of dynamic experiments, the roads have communication zones used by the CAVs when accessing them. The zones were based on the work of Gaciarz, Aknine and Bhouri (2015), who proposed an inner and external area similar to the synchronization zones suggested by Philippe (2020), where communication and decision-making occur. Both authors also define a conflict zone and a shared zone for the intersection region.

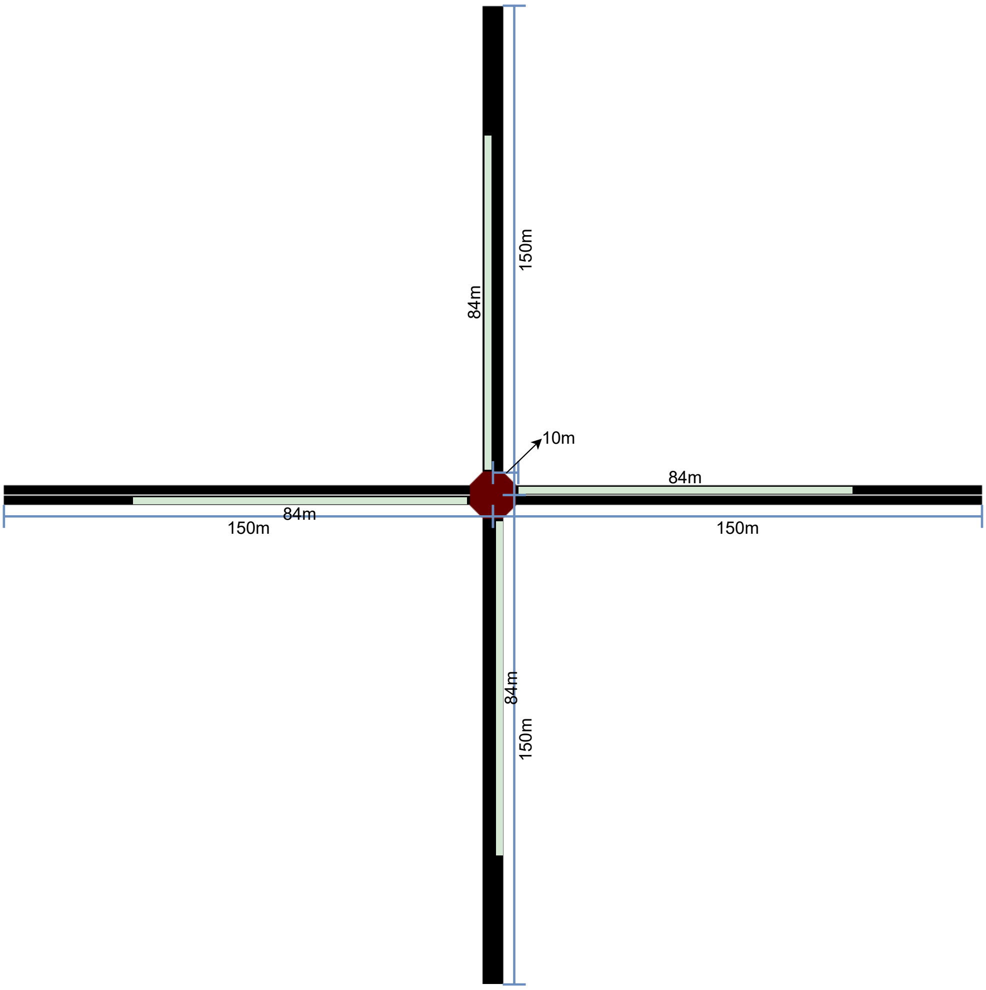
When a CAVs enters the communication and negotiation zone, it immediately sends out a broadcast message to the other CAVs in the area. If there are other CAVs within the zone, they will initiate a negotiation process. If, on the other hand, there are vehicles in the zone that have completed the negotiation, they will simply share the decisions that they made in an updated manner, allowing the new vehicle to make its own decision. The PC algorithm is used in the same way, the only difference is that the agents are already following a previously established strategy.

From a practical perspective, negotiation still takes place, but the vehicles always choose the same strategy that is currently being executed. Once the vehicle leaves the communication zone, it no longer needs to participate in the negotiation process and will not respond to any further messages of this region.

4.6.1 Crossroad 1-lane Scenario Design

Figure 17 illustrates the Crossroad with 1 lane scenario, including the dimensions and lengths of the roads. All roads have a maximum permitted speed of 60 km/h. The green region in the figure represents the designated communication and negotiation area, which was arranged equidistantly in all scenarios.

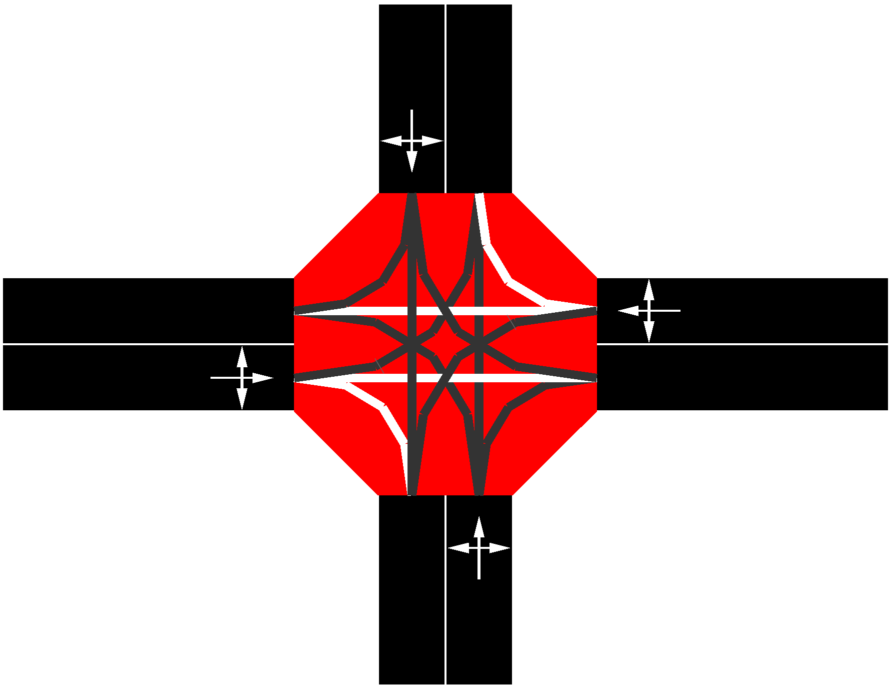
Figure 17 – Crossroad 1-lane Scenario Design



Source: Elaborated by the author.

Figure 18 shows the possible routes for all vehicles in the scenario. The red sector represents the conflict zone, and the lines depict the possible trajectories. The white lines indicate high-priority routes, while the gray lines indicate low-priority routes, meaning that vehicles must yield the right-of-way to higher-priority vehicles.

Figure 18 – Crossroad 1-lane Routes

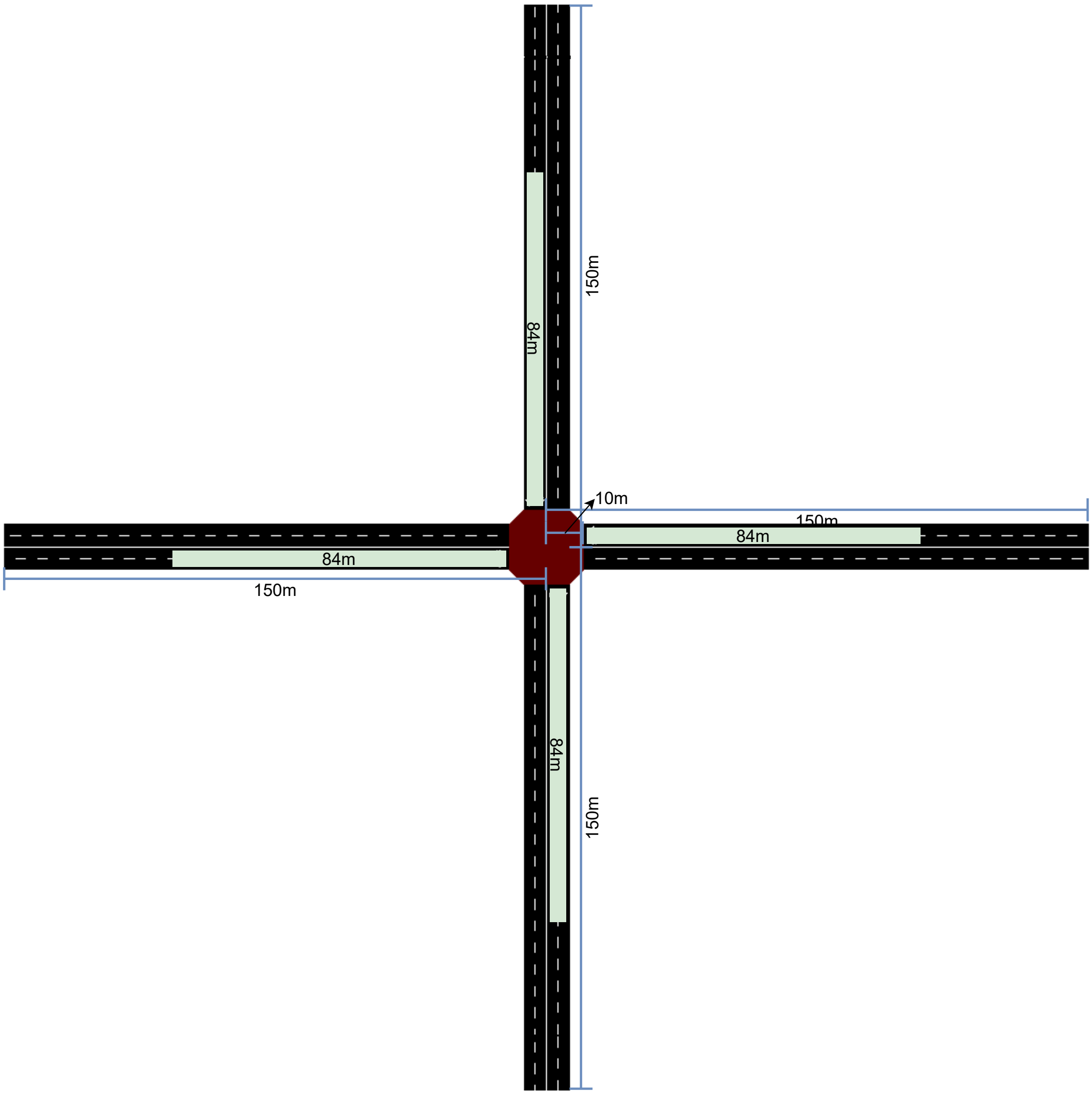


Source: Elaborated by the author.

4.6.2 Crossroad 2-lane Scenario Design

Figure 19 illustrates the Crossroad with 2 lanes scenario, including the dimensions and lengths of the roads. All roads have a maximum permitted speed of 60 km/h. The green region in the figure represents the designated communication and negotiation area, which was arranged equidistantly in all scenarios.

Figure 19 – Crossroad 2-lane Scenario Design

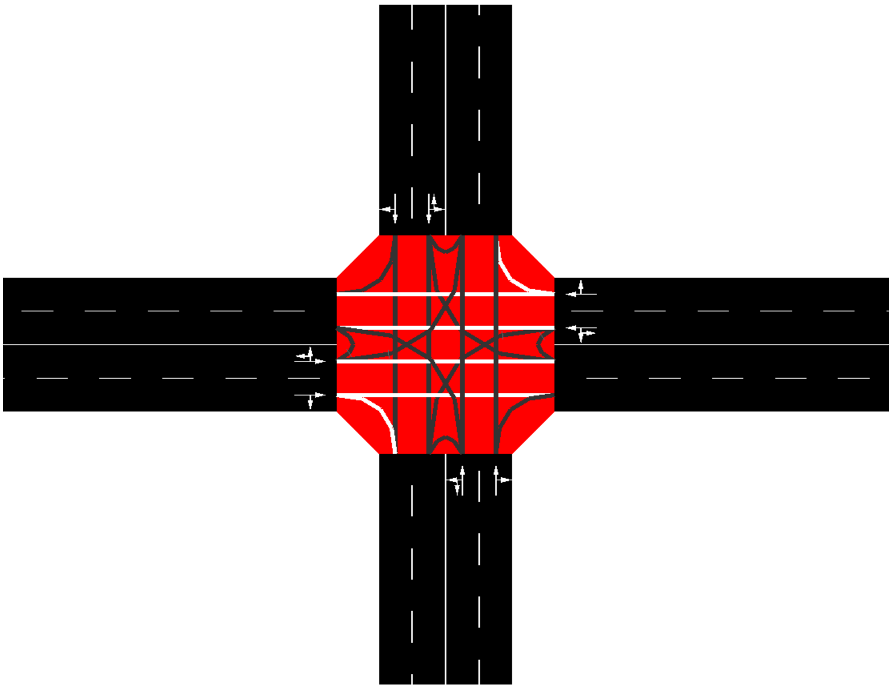


Source: Elaborated by the author.

Figure 20 illustrates the possible routes for all vehicles in the scenario. The red sector represents the conflict zone, and the lines depict the possible trajectories. The white lines indicate high-priority routes, while the gray lines indicate low-priority routes, meaning that vehicles must yield the right-of-way to higher-priority vehicles.

It is worth to mention that in this 2 lane scenario, vehicles can only make left turns if they are in the left lane and right turns if they are in the right lane. No lane changing or overtaking is permitted in this scenario.

Figure 20 – Crossroad 2-lane Routes

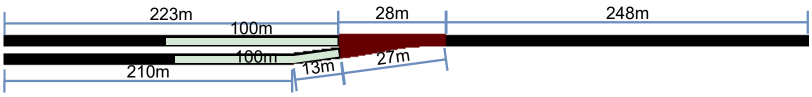


Source: Elaborated by the author.

4.6.3 Ramp Merge Scenario Design

Figure 21 illustrates the Ramp Merge with 1 lane scenario, including the dimensions and lengths of the lanes. All roads have a maximum permitted speed of 60 km/h. The green region in the figure represents the designated communication and negotiation area. The two routes have a merging negotiation zone of 100 meters.

Figure 21 – Ramp Merge Scenario Design

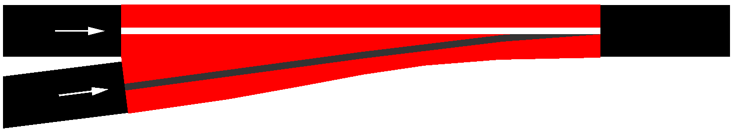


Source: Elaborated by the author.

Figure 22 presents the possible routes for both routes. The red sector represents the conflict zone, and the lines depict the possible trajectories. The white lines indicate high-priority routes, while the gray lines indicate low-priority routes, meaning that vehicles must yield the right-of-way to higher-priority vehicles.

In this case, the left lane has the highest priority and vehicles in the merging lane must yield the right-of-way to vehicles in the left lane.

Figure 22 – Ramp Merge Routes

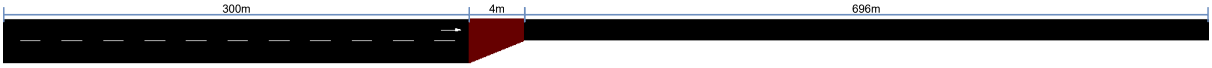


Source: Elaborated by the author.

4.6.4 Lane Change Scenario Design

The lane change scenario, presented in Figure 23, is represented simply by a 300m two-lane road where vehicles in the right lane are required to change to the left lane, as the right lane leads to a dead end. This scenario does not depict fixed communication zones, as communication for negotiation is dynamically activated when required by the CAVs.

Figure 23 – Lane Change Scenario Design



Source: Elaborated by the author.

4.7 Comparative Methods

SUMO provides some models for intersection management and lane-changing. They are very complex and has several parameters that tries to replicate real world behavior that can be used to study urban mobility. These models can also be used as benchmark as demonstrated in the study by Dongxin *et al.* (2021).

SUMO models intersections in a discrete manner, dividing them into cells that stores the right-of-way of vehicles based on their arrival time and time step. The model has links that connect roads and lanes, allowing vehicles to request passage through the intersection. Vehicles reserve their passage through the intersection using time slots, creating a structure that reproduces realistic behaviors in intersection scenarios (KRAJZEWICZ; ERDMANN, 2013).

SUMO uses a decision tree scheme to handle the lane-changing (ERDMANN, 2015) and regulates the ego vehicle speed in relation to the leader and follower respectively.

The first test is related to the scenario where the ego vehicle has a blocking leader. The procedure evaluates if it is feasible to either overtake or remain behind the leader. If overtaking is possible, the ego vehicle sends a request to the leader to reduce its speed. If the leader is not blocking, the speed is regulated to maintain a safe distance from the leader and prevent blocking. Finally, if there is no leader, sets the maximum safe speed.

The second test examines if there is a blocking follower. If the ego vehicle is able to perform a lane change, it requests the follower to reduce speed. The test also evaluates whether the follower should overtake the ego vehicle. If it is possible, ego vehicle reduce the speed to allow the follower to safely pass.

We also implemented a Traffic Light System (TLS) using SUMO to provide a comparison in scenarios involving intersections. In the experiments we used fixed cycles with a total duration of 60 seconds per cycle. The cycle is divided into two equal parts, each one dedicated to one road at the intersection. We also used the default SUMO traffic light duration time, wherein each half of the cycle is divided into 26 seconds of green light and 4 seconds of yellow light.

Finally, we use the metrics Average Travel Time, Average Speed, Average Waiting Time, Arrived Vehicles and Traffic Flow to evaluate our method and compare with SUMO and TLS methods. These metrics are widely used in several works in the literature (DONGXIN *et al.*, 2021), (LEVIN; REY, 2017), (SCHEPPERLE; BöHM; FORSTER, 2007), (LISSAC; DJAHEL; HODGKISS, 2019) and (WANG; CAI; LU, 2020). As mentioned in the Section 1.1, our main goal is to enhance the flow of vehicles and reduce travel time.

The Average Travel Time is the average time that the vehicles took to run the simulation, that is, the time spent passing through the scenario. The Average Speed is the average speed calculated in relation to all vehicles in the simulation. Average Waiting Time is the average time the vehicles drives below 0.1 m/s. The Arrived Vehicles are the vehicles that finished the simulation and the Traffic Flow is the number of vehicles that passed through the scenario in a given time, and we used simulations of 120 seconds.

4.8 Final Remarks

In this chapter, we present the methods used to perform the experiments. We show the necessary adaptations in the simulator to be able to run the strategies calculated by our approach, as well as the tools required for collision detection position estimation. Finally, we also present the methods and metrics used for comparison with other approaches.

CHAPTER

5

PROBABILITY COLLECTIVES IN STATIC SCENARIOS

In this chapter, we investigate the performance of our method in static scenarios to verify the optimization capacity in addressing conflict situations in four different scenarios.

We start by testing the capability of our model in resolving conflicts in a crossroad with one lane. The negotiation process is employed to solve the problem in a static manner, that is, there is no traffic flow and the CAVs that participates in the negotiation were previously inserted into the simulation. In this way, the negotiation will only occur once and when they finish executing the adopted strategies the vehicles go back to driving at the road speed limit.

Next, we evaluate the optimization behavior by comparing scenarios where CAVs have different private cost functions to scenarios where all CAVs have the same private cost function.

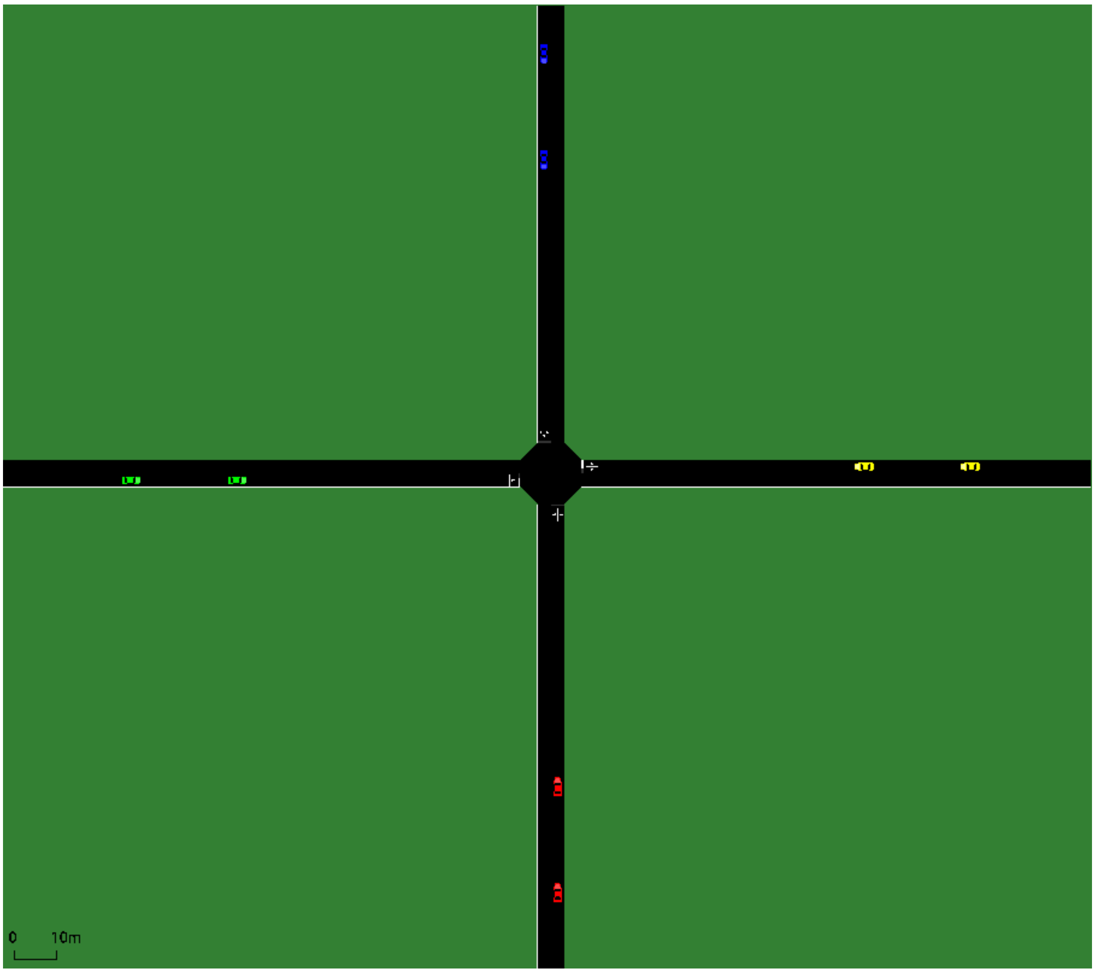
5.1 Number of strategies

In this experiment, we evaluated the relationship between the number of agents and the number of strategies, and also investigate the number of interactions required to resolve conflicts. It is important to note that each interaction represents two communication steps, as the vehicles must exchange their strategies with positioning information, calculate the costs, and then resend the costs to one another. In this static scenario, we intentionally arranged the positions of the vehicles in such a way that collisions would occur if no action was taken to prevent them. It is worth to mention that this scenario represents the worst case, as it represents a situation in which the vehicles must change their speeds in order to avoid accidents.

Figure 24 illustrates the arrangement of the vehicles in the scenario. The figure provides a visual representation of the position of the vehicles relative to one another and the layout of the environment in which the experiments were conducted.

The $|\Psi_{speed}|$ represents the size of the strategy set and m the maximum number of strategies that can be chosen, as we already defined in Section 3.2.

Figure 24 – Vehicles Static Positioning



Source: Elaborated by the author.

In our experiments, we start each round with 2 agents and gradually increase the number of agents up to 8. Once the agents find a set of strategies that successfully resolves the conflict, we stop the optimization process and accept this set of strategies. The vehicles then execute these strategies in order to avoid collisions. We set a stopping condition of a maximum of 50 iterations. If the agents reach this limit, the negotiation stops and it is assumed that the optimization is not feasible for the given number of agents and strategies.

To evaluate the performance of different strategies for conflict resolution, we conducted 25 rounds of experiments for each different number of agents. The results of these experiments are presented in Tables 4, 5, 6 and 7 which show the mean, mode, and median statistical results for the number of interactions required to resolve conflicts. We chose to include the mode in the tables to highlight the most frequent numbers of interactions and the median to demonstrate the tendency towards centrality.

Table 4 – Evaluation of the number of interactions for conflict resolution for strategy set $|\Psi_{speed}| = 6$ and the maximum number of strategies that can be chosen $m = 3$.

Number of Agents	**Average of Interactions**	**Mode of Interactions**	**Median of Interactions**
2	2	1	1
3	13.84	13	13
4	43.76	50	50
5	50	50	50
6	50	50	50
7	50	50	50
8	50	50	50

Source: Research data.

Table 5 – Evaluation of the number of interactions for conflict resolution for strategy set $|\Psi_{speed}| = 9$ and the maximum number of strategies that can be chosen $m = 5$.

Number of Agents	**Average of Interactions**	**Mode of Interactions**	**Median of Interactions**
2	1.04	1	1
3	1.44	1	1
4	5.24	1	3
5	11.04	2	7
6	20.44	50	16
7	40.08	50	50
8	50	50	50

Source: Research data.

Table 6 – Evaluation of the number of interactions for conflict resolution for strategy set $|\Psi_{speed}| = 11$ and the maximum number of strategies that can be chosen $m = 7$.

Number of Agents	**Average of Interactions**	**Mode of Interactions**	**Median of Interactions**
2	1	1	1
3	1.04	1	1
4	2.68	1	1
5	5.04	2	3
6	12.88	1	9
7	18.96	50	13
8	31.84	50	31

Source: Research data.

Table 7 – Evaluation of the number of interactions for conflict resolution for strategy set $|\Psi_{speed}| = 14$ and the maximum number of strategies that can be chosen $m = 10$.

Number of Agents	Average of Interactions	Mode of Interactions	Median of Interactions
2	1	1	1
3	1	1	1
4	1.08	1	1
5	1.76	1	1
6	2.52	1	2
7	3.8	2	3
8	7.92	1	4

Source: Research data.

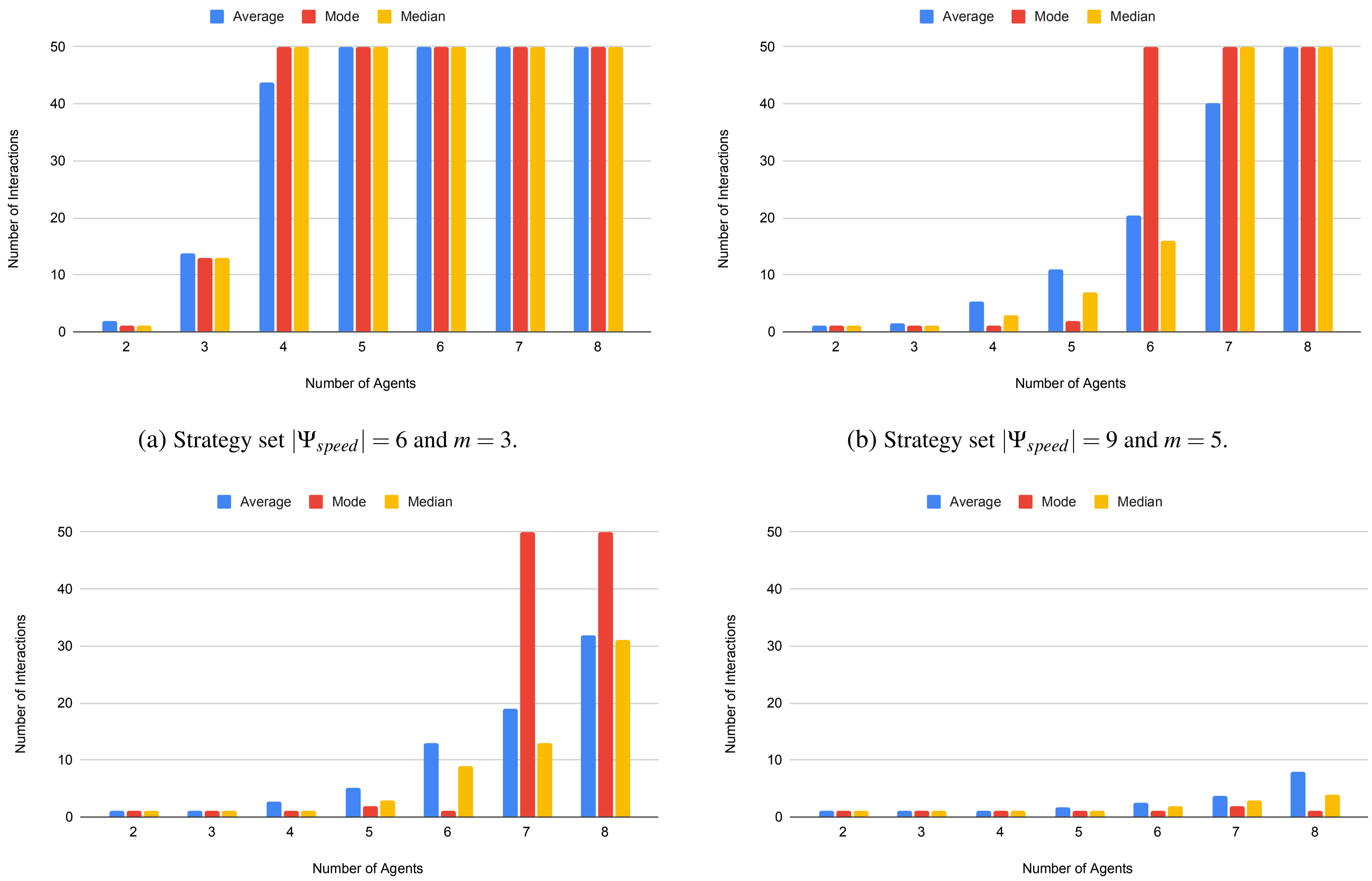
As we increase the number of agents in the scenario, we find that it becomes necessary to also increase the number of strategies in order to turn possible the convergence. This is because the vehicles must have a strategy for altering their speed profiles and avoiding collisions, particularly in this scenario where we have intentionally positioned the vehicles in a way that is likely to result in a collision if no action is taken to prevent it.

The results in Table 4 indicate that for 5 strategies, only 4 agents can resolve the conflict, with an average of 42.76 iterations. However, as the number of strategies increases, the optimization allows for more agents to resolve the conflict with fewer iterations. For example, with 9 strategies (Table 5), 4 agents can solve the conflict with an average of 5.24 iterations and with 9 strategies it is possible to resolve the conflict for 7 agents. With 11 strategies (Table 6), 8 agents can resolve the conflict with an average of 31.84 iterations, and with 14 strategies (Table 7), 8 agents can resolve the conflict with an average of 7.92 iterations.

Figure 25 presents a graphical illustration of the data found in Tables 4, 5, 6, and 7. It allows us to compare the number of iterations necessary for the algorithm to converge and resolve conflicts for different numbers of agents and varying strategy number.

In other words, as the complexity of the scenario increases with more vehicles interacting, it becomes more difficult to find a solution that allows all of the vehicles to safely navigate the environment. Therefore, a larger number of strategies may be needed to explore a wider range of options and find a solution that works for all of the vehicles.

Figure 25 – Comparison of the number of interactions and different number of strategies needed to resolve a conflict.



(a) Strategy set $|\Psi_{speed}| = 6$ and $m = 3$.

(b) Strategy set $|\Psi_{speed}| = 9$ and $m = 5$.

(c) Strategy set $|\Psi_{speed}| = 11$ and $m = 7$.

(d) Strategy set $|\Psi_{speed}| = 14$ and $m = 10$.

Source: Elaborated by the author.

5.2 CAVs with different private cost functions

In this section, we present the results of the negotiation in a static scenario, where the negotiation occurs only once. The number of vehicles is fixed and there is no traffic flow. The aim is to demonstrate the applicability of our method to various scenarios, as well as in optimizing not only equal private cost function (when all vehicles have the same private cost function), but also when there is different private cost functions among the vehicles.

In each scenario, we present three graphs: one showing the speed profile and global cost for when the CAVs have equal costs, another for when the CAVs have different costs, and a third graph that compares the chosen strategies. This strategy comparison shows the frequency with which a particular strategy was chosen by a vehicle of a determined class.

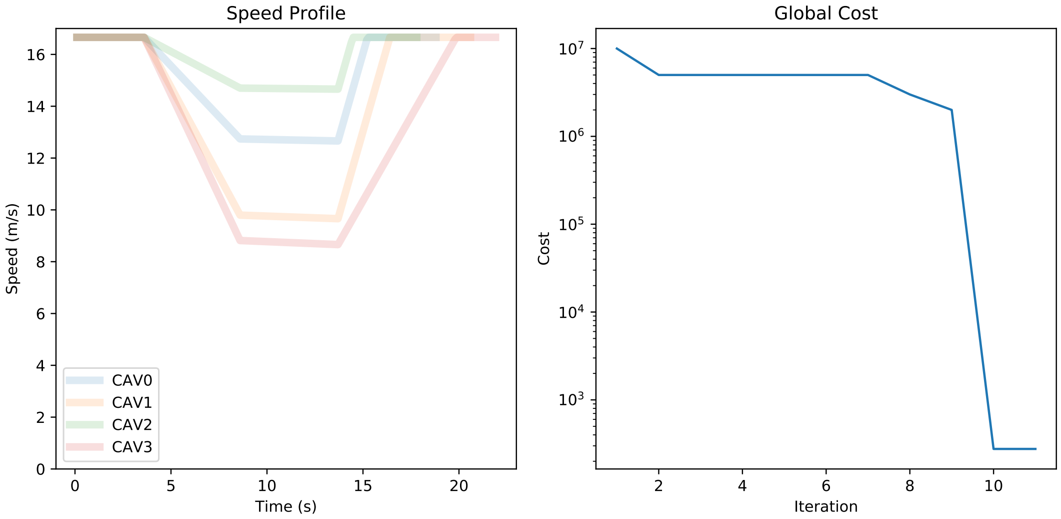
There are 9 strategies, ranging from 0 to 8, with strategy 0 having the lowest cost and strategy 8 having the highest cost. The blue bars represent the results when all vehicles have equal costs, while the orange bars represent the results when each vehicle has a different, unique cost function. The objective of this analysis is to verify the change in the profile of the strategy choice when setting different cost functions for the CAVs, remembering that the experiments were run 25 times for equal costs and 25 times for different costs.

5.2.1 Crossroad 1-lane

In the Crossroad 1-lane scenario, as described in Section 4.6.1, four vehicles were positioned at equal distances from the center of the intersection. The positioning of each vehicle was designed to force a conflict which is result in an imminent collision, for this each vehicle was placed on an arm of the crossing. At the beginning of the simulation, all vehicles start at 60 km/h, and after a predetermined fixed time, the vehicles began negotiating to resolve the conflict. Upon completion of the optimization process, the vehicles set the selected strategies and adjust their speed profiles to ensure a safe passage through the intersection.

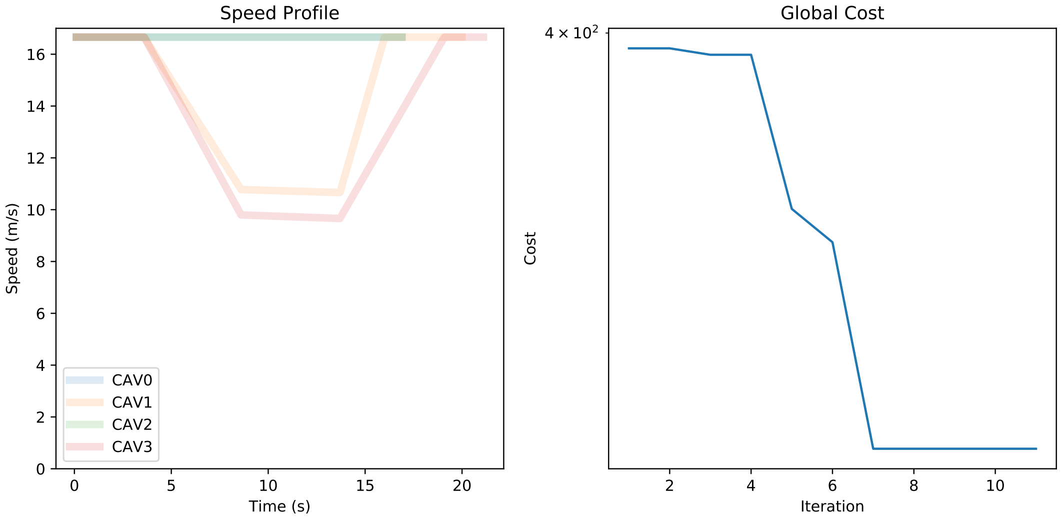
A sample of 25 experiments was taken and is presented in Figures 26 and 27, which shows the speed profile and the global cost of the agents. The negotiation began at 3.5s and after the optimization, the vehicles adjusted their speeds to safely navigate the intersection without collision, and subsequently returned to the road's speed limit. The cost was observed to decrease over the iterations, as the optimization process caused the agents to search for solutions that resolve the problem, although these solutions may not necessarily be the most optimal ones.

Figure 26 – Speed profile of CAVs with equal private cost functions and the evolution of the Global Cost over the iterations for the Crossroad 1-lane scenario



Source: Elaborated by the author.

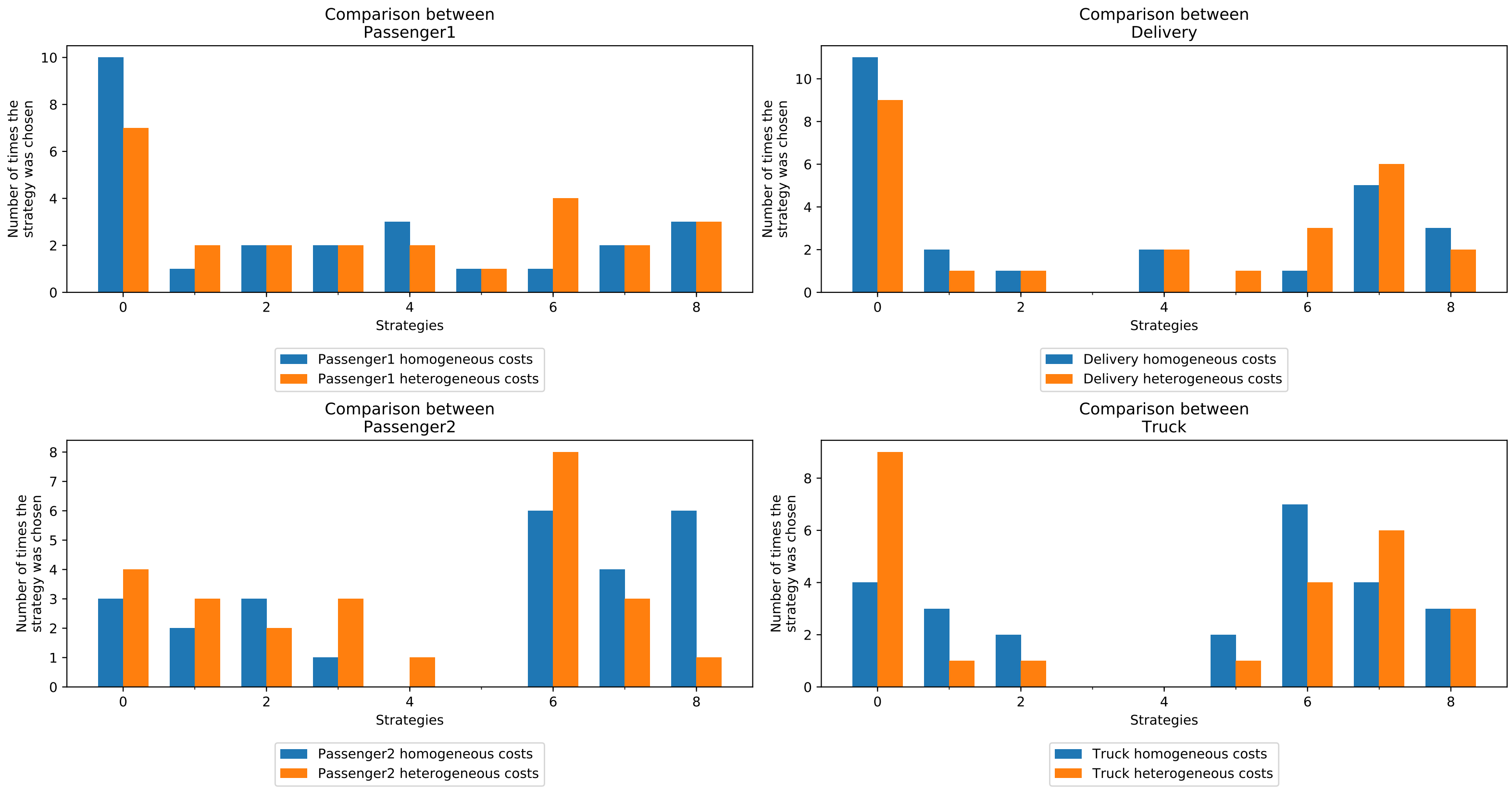
Figure 27 – Speed profile of CAVs with different private cost functions and the evolution of the Global Cost over the iterations for the Crossroad 1-lane scenario



Source: Elaborated by the author.

Figure 28 presents the comparison of strategies selected by the vehicles. The blue bars represent the experiment where the vehicles have the same private cost function and the orange bars, different private cost functions. In this case, we can verify that the strategies profile is not significantly changed for the class Passanger1, Passenger2 and Delivery. We can noticed a decrease in the frequency of the best strategies being chosen by the Passenger1 and Delivery classes. However, for the Truck class, there was a significant increase in lower cost strategies.

Figure 28 – Comparison of strategies chosen by CAVs with equal and different private costs for the Crossroad 1-lane scenario



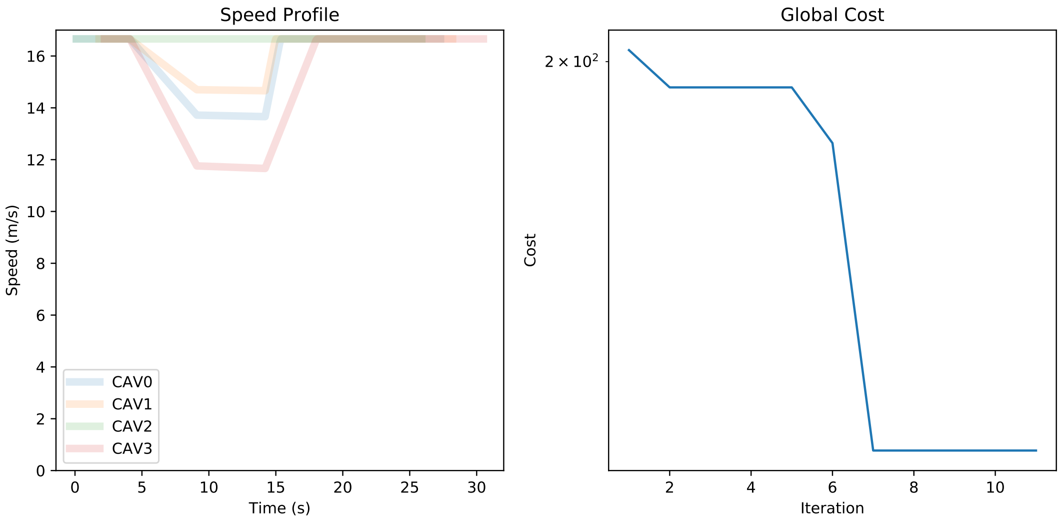
Source: Elaborated by the author.

5.2.2 Crossroad 2-lanes

In the Crossroad 2-lanes scenario, as described in Section 4.6.2, four vehicles were positioned at equal distances from the center of the intersection. The positioning of each vehicle was designed to force a conflict which is result in an imminent collision, for this each vehicle was placed on an arm of the crossing. At the beginning of the simulation, all vehicles start at 60 km/h, and after a predetermined fixed time, the vehicles began negotiating to resolve the conflict. Upon completion of the optimization process, the vehicles set the selected strategies and adjust their speed profiles to ensure a safe passage through the intersection.

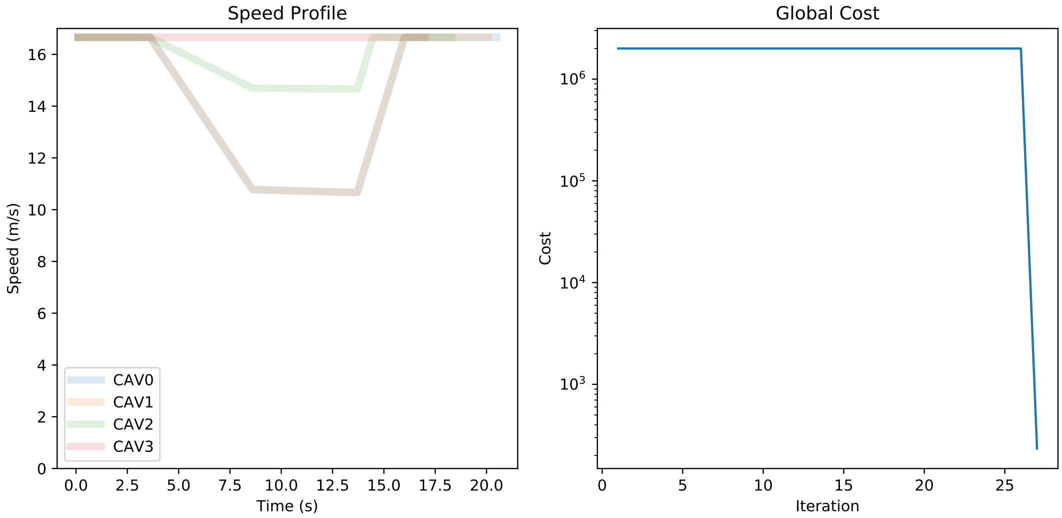
A sample of 25 experiments was taken and is presented in Figures 29 and 30, which shows the speed profile and the global cost of the agents. The negotiation began at 3.5s and after the optimization, the vehicles adjusted their speeds to safely navigate the intersection without collision, and subsequently returned to the road's speed limit. The cost was observed to decrease over the iterations, as the optimization process caused the agents to search for solutions that resolve the problem, although these solutions may not necessarily be the most optimal ones.

Figure 29 – Speed profile of CAVs with equal private cost functions and the evolution of the Global Cost over the iterations for the Crossroad 2-lanes scenario



Source: Elaborated by the author.

Figure 30 – Speed profile of CAVs with different private cost functions and the evolution of the Global Cost over the iterations for the Crossroad 2-lanes scenario



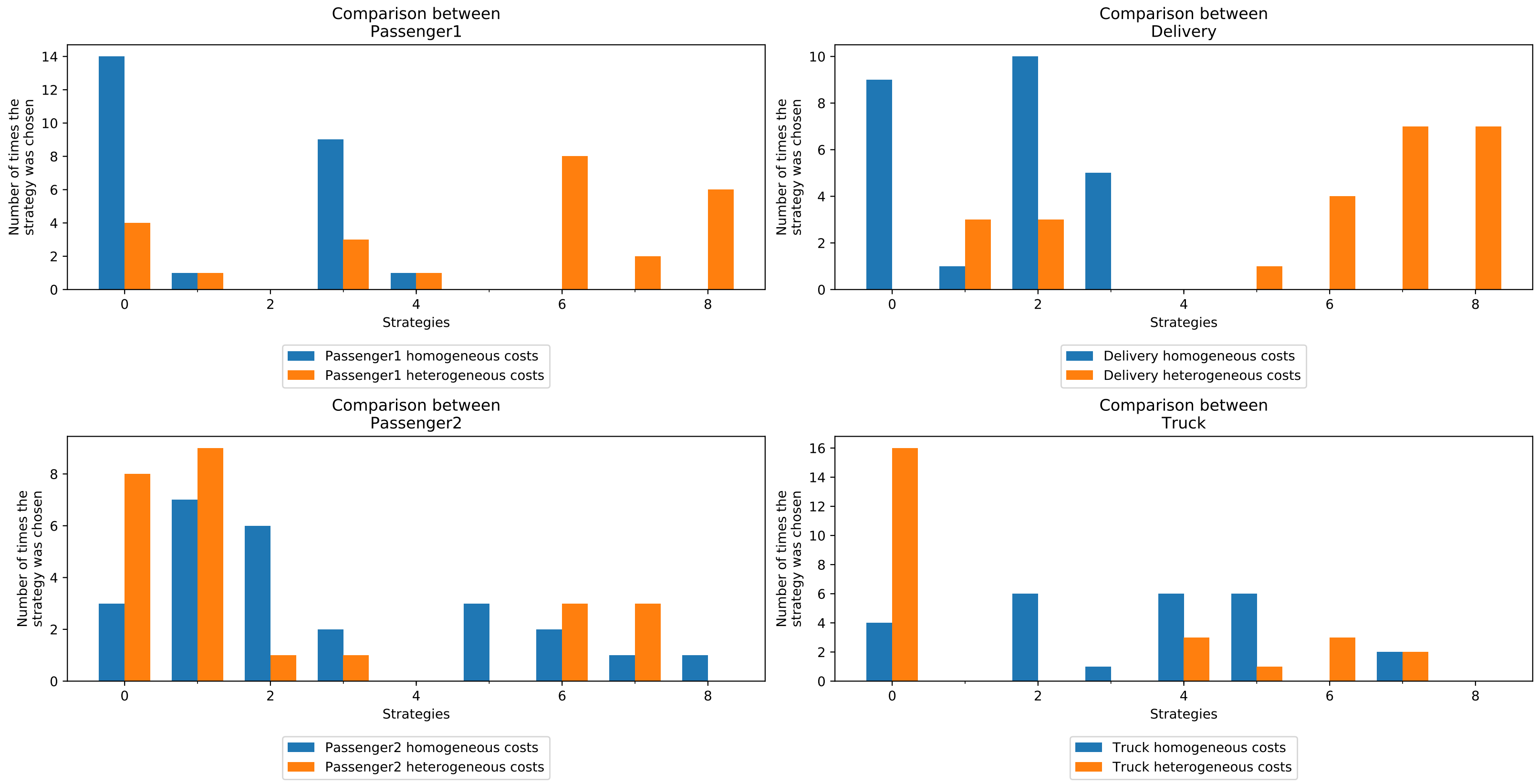
Source: Elaborated by the author.

Figure 31 presents the comparison of strategies selected by the vehicles. The blue bars represent the experiment where the vehicles have the same private cost function and the orange bars, different private cost functions.

In this scenario, we see a significant change in all vehicle classes. Firstly, the Passenger1 class vehicles had a considerable decrease in lower cost strategies, on the other hand, the Truck class had a significant increase in the number of lower cost strategies. The Passenger1 class went from 14 to 4 lowest cost strategies, while the Truck class increased from 4 to 16 lowest cost strategies.

In the case of Passenger2, the class had a significant increase in lower cost strategies. In the case of the Delivery class, which has the second highest cost, it had a decline in relation to equal costs. This can be explained by the significant increase in cost improvement of Truck class vehicles to the detriment of Delivery vehicle class.

Figure 31 – Comparison of strategies chosen by CAVs with equal and different private costs for the Crossroad 2-lane scenario



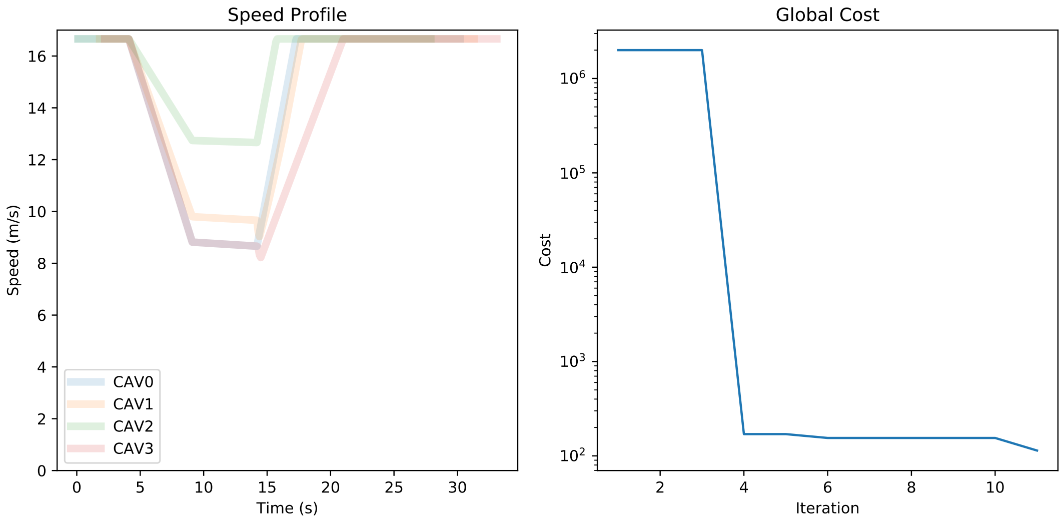
Source: Elaborated by the author.

5.2.3 Ramp Merge

In the Ramp Merge scenario, as described in Section 4.6.4, the Passenger1 and Passenger2 class vehicles were positioned on the upper road respectively in this order and the Truck and Delivery class vehicles were positioned on the lower road respectively in this order that is an on-ramp merge. The positioning of each vehicle was designed to force a conflict which is result in an imminent collision, for this each vehicle was placed on an arm of the crossing. At the beginning of the simulation, all vehicles start at 60 km/h, and after a predetermined fixed time, the vehicles began negotiating to resolve the conflict. Upon completion of the optimization process, the vehicles set the selected strategies and adjust their speed profiles to ensure a safe passage through the intersection.

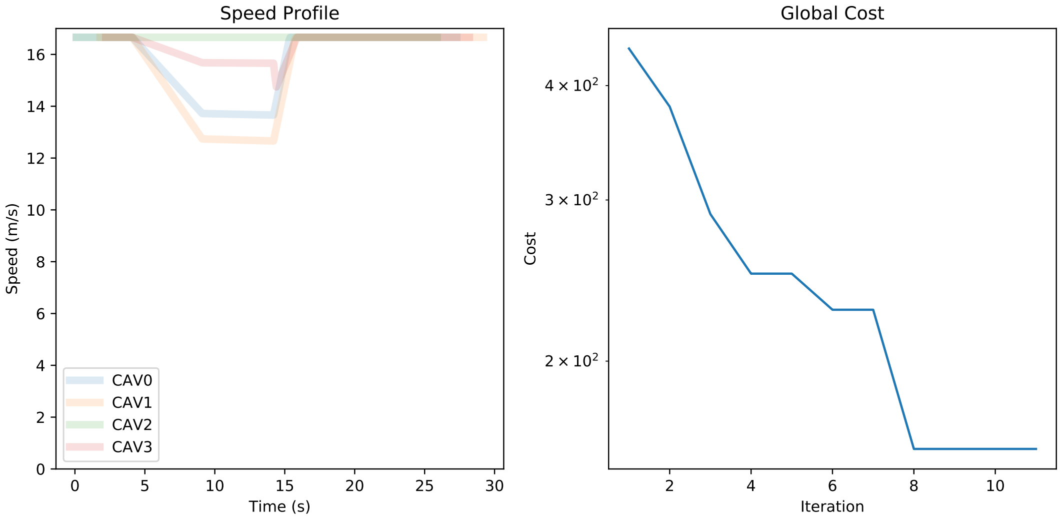
A sample of 25 experiments was carried out and it is presented in Figures 32 and 33, which shows the speed profile and the global cost of the agents. The negotiation began at 4s and after the optimization, the vehicles adjusted their speeds to safely navigate the intersection without collision, and subsequently returned to the road's speed limit. The cost was observed to decrease over the iterations, as the optimization process caused the agents to search for solutions that resolve the problem, although these solutions may not necessarily be the most optimal ones.

Figure 32 – Speed profile of CAVs with equal private cost functions and the evolution of the Global Cost over the iterations for the Ramp merge scenario



Source: Elaborated by the author.

Figure 33 – Speed profile of CAVs with different private cost functions and the evolution of the Global Cost over the iterations for the Ramp merge scenario



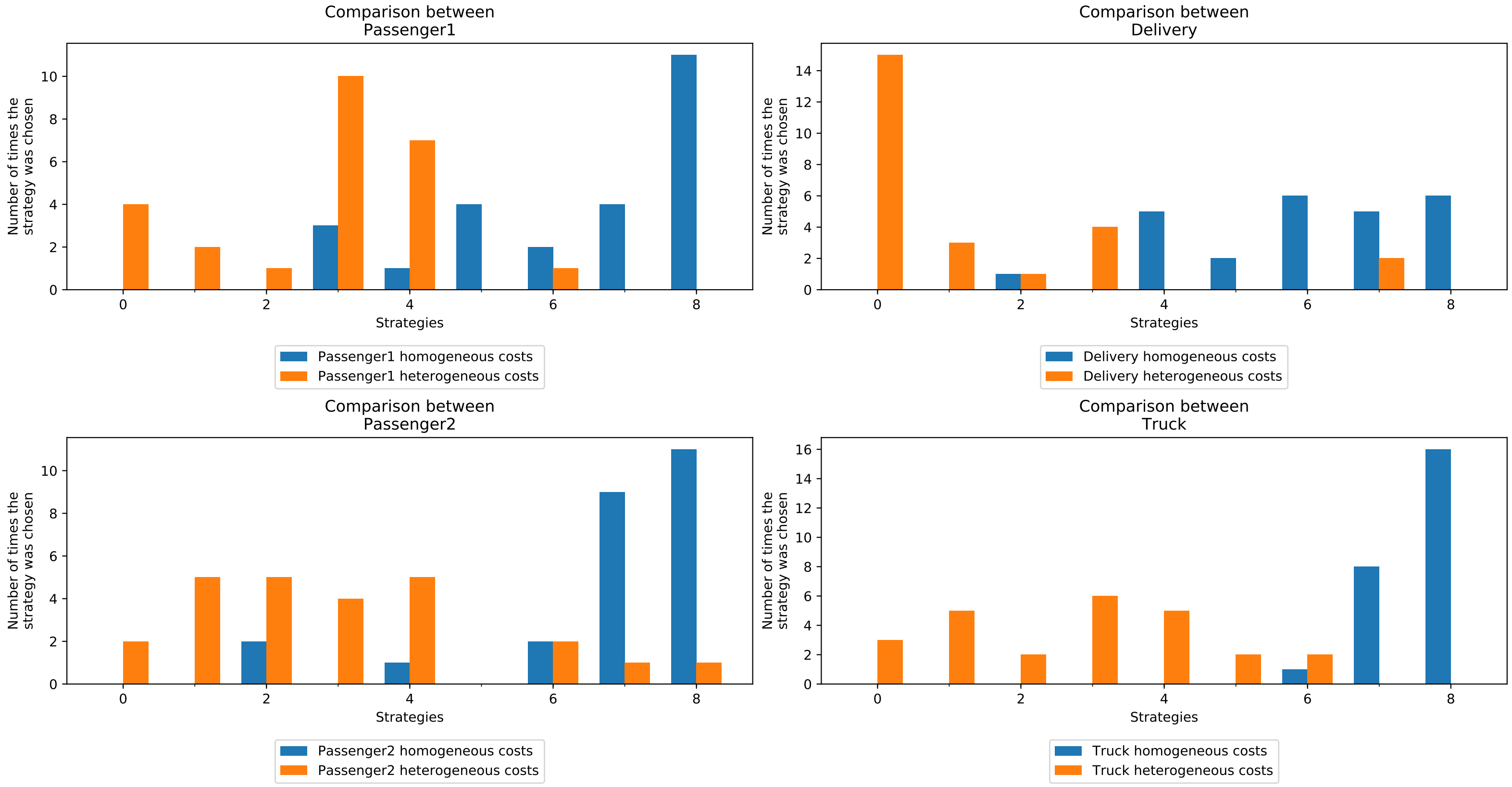
Source: Elaborated by the author.

In the comparison of the agents' strategy choices, presented in Figure 34, we take into consideration the different classes and private cost functions of the vehicles. The blue bars represent the experiment where the vehicles have the same private cost function and the orange bars, different private cost functions.

In this scenario, we can see a significant change in all vehicle classes. All classes had an improvement and were able to choose more lower cost strategies. The Delivery vehicle class achieved the best performance by choosing the best cost strategy 15 times. The Passenger2 and Truck classes had a more homogeneous improvement among the best cost strategies. The Passenger1 class vehicles also had a significant improvement when they were able to improve from the worst strategy to a more intermediate strategy.

It is interesting to note that in this case the vehicles are "side by side" on the roads, in which each road has only one lane. So in this arrangement the Delivery vehicle is next to Passenger2, despite the algorithm being run for all vehicles at the same time we can interpret it as a negotiation between these two vehicles which allowed the vehicle that has the most costly private cost function to choose lowest cost strategies.

Figure 34 – Comparison of strategies chosen by CAVs with equal and different private costs for the Ramp Merge scenario



Source: Elaborated by the author.

5.2.4 Lane Change

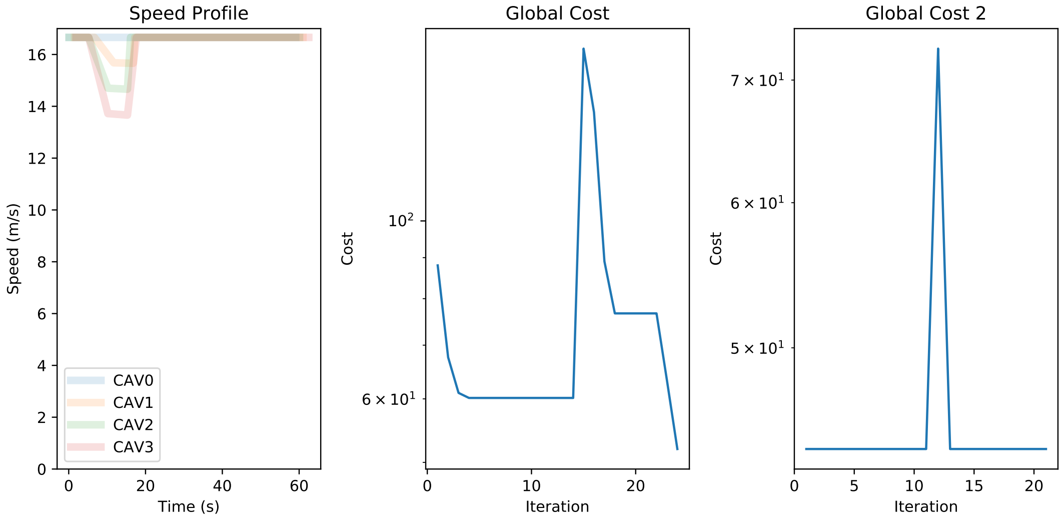
In the Lane Change scenario, as described in Section 4.6.4, four vehicles were positioned on the two lane road. The Truck and Delivery respectively in this order on the left lane and the Passenger1 and Passenger2 respectively in this order on the right lane. In this case, the experiment differs from the others because they randomly generate the intention to change lanes and thus a negotiation is called. All vehicles driving on this road within a radius of 50m are called for negotiation. In addition, vehicles that are already in the process of executing the negotiated maneuver within a 90m radius are also called for negotiation to provide their planning. Again, the positioning of each vehicle was designed to force a conflict which is result in an imminent collision. For this, we positioned the vehicles side by side. At the beginning of the simulation, all vehicles start at 60 km/h, and after a predetermined fixed time, the vehicles began to resolve the conflict. Upon completion of the optimization process, the vehicles set the selected strategies and adjust their speed profiles to ensure a safe passage through the intersection.

A sample of 25 experiments was carried out and is presented in Figures 35 and 36, which shows the speed profile and the global cost of the agents. The negotiation began randomly when vehicles generate intent to change lanes and after the optimization, the vehicles adjusted their speeds to safely execute the lane change maneuver, and subsequently returned to the road's speed limit. In this case, we present a third graph that also represents the cost, because as the intention to change lane is randomly generated, it may be that one of the vehicles does not participate in the negotiation because the intention to change lane has not yet been generated. In this way, when the intent is generated by the last vehicle, it initiates a negotiation call, but the other vehicles cannot change their running strategies and therefore we see few cost changes. The spike generated in Global Cost 2 is the choice of a high-cost strategy that was later discarded with more iterations.

The cost was observed to decrease over the iterations, as the optimization process caused the agents to search for solutions that resolve the problem, although these solutions may not necessarily be the most optimal ones.

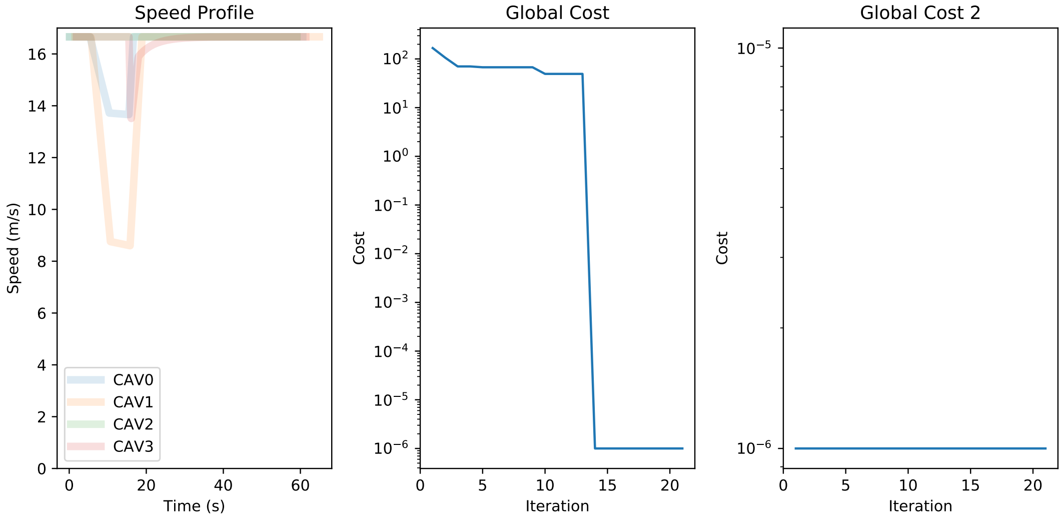
Figure 37 presents the comparison of strategies selected by the vehicles. Therefore, the blue bars represent the experiment where the vehicles have the same private cost function and the orange bars, different private cost functions. In this scenario, we see a significant improvement in the choice of strategies for vehicles of the Delivery and Truck classes. The Passenger1 class vehicle moved from low-cost strategies to mid-cost strategies. In the case of Passenger2, the class had a significant decrease in lower cost strategies.

Figure 35 – Speed profile of CAVs with equal private cost functions and the evolution of the Global Cost over the iterations for the Lane Change scenario



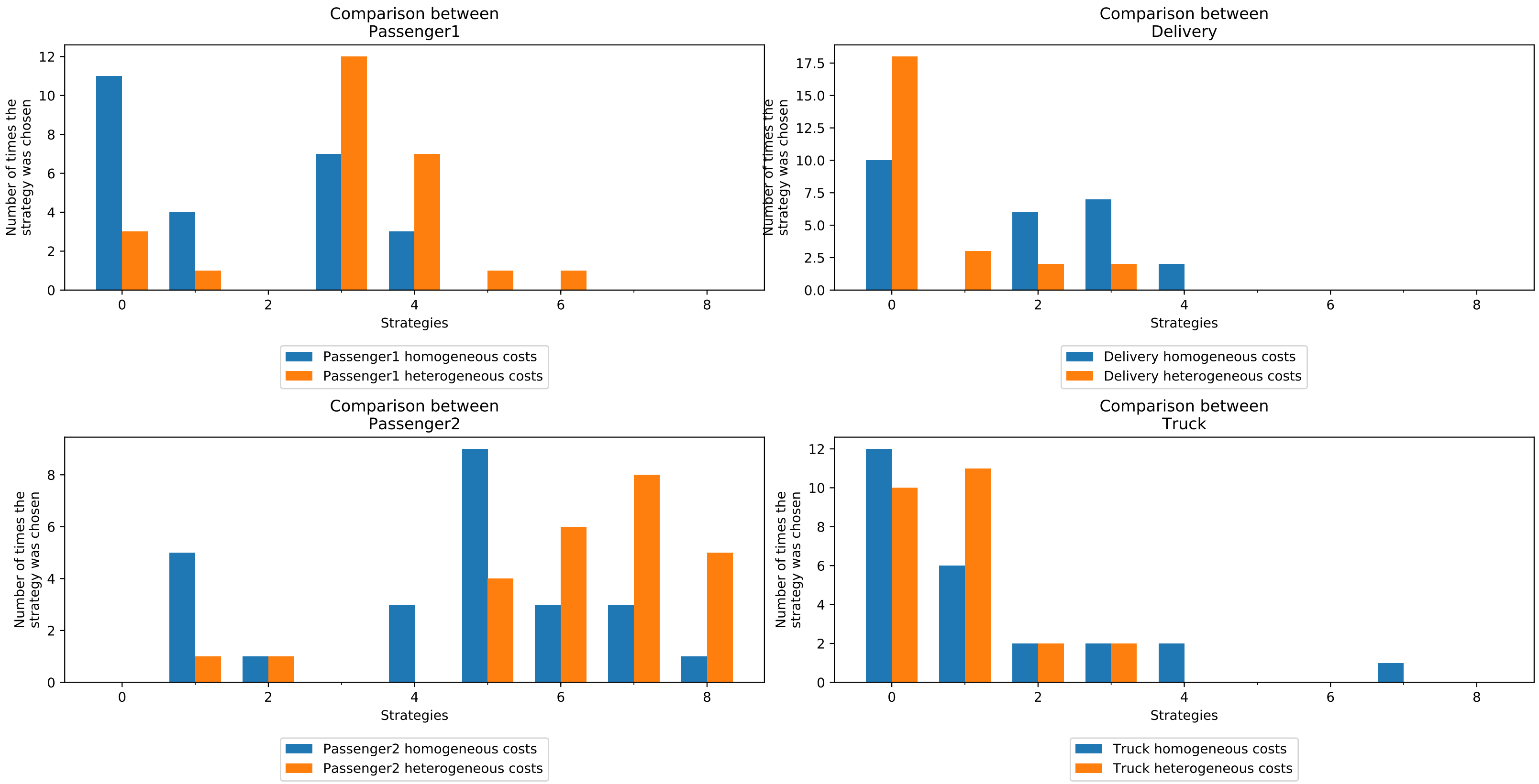
Source: Elaborated by the author.

Figure 36 – Speed profile of CAVs with different private cost functions and the evolution of the Global Cost over the iterations for the Lane Change scenario



Source: Elaborated by the author.

Figure 37 – Comparison of strategies chosen by CAVs with equal and different private costs for the Lane Change scenario



Source: Elaborated by the author.

5.3 Final Remarks

The results of our analysis show that the distribution of strategy choices among vehicles was not as uniform as expected when they had equal costs. This can be attributed to the fact that the vehicles have different dimensions, which affects their strategy selection. The optimization rebalanced the cost between agents with different private cost functions that needed to change the profile of choosing strategies because of costs.

For the scenarios presented, at first, a frequency of choice of lower cost strategies by the vehicles that have more expensive private cost functions was expected. This was confirmed for the case of the Truck class, which has the most costly private cost function and had an increase in the lowest cost strategies in all scenarios. In the case of the Delivery class vehicles, which have the second most expensive private cost function, the only significant increase was in the Ramp Merge and Lane Change scenario, in the Crossroad 2-lanes scenario it had a significant worsening and in the Crossroad 1- lane followed the similar pattern.

Vehicles with less costly private cost function, Passenger1 and Passenger2, had improvements only in the Ramp Merge scenario, in the rest they chose more costly strategies to the detriment of vehicles with more costly private cost function class or maintained the same profile.

CHAPTER 6

PROBABILITY COLLECTIVES IN DYNAMIC SCENARIOS

In the previous chapter, we conducted a study to evaluate the effectiveness of the Probability Collectives for CAVs model in solving a static problem through one-shot negotiation. In this chapter, we evaluate the method in a dynamic scenario, where there is a flow of vehicles and we also assumed that the vehicle arrivals are based on Poisson distribution.

To present the results of the simulations, we first provide a summary of the results in tables, detailing the variations of each proposed metric. Then, we present the results graphically to have a better understanding of the performance of the models over the simulation time. It is worth to mention that most of the results exhibit similar patterns and we will attempt to generalize the analyses to other experiments when possible.

We conducted experiments in four distinct scenarios: a single-lane crossroad, a two lane crossroad, a single-lane on-ramp merge, and a two lane road for lane change. For each scenario, we performed simulations with homogeneous CAVs, heterogeneous CAVs, heterogeneous CAVs with random speeds, and heterogeneous CAVs with different private cost functions and random speeds. Each experiment set was performed 25 times because of the randomness of the simulations.

Table 8 presents the distribution values used for each scenario. In each scenario, there are multiple routes, and each route is associated with a specific Poisson distribution. Additionally, Table 8 provides the overall distribution value used in the experiments.

Table 8 – Poisson distribution used in test scenarios

Scenario	Number of routes	Poisson distribution by route	Total Poisson Distribution
Crossroad 1-lane	12	0.04	0.48
Crossroad 2-lane	12	0.05	0.6
Ramp Merge	2	0.34	0.68
Lane Change	2	0.13	0.26

Source: Research data.

6.1 Crossroad 1-lane

The following Tables 9, 10, 11 and 12 summarize the results of the Crossroad 1-lane scenario and the four experiment variations: homogeneous CAVs, heterogeneous CAVs, heterogeneous CAVs with random speeds, and heterogeneous CAVs with different private cost functions and random speeds. The tables present the average and standard deviation of the proposed metrics based on the results of 25 runs for each model. This comparison allows us to observe the impact of adding heterogeneity and randomness to the system, within the same experimental variation and across different experiments.

If we now turn to the Table 9, the first experiment variation of the Crossroad 1-lane scenario, we can see the comparison of the four models. The SUMO model has the highest average travel time and lowest average speed because vehicles have to wait for the right of way at the intersection, on the other hand, the TLS model improves the previous one as it has the right of way guaranteed by the traffic lights. In any case, both models have to deal with the traffic priorities by stopping some vehicles which increases the waiting time. Our PC-based model performs better overall. It maintained an average speed close to the road speed limit and consequently had the lowest average travel time compared to the other models. With a higher average speed compared to the other models, it was possible to increase the number of vehicles and traffic flow.

We can do the same analysis as before for each Crossroad 1-lane scenario presented in the Tables 10, 11 and 12. Likewise, our model keeps the speed close to the road limit and with a low standard deviation, even considering more complex situations. That led to a decrease in average travel time by maintaining a continuous traffic flow.

In this Crossroad 1-lane scenario experiment, our model ensured the right of way by modifying the speed profile of vehicles and maintaining a fluid flow of traffic. This resulted in the highest vehicle throughput compared to the other models. Additionally, our model exhibited higher stability in terms of speed, as evidenced by its lower standard deviation, thus avoiding any oscillations in speed.

Table 9 – Comparison of models in the crossroad 1-lane homogeneous vehicles scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	42.08	11.58	24.50	5.42	8.50	1.68
Average Speed (m/s)	5.72	1.72	7.84	1.23	16.53	0.16
Average Waiting Time (s)	26.68	11.67	10.14	3.37	0.00	0.00
Arrived Vehicles	42.76	5.26	48.32	4.63	60.36	7.86
Flow (veh/h)	1282.80	157.86	1449.60	139.03	1810.80	235.72

Source: Research data.

Table 10 – Comparison of models in the crossroad 1-lane heterogeneous vehicles scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	30.32	10.54	22.24	5.78	8.30	1.07
Average Speed (m/s)	7.94	2.64	8.77	1.63	16.55	0.14
Average Waiting Time (s)	15.21	9.62	9.28	3.73	0.00	0.00
Arrived Vehicles	35.36	3.72	37.00	4.71	47.44	5.72
Flow (veh/h)	1060.80	111.54	1110.00	141.24	1423.20	171.70

Source: Research data.

Table 11 – Comparison of models in the crossroad 1-lane heterogeneous vehicles with random speeds scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	28.78	12.45	29.18	9.90	9.10	1.83
Average Speed (m/s)	8.82	3.03	7.35	2.26	16.54	0.26
Average Waiting Time (s)	14.84	11.21	13.03	6.79	0.00	0.00
Arrived Vehicles	38.24	4.14	45.64	5.09	49.16	4.87
Flow (veh/h)	1147.20	124.08	1369.20	152.70	1474.80	146.12

Source: Research data.

Table 12 – Comparison of models in the crossroad 1-lane heterogeneous vehicles with random speeds and different private cost functions scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Algorithm	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	30.18	12.46	20.63	6.18	9.27	2.16
Average Speed (m/s)	8.63	3.16	9.54	1.97	16.53	0.22
Average Waiting Time (s)	16.40	10.79	8.66	3.66	0.00	0.00
Arrived Vehicles	36.84	4.53	39.52	5.06	49.60	7.29
Flow (veh/h)	1105.20	136.02	1185.60	151.77	1488.00	218.57

Source: Research data.

Comparing Tables 9, 10, 11 and 12, i.e., between the variations of experiments within the crossroad 1-lane scenario, we can verify the decrease in flow for all models. That is due to the increase in complexity caused by the addition of heterogeneous vehicles and random speeds. As we add more complexity to the system we see a decrease in flow and an increase in travel time and average speed. However, our model was also able to guarantee the right of way and keep the speed average close to the road limit.

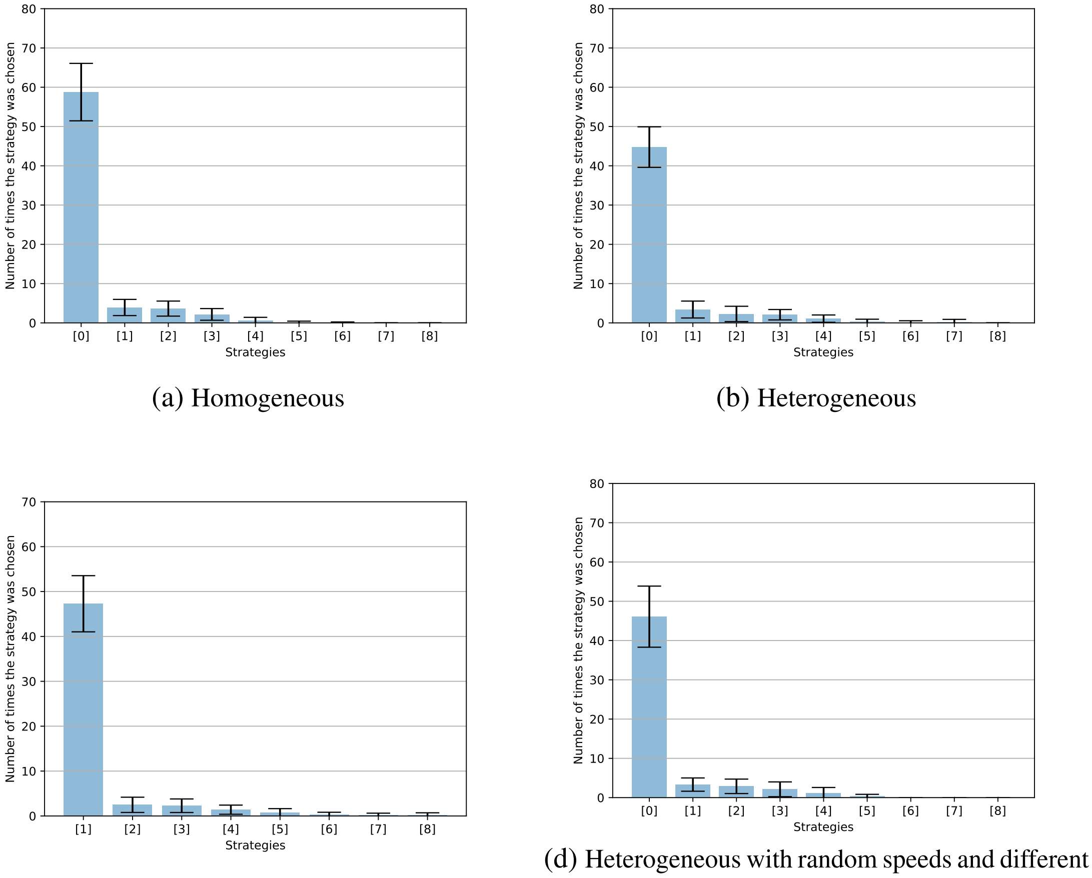
As we can see in the previous tables, our model always keeps the average speed close to the road limit the reason for this is that during optimization the agents mostly found strategies that didn't need to change speed, because no collisions were found and so they were left with the lowest cost action. Figure 38 shows the summary statistics for the strategies chosen during the 25 simulation runs for each scenario variation. The blue bars represent the average number of times a strategy was chosen and the inner bars are the standard deviation. For all scenario variations, agents mostly chose lower-cost actions.

The summary statistics for the strategies chosen during 25 simulation runs is displayed in Figure 38. The blue bars indicate the average frequency of strategy selection, while the inner bars show the standard deviation. It can be observed that most of the selected strategies were those with the lowest cost. The randomness of vehicle arrivals allows the choice of better strategies because most situations do not represent a worst case.

The graphs in Figures 39, 40, 41 and 42 present the results for average speed, average waiting time, and the number of vehicles over the experiment for the Crossroad 1-lane experiments and models. In this instance, we examine more closely the experiments conducted over time to provide a more comprehensive analysis of the results presented in the previous tables.

First, as we can notice, there are several colored lines and a highlighted bold black line. The colored lines of each graph in the background represent one specific run, and the black lines

Figure 38 – Number of times that one strategy was chosen



(a) Homogeneous

(b) Heterogeneous

(c) Heterogeneous with random speeds

(d) Heterogeneous with random speeds and different private cost functions

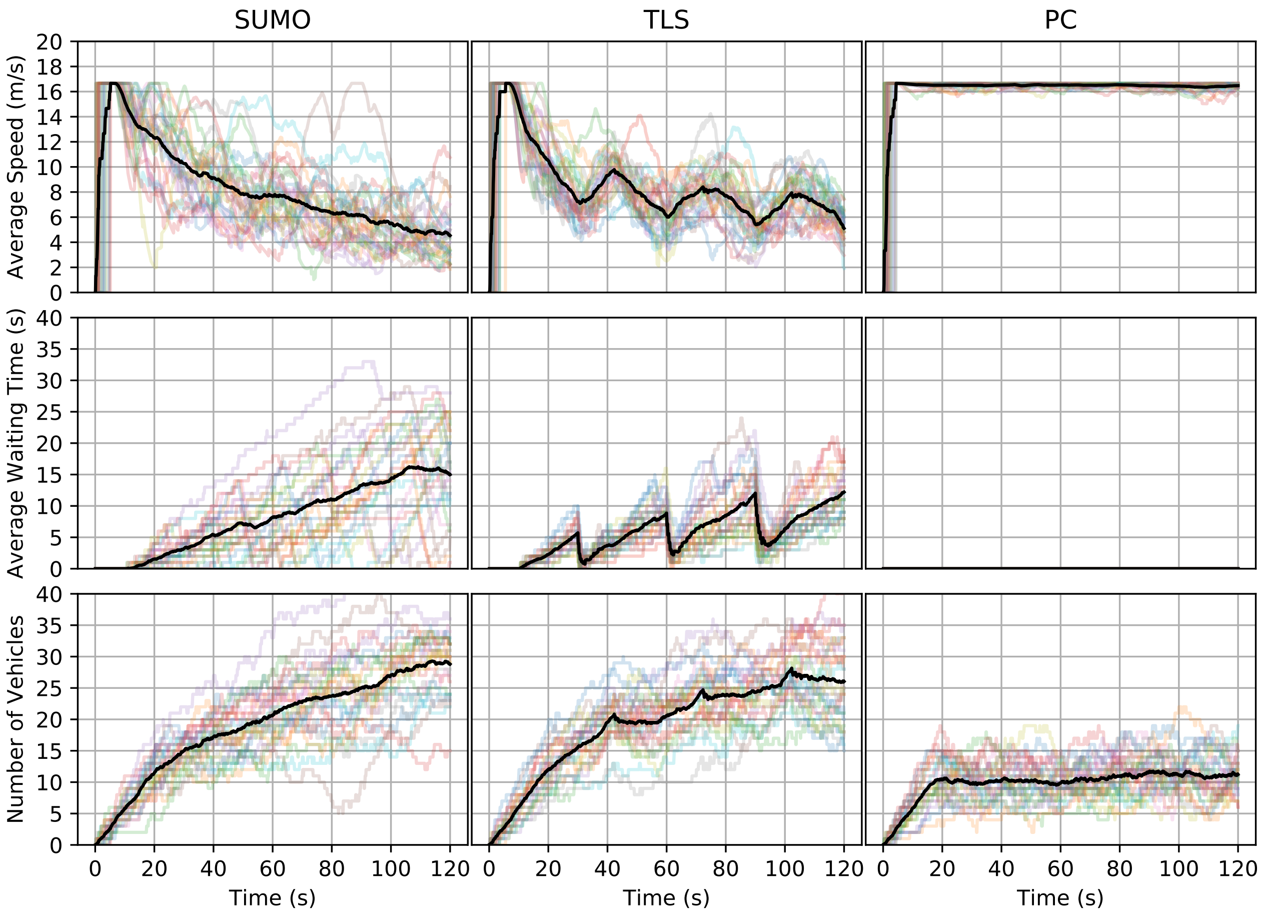
Source: Elaborated by the author.

represent the mean of all runs.

Although each experiment has its particularities, it is possible to notice the same pattern throughout the four experiments, homogeneous, heterogeneous, heterogeneous with random speeds, and heterogeneous with different private cost functions and random speeds for each model.

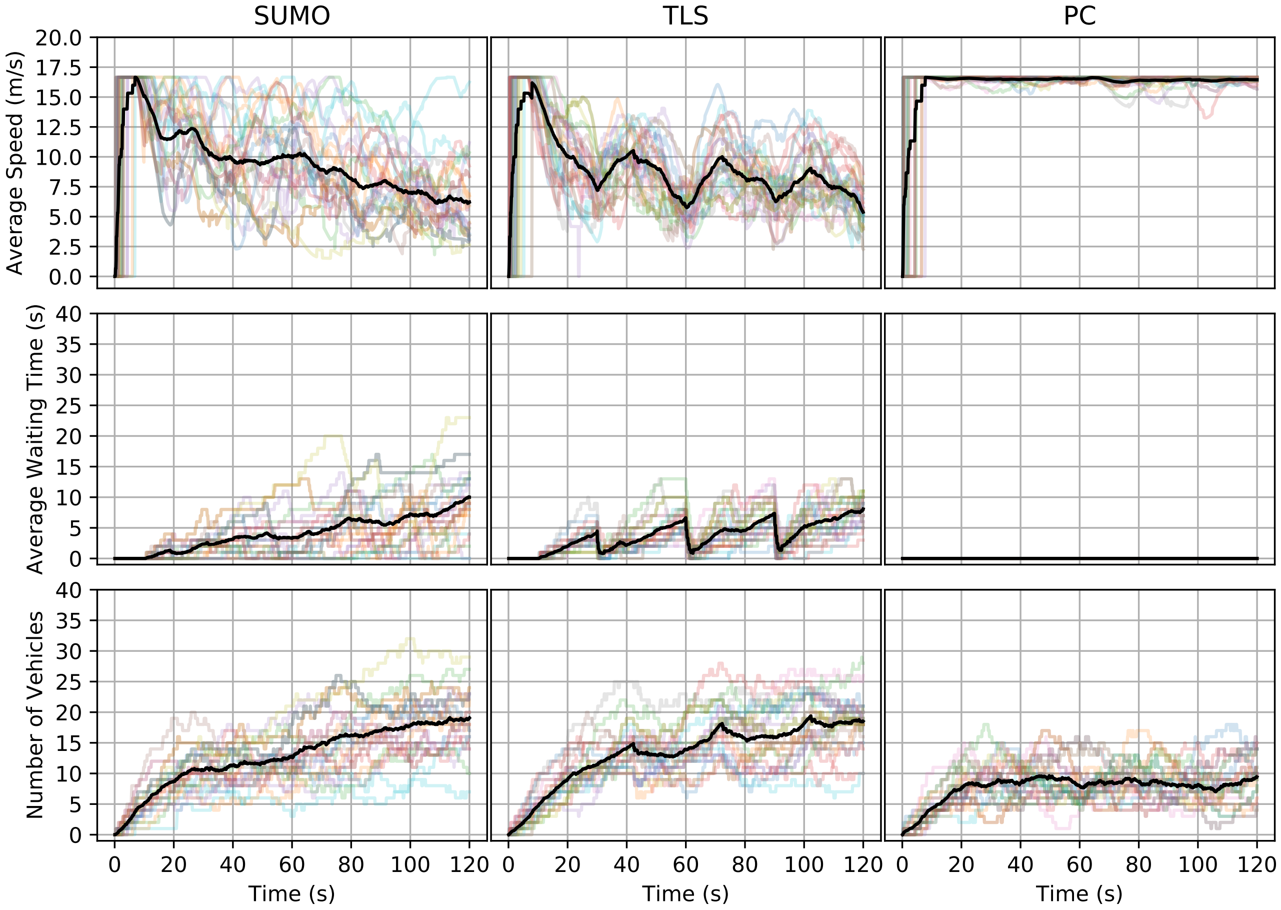
With regards to our PC-based model, there is no waiting time, as seen in the previous tables. This is achieved by adjusting the speed profile of the vehicles, which guarantees the right of way. As a result, both the average speed and the number of vehicles remain stable over time.

Figure 39 – Comparison of models in the crossroad 1-lane homogeneous vehicles scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



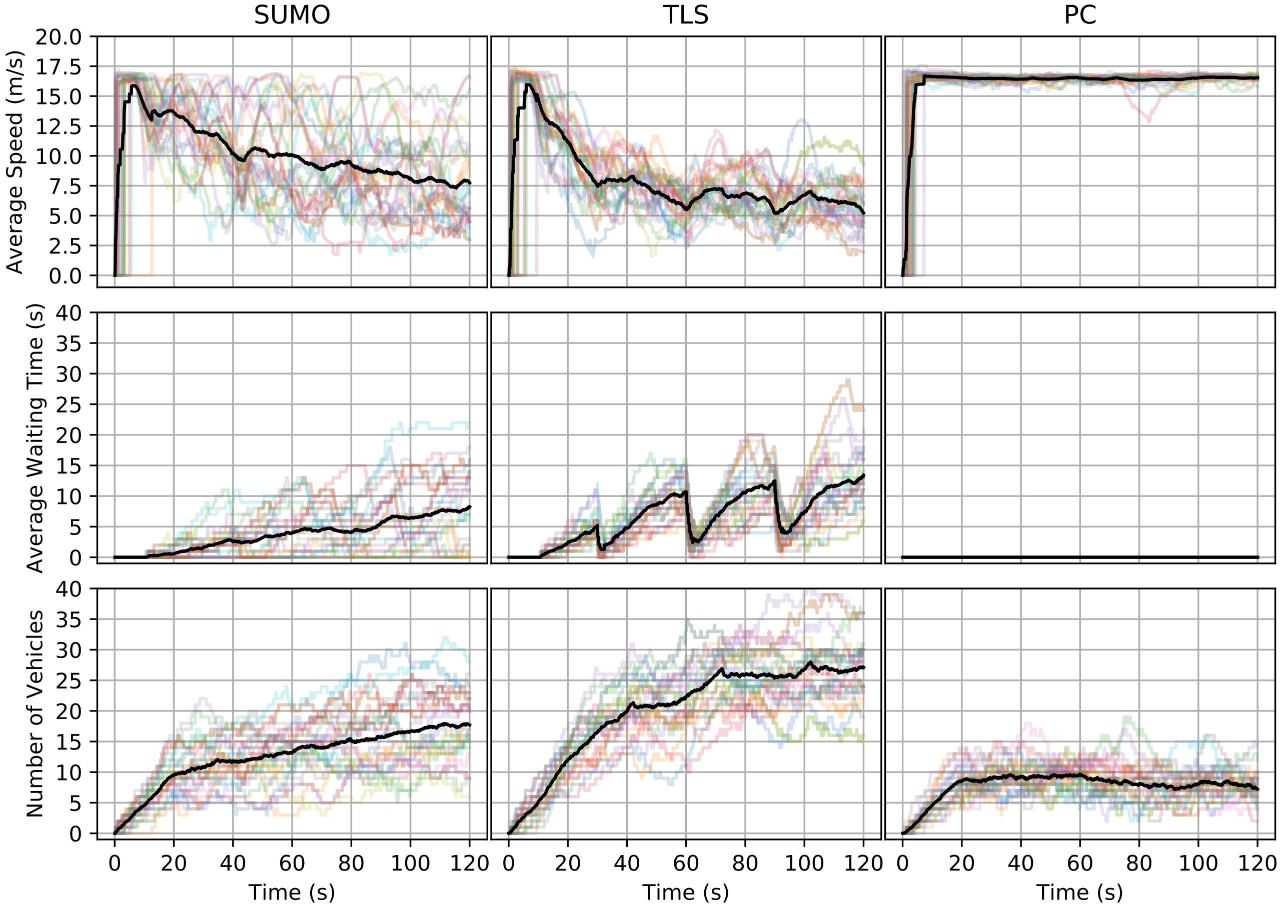
Source: Elaborated by the author.

Figure 40 – Comparison of models in the crossroad 1-lane heterogeneous vehicles scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



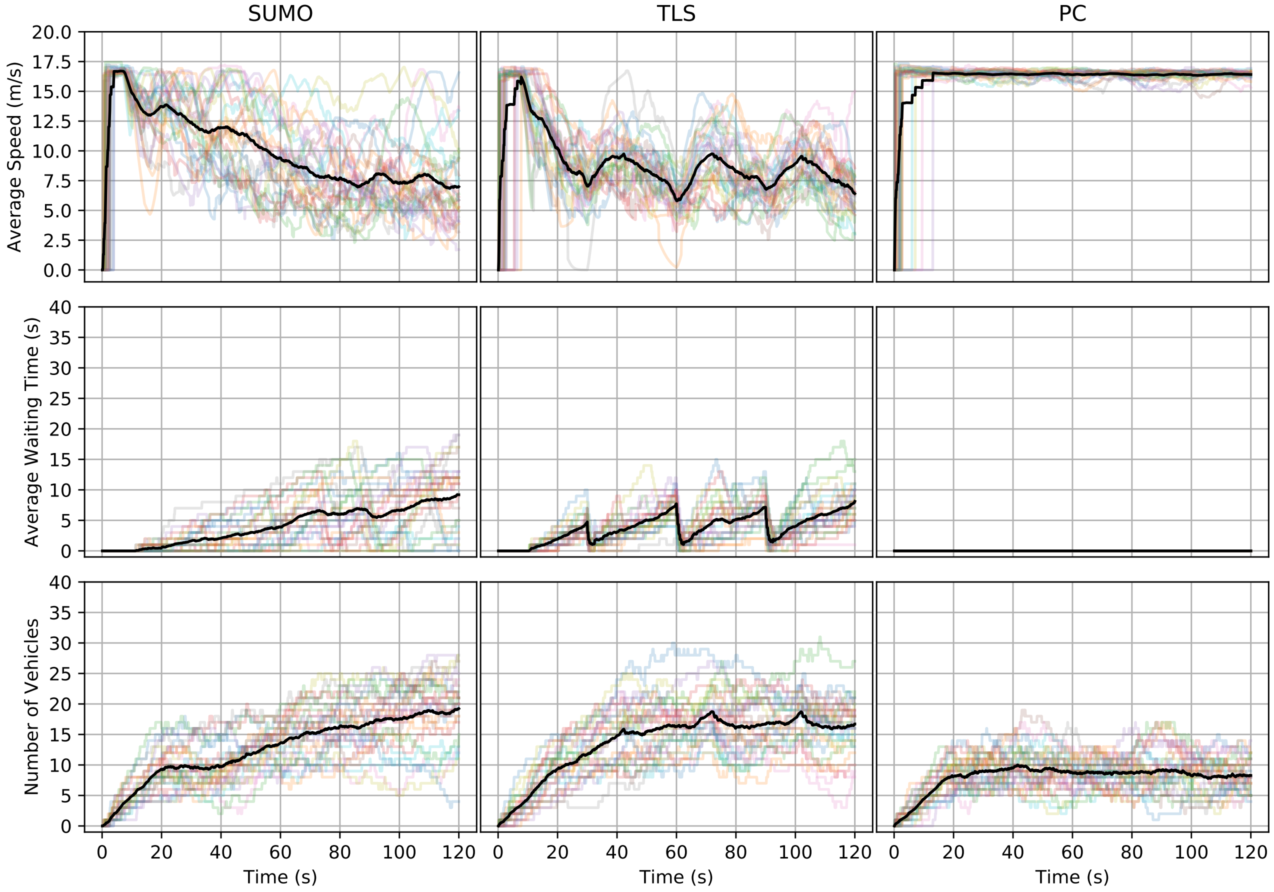
Source: Elaborated by the author.

Figure 41 – Comparison of models in the crossroad 1-lane heterogeneous vehicles with random speeds scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



Source: Elaborated by the author.

Figure 42 – Comparison of models in the crossroad 1-lane heterogeneous vehicles with random speeds and different private cost functions scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



Source: Elaborated by the author.

6.2 Crossroad 2-lane

In this scenario, the following Tables 13, 14, 15 and 16 summarize the results of the Crossroad 2-lane scenario and the four variations with homogeneous, heterogeneous, heterogeneous with random speeds, and heterogeneous with different private cost functions and random speeds. These tables allow us to compare the models within each experiment variation and also across experiments, taking into account the increasing complexity introduced by heterogeneous with different private cost functions and random speeds. For each model, the tables show the average and standard deviation of the proposed metrics, based on 25 runs.

If we compare the four models using the results from the Tables 13, 14, 15 and 16, as we saw in the previous scenario, our PC-based model performs better on the average speed close to the road speed limit and consequently has the lowest average travel time when compared to the other models. The SUMO model has the highest average travel time because vehicles have to wait for the right of way at the intersection; on the other hand, the TLS model improves the previous one as it has the right of way guaranteed by the traffic lights. In any case, both models have to deal with the right of way by stopping some vehicles and increasing the waiting time. Our model guarantees the right of way by dynamically adjusting the speed profile of vehicles. This optimization enables a higher flow of vehicles at the intersection compared alternative models. Additionally, our model shows better speed stability indicated by a lower standard deviation, effectively reducing speed oscillations and enhancing overall traffic stability.

If we make a comparison between the Tables 13, 14, 15 and 16, i.e., between the variations of experiments within the crossroad 2-lane scenario, we can verify the decrease in flow for all models. That is due to the increase in complexity caused by the addition of heterogeneous vehicles and random speeds. Our model was also able to guarantee the right of way and keep the speed average close to the road limit. But if one compares it with the previous experiments, we could notice that traffic flow is higher than in the Crossroad 1-lane scenario. That is due to the increase of a lane which allows a greater flow.

Table 13 – Comparison of models in the crossroad 2-lane homogeneous vehicles scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	26.58	10.41	19.12	4.25	9.52	1.71
Average Speed (m/s)	9.58	2.66	9.98	1.62	16.53	0.18
Average Waiting Time (s)	14.07	8.84	7.66	2.78	0.00	0.00
Arrived Vehicles	52.84	6.00	57.20	6.02	68.56	7.83
Flow (veh/h)	1585.20	180.14	1716.00	180.62	2056.80	234.96

Source: Research data.

Table 14 – Comparison of models in the crossroad 2-lane homogeneous vehicles scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	26.19	11.03	17.53	3.66	9.48	1.85
Average Speed (m/s)	9.57	2.74	10.45	1.46	16.53	0.20
Average Waiting Time (s)	13.70	9.88	7.24	2.67	0.00	0.00
Arrived Vehicles	50.12	5.03	50.36	7.49	59.72	6.34
Flow (veh/h)	1503.60	150.83	1510.80	224.81	1791.60	190.23

Source: Research data.

Table 15 – Comparison of models in the crossroad 2-lane heterogeneous vehicles with random speeds scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	25.61	9.48	20.58	4.34	9.11	1.81
Average Speed (m/s)	9.80	2.64	9.56	1.79	16.50	0.21
Average Waiting Time (s)	13.01	7.95	8.47	2.95	0.00	0.00
Arrived Vehicles	50.08	5.57	52.16	6.92	62.00	7.22
Flow (veh/h)	1502.40	167.01	1564.80	207.61	1860.00	216.51

Source: Research data.

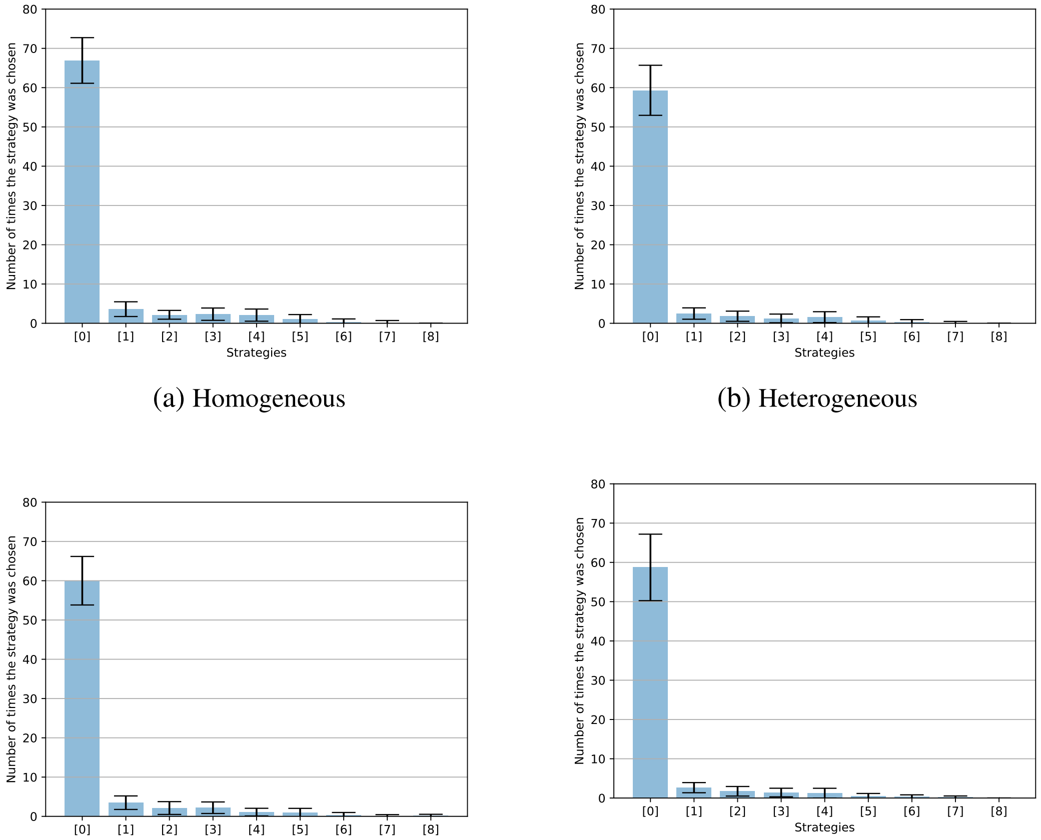
Table 16 – Comparison of models in the crossroad 2-lane heterogeneous vehicles with random speeds and different private cost functions scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Algorithm	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	23.90	9.49	18.66	3.74	9.04	1.69
Average Speed (m/s)	10.00	2.45	10.00	1.28	16.67	0.23
Average Waiting Time (s)	11.89	8.66	7.38	2.29	0.00	0.00
Arrived Vehicles	52.12	7.21	51.12	6.10	58.84	9.39
Flow (veh/h)	1563.60	216.22	1533.60	182.96	1765.20	281.65

Source: Research data.

The summary statistics for the strategies chosen during 25 simulation runs is displayed in Figure 43. The blue bars indicate the average frequency of strategy selection, while the inner bars show the standard deviation. It can be observed that most of the selected strategies were those with the lowest cost. The randomness of vehicle arrivals allows the choice of better strategies because most situations do not represent a worst case.

Figure 43 – Number of times that one strategy was chosen in the crossroad 2-lane scenario



(a) Homogeneous

(b) Heterogeneous

(c) Heterogeneous with random speeds

(d) Heterogeneous with random speeds and different private cost functions

Source: Elaborated by the author.

The graphs in the Figures 44, 45, 46 and 47 show the results for average speed, average waiting time, and the number of vehicles over the experiment for the Crossroad 2-lane experiments for all models. We analyze in depth the experiments conducted over time to provide a more comprehensive analysis of the results presented in the previous tables.

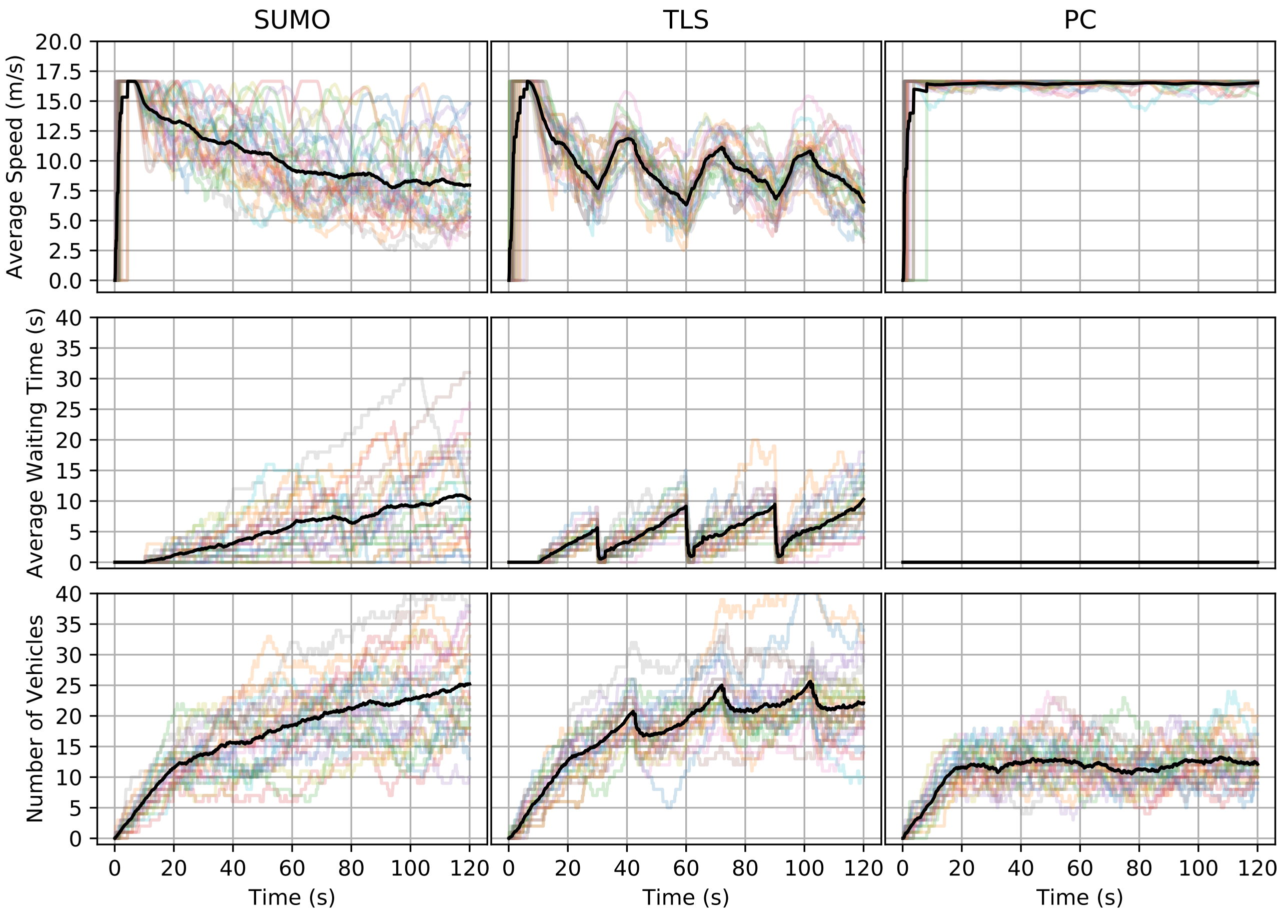
First, as we analyzed before, there are several colored lines and a bold black line in the graphs. The colored lines of each graph in the background represent one specific run, and the black lines represent the mean of all runs. Although each experiment has its particularities, it is possible to notice the same pattern throughout the four experiments, homogeneous, heterogeneous, heterogeneous with random speeds, and heterogeneous with different private cost functions and random speeds for each model.

Although the SUMO model does not guarantee the right of way at intersections, this time, there are some vehicles that could use one of the side lanes to make the turn and then the flow result is higher. However, we still notice traffic jams over time because of the vehicle's right of way. As vehicles arrive randomly, some need to stop at the intersection, and consequently the number of vehicles increases as well as the average waiting time and average speed decreases. We can observe this pattern in SUMO's graphical results.

As with the previous model, the TLS we can notice the same trend of decreasing speed and increasing the average waiting time, consequently increasing the number of vehicles. However, we can notice a wave pattern in the graphs where each phase corresponds to a cycle of the traffic light system. Although this model guarantees the right of way, the entry of new vehicles causes a decrease in the flow over time.

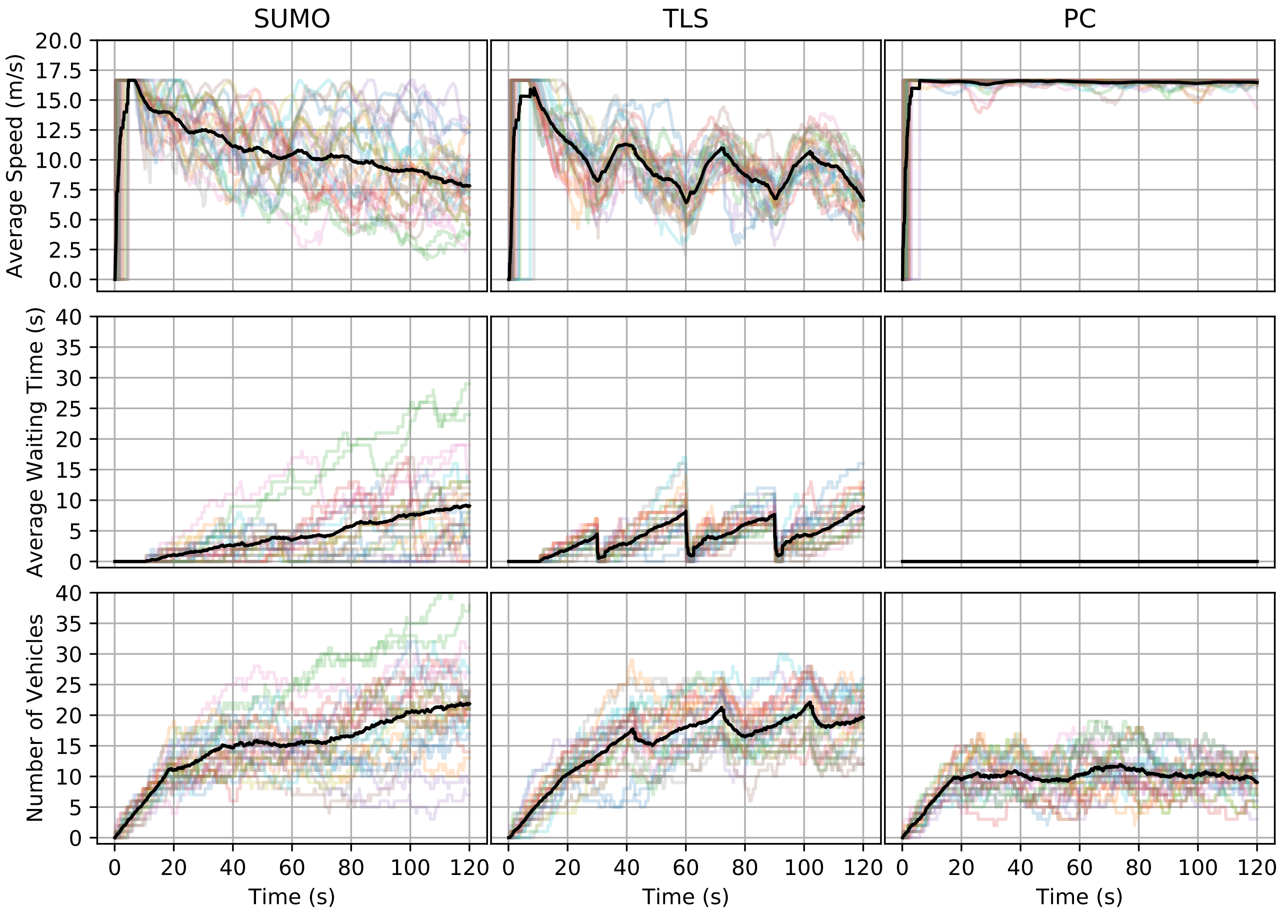
Concerning our PC-based model, there is no waiting time as presented in the previous tables. That is because the model guarantees the right of way by changing the speed profile of the vehicles. In this way, the average speed and the number of vehicles are remain stable over time.

Figure 44 – Comparison of models in the crossroad 2-lane homogeneous vehicles scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



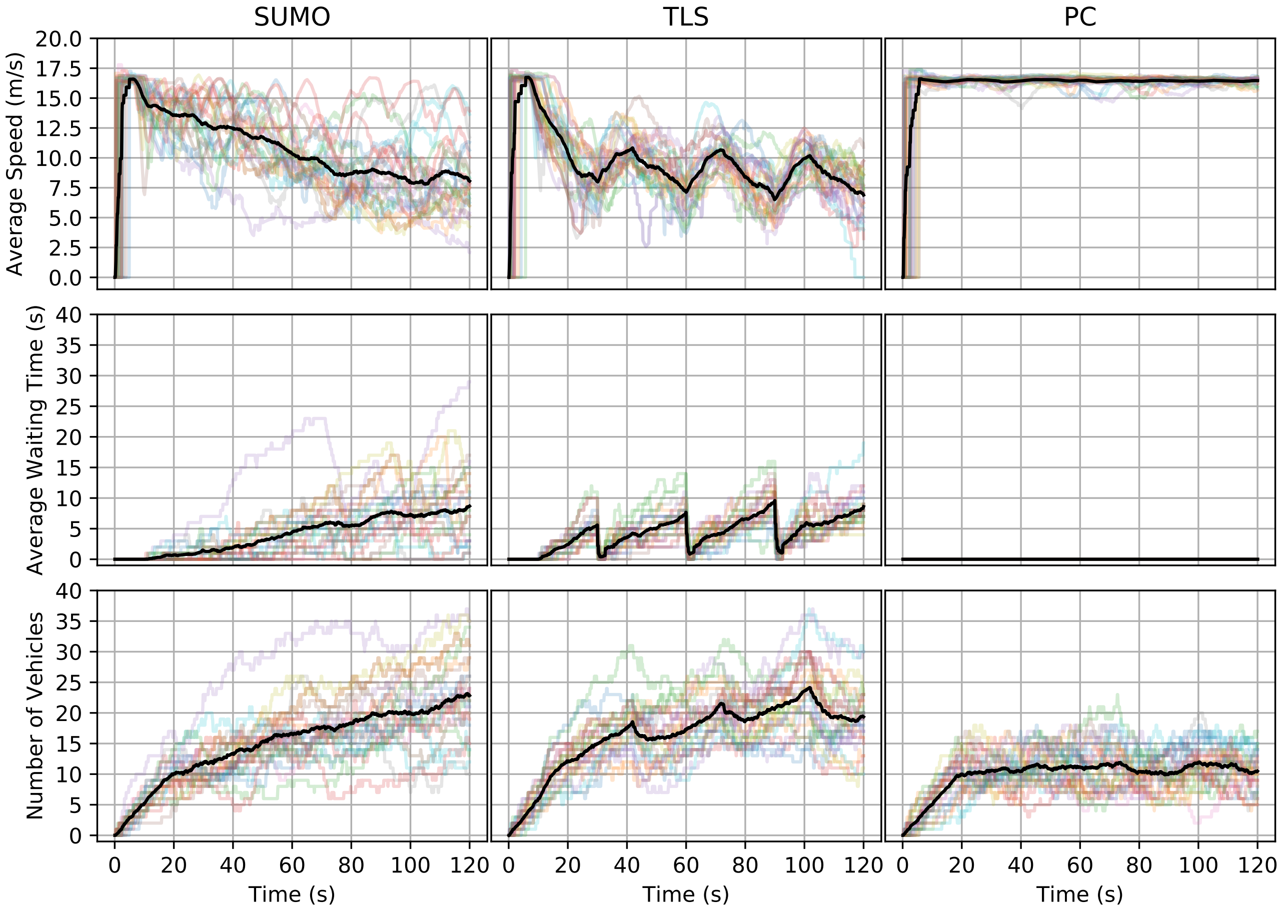
Source: Elaborated by the author.

Figure 45 – Comparison of models in the crossroad 2-lane heterogeneous vehicles scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



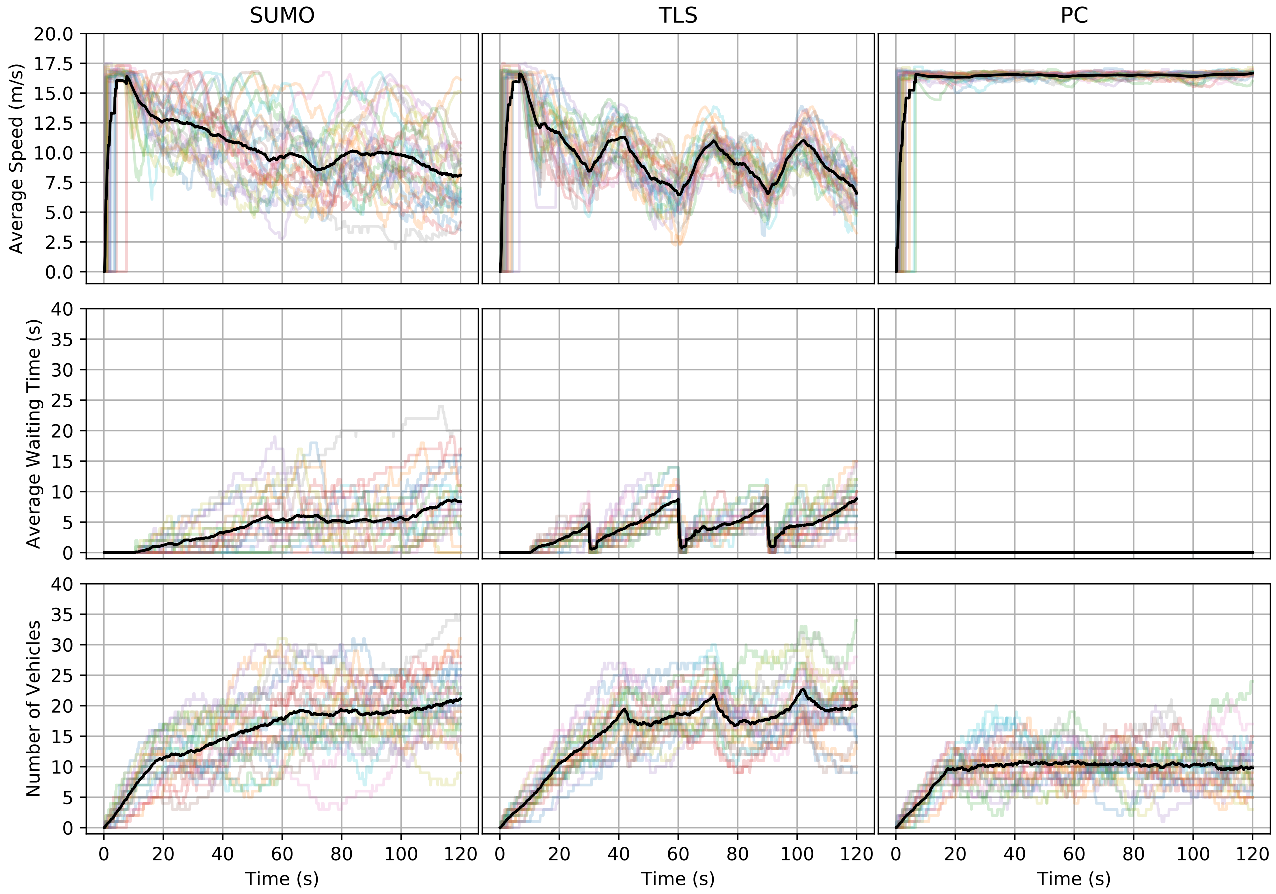
Source: Elaborated by the author.

Figure 46 – Comparison of models in the crossroad 2-lane heterogeneous vehicles with random speeds scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



Source: Elaborated by the author.

Figure 47 – Comparison of models in the crossroad 2-lane heterogeneous vehicles with random speeds and different private cost functions scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



Source: Elaborated by the author.

6.3 Ramp Merge

In this experiment, we have changed the type of scenario from crossroad to ramp merge. The following Tables 17, 18, 19 and 20 summarize the results of the ramp merge scenario and again we show the three variations, homogeneous vehicles, heterogeneous vehicles, and heterogeneous vehicles with random speeds. We can compare the models within the same experiment and also between experiments considering the increase in complexity, heterogeneous vehicles and random speeds. The tables show the average and standard deviation for the proposed metrics considering all 25 runs.

The Table 17 presents the first experiment variation of the Ramp Merge scenario. First, we can notice the same pattern of results as in the previous scenarios. As the SUMO model does not guarantee the right of way because some vehicles had to wait to pass through the junction, it presented the highest average travel time; on the other hand, the TLS model controls the passage of vehicles through the traffic light cycles and improves traffic flow. Our model, on the other hand, presents the shortest travel time and the highest average speed, which allowed for greater traffic flow.

We can do the same analysis as before for each crossroad 1-lane scenario presented in the Tables 10 and 11. Likewise, our model keeps the speed close to the road limit and with a low standard deviation, even considering more complex scenarios. That led to a decrease in average travel time by maintaining a continuous traffic flow.

Table 17 – Comparison of models in the ramp merge homogeneous vehicles scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Algorithm	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	55.59	7.00	28.27	3.76	14.88	1.30
Average Speed (m/s)	6.44	0.91	9.87	1.17	16.62	0.05
Average Waiting Time (s)	33.67	7.83	7.45	2.23	0.00	0.00
Arrived Vehicles	33.52	4.18	45.60	4.50	60.76	7.63
Flow (veh/h)	1005.60	125.53	1368.00	135.00	1822.80	229.01

Source: Research data.

Table 18 – Comparison of models in the ramp merge heterogeneous vehicles scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	58.80	9.53	28.68	3.00	14.72	1.96
Average Speed (m/s)	6.21	0.68	9.20	0.90	16.60	0.08
Average Waiting Time (s)	38.32	11.42	7.68	2.23	0.00	0.00
Arrived Vehicles	32.16	3.13	42.92	5.21	63.60	8.57
Flow (veh/h)	964.80	93.95	1287.60	156.35	1908.00	257.05

Source: Research data.

Table 19 – Comparison of models in the ramp merge heterogeneous vehicles with random speeds scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	57.93	7.89	29.69	3.69	14.78	2.11
Average Speed (m/s)	6.17	1.08	9.05	1.25	16.52	0.14
Average Waiting Time (s)	36.57	9.53	8.19	2.55	0.00	0.00
Arrived Vehicles	30.80	5.69	42.32	4.17	62.80	7.83
Flow (veh/h)	924.00	170.59	1269.60	125.12	1884.00	234.79

Source: Research data.

Table 20 – Comparison of models in the ramp merge heterogeneous vehicles with random speeds and different private cost functions scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Algorithm	**SUMO**		**TLS**		**PC**	
Statistical	AVG	STD	AVG	STD	AVG	STD
Average Travel Time (s)	56.43	8.45	28.82	3.36	14.82	1.83
Average Speed (m/s)	6.48	0.83	9.43	0.88	16.78	0.08
Average Waiting Time (s)	35.02	10.00	7.73	2.18	0.00	0.00
Arrived Vehicles	33.08	4.39	42.12	4.17	61.76	6.72
Flow (veh/h)	992.40	131.60	1263.60	125.00	1852.80	201.67

Source: Research data.

If we compare the three models using the results from the Tables 17, 18 and 19, as we saw in the previous scenario, our PC-based model performs better on the average speed close to the road speed limit and consequently has the lowest average travel time when compared to the other models. The SUMO model has the highest average travel time because vehicles have to wait for the right of way at the junction, on the other hand, the TLS model improves the previous one as it has the right of way guaranteed by the traffic lights. In any case, both models have to deal with the right of way by stopping some vehicles and increasing the waiting time.

In this ramp merge 1-lane scenario our model guaranteed the right of way because it changes the speed profile of the vehicles and tends to keep the traffic fluid. That allows more vehicles to pass through the intersection, thus, obtaining the highest flow compared to the other models. Our model also has higher stability in speed as it has a lower standard deviation, which avoids speed oscillations.

If we make a comparison between the Tables 17, 18, 19 and 20, i.e., between the variations of experiments within the Crossroad 2-lane scenario, we can verify the decrease in flow for all models. That is due to the increase in complexity caused by the addition of heterogeneous vehicles and random speeds. Our model was also able to guarantee the right of way and keep the speed average close to the road limit. But if one compares it with the previous experiments, we could notice that traffic flow is higher than in the crossroad 1-lane scenario. That is due to the increase of a lane which allows a greater flow.

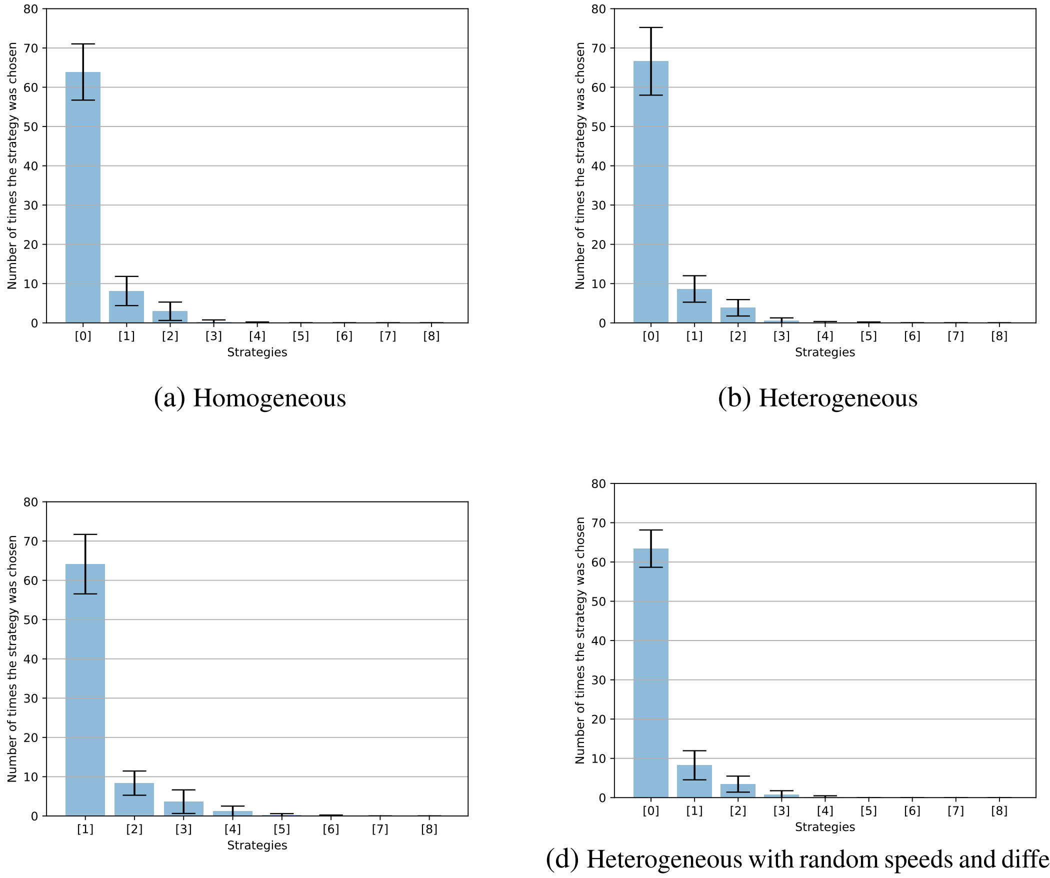
The summary statistics for the strategies chosen during 25 simulation runs is displayed in Figure 48. The blue bars indicate the average frequency of strategy selection, while the inner bars show the standard deviation. It can be observed that most of the selected strategies were those with the lowest cost. The randomness of vehicle arrivals allows the choice of better strategies because most situations do not represent a worst case.

The graphs in the Figures 49, 50, 51 and 52 show the results for average speed, average waiting time, and the number of vehicles over the experiment for the ramp merge 1-lane experiments for all models. In this instance, we present the experiments over time to provide a more comprehensive analysis of the results presented in the previous tables.

First, as we analyzed before, there are several colored lines and a bold black line in the graphs. The colored lines of each graph in the background represent one specific run, and the black lines represent the mean of all runs. Although each experiment has its particularities, it is possible to notice the same pattern throughout the three experiments, homogeneous, heterogeneous, and heterogeneous with random speeds for each model.

Although the SUMO model does not guarantee the right of way at intersections, this time, there are some vehicles that could use one of the side lanes to make the turn and then the flow result is higher. However, we still notice traffic jams over time because of the vehicle's right of way. As vehicles arrive randomly, some need to stop at the intersection, and consequently the

Figure 48 – Number of times that one strategy was chosen in the ramp merge scenario



(a) Homogeneous

(b) Heterogeneous

(c) Heterogeneous with random speeds

(d) Heterogeneous with random speeds and different private cost functions

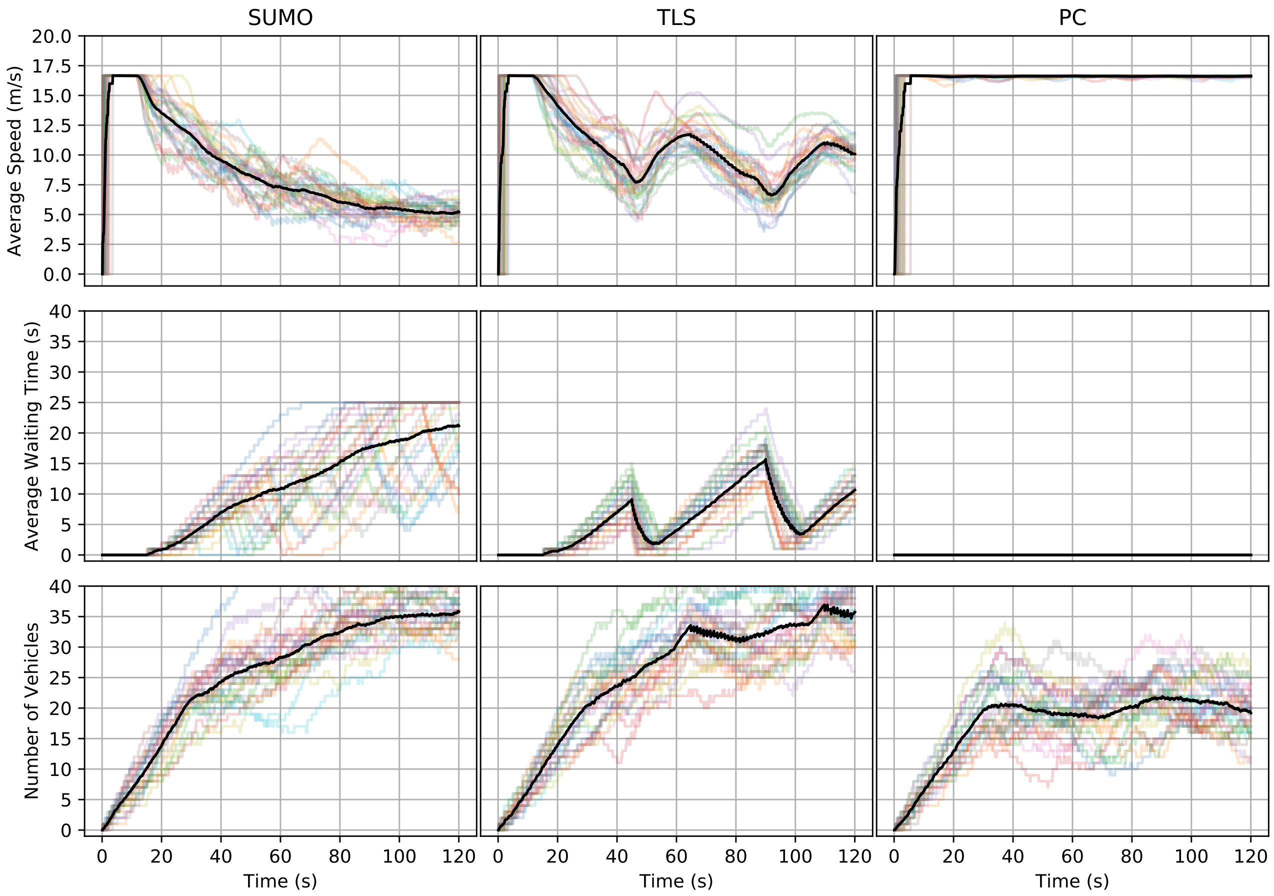
Source: Elaborated by the author.

number of vehicles increases as well as the average waiting time and average speed decreases. We can observe this pattern in SUMO's graphical results.

As with the previous model, the TLS we can notice the same trend of decreasing speed and increasing the average waiting time, consequently increasing the number of vehicles. However, we can notice a wave pattern in the graphs where each phase corresponds to a cycle of the traffic light system. Although this model guarantees the right of way, the entry of new vehicles causes a decrease in the flow over time.

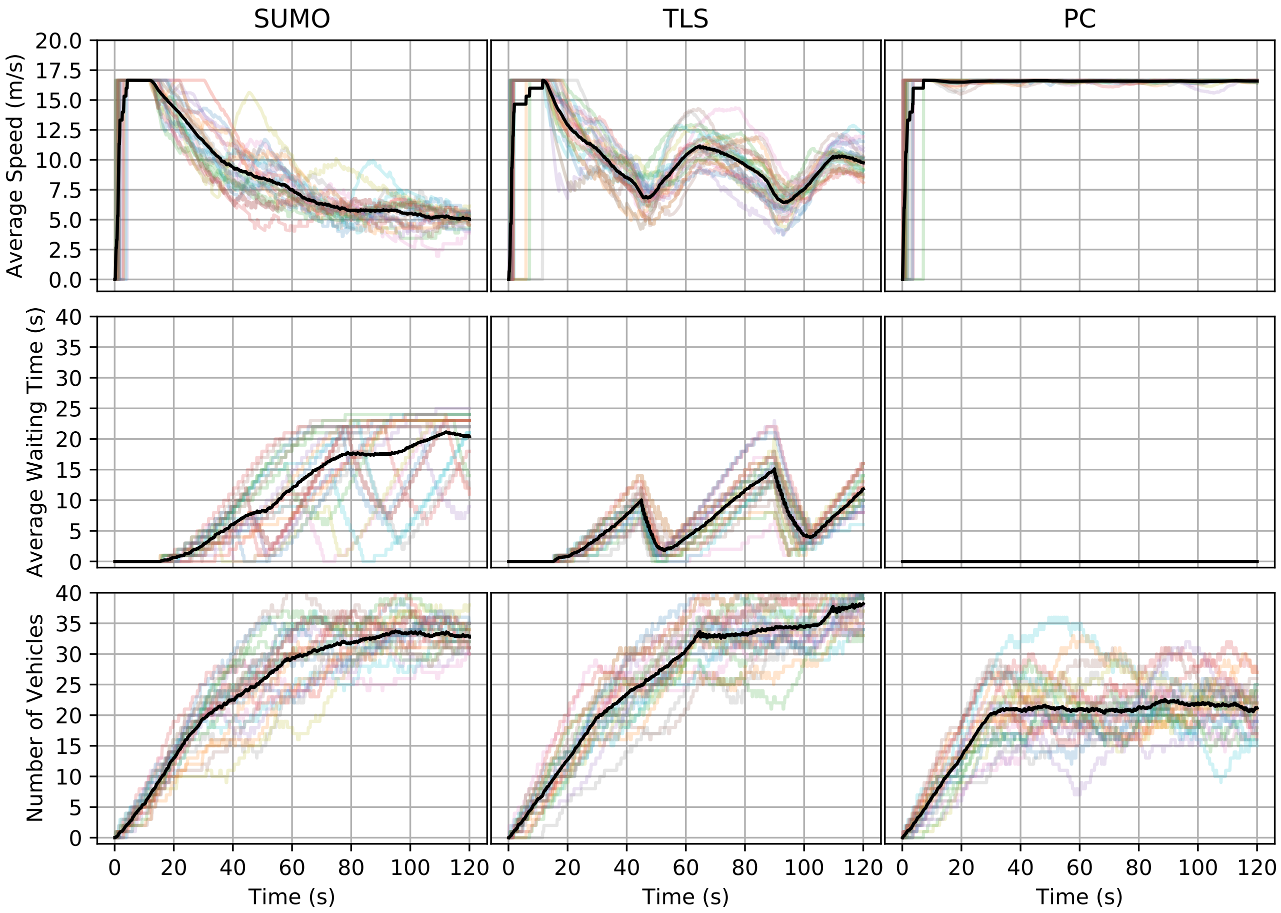
Concerning our PC-based model, there is no waiting time as presented in the previous tables. That is because the model guarantees the right of way by changing the speed profile of the vehicles. In this way, the average speed and the number of vehicles remain stable over time.

Figure 49 – Comparison of models in the ramp merge homogeneous vehicles scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



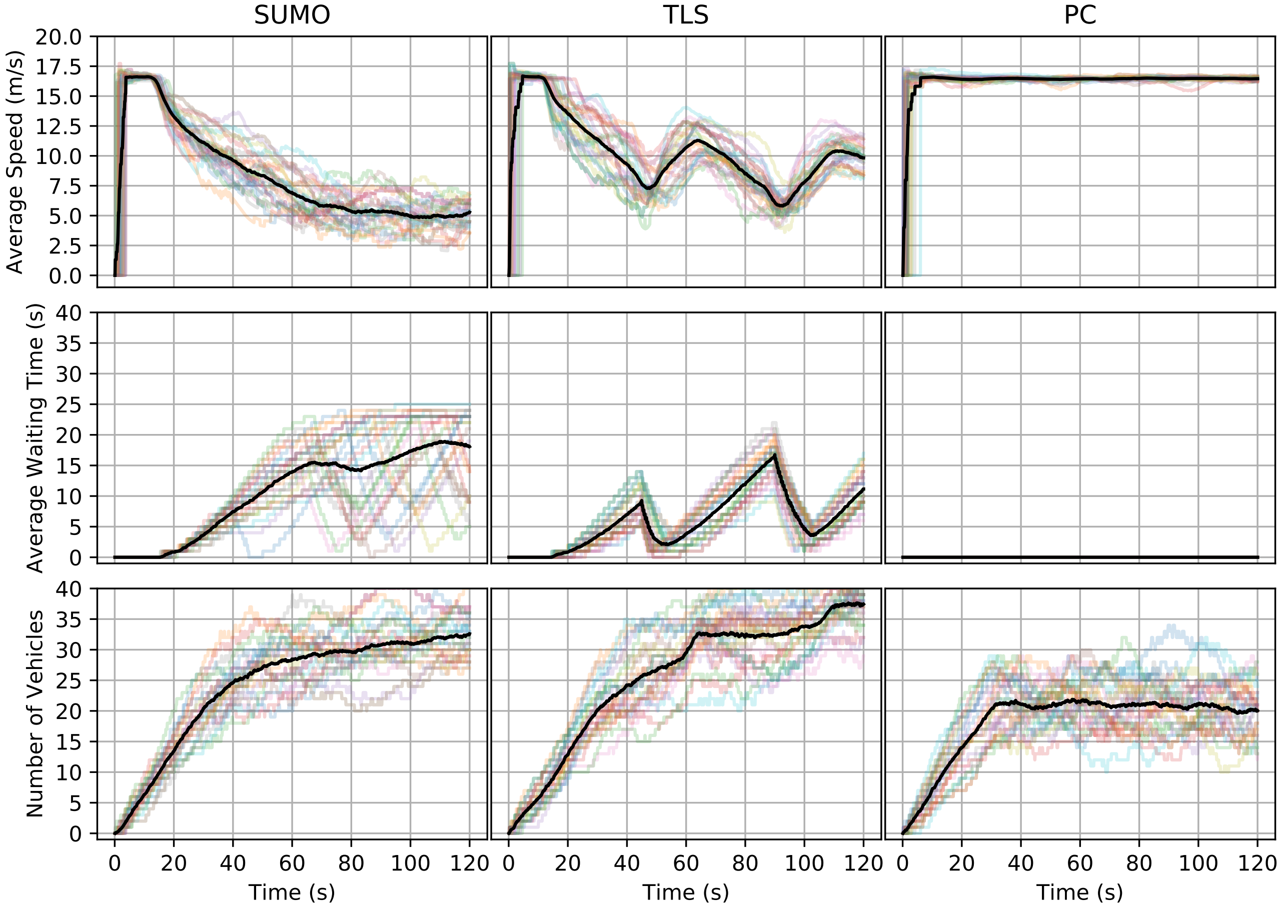
Source: Elaborated by the author.

Figure 50 – Comparison of models in the ramp merge heterogeneous vehicles scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



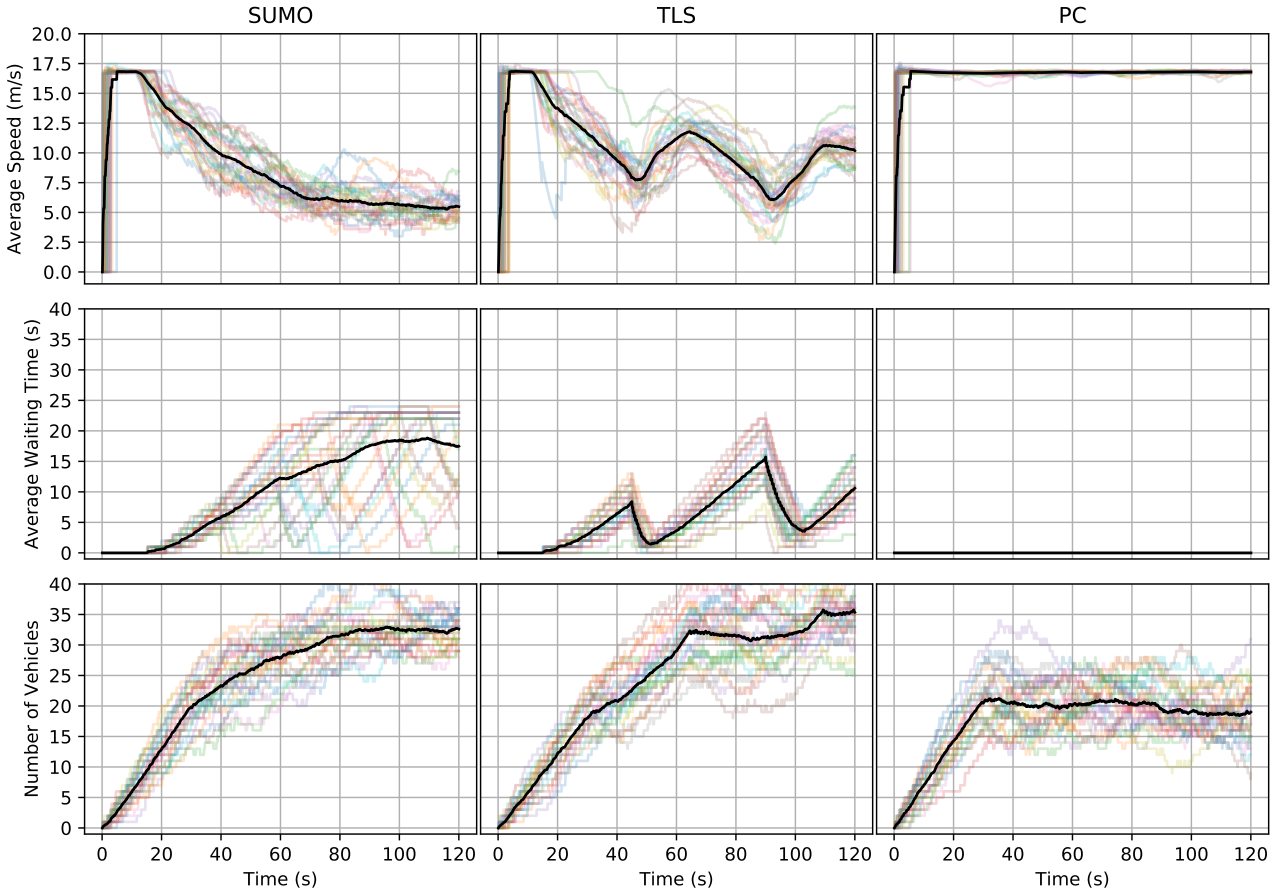
Source: Elaborated by the author.

Figure 51 – Comparison of models in the ramp merge heterogeneous vehicles with random speeds scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



Source: Elaborated by the author.

Figure 52 – Comparison of models in the ramp merge heterogeneous vehicles with random speeds and different private cost functions scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



Source: Elaborated by the author.

6.4 Lane Change

This experiment is distinct from the others as the vehicles enter the simulation dynamically, following a Poisson distribution, and their lane-change intentions are randomly generated, triggering a negotiation. Because it is a lane change problem this experiment also does not have a variation with TLS. Vehicles driving within a 50m radius in this road are called for negotiation. In addition, vehicles that are executing a previously negotiated maneuver within a 90m radius are also called for negotiation to share their planning.

The following Tables 21, 22, 23 and 24 summarize the results of the lane change scenario and again we show the three variations, homogeneous vehicles, heterogeneous vehicles, and heterogeneous vehicles with random speeds. We can compare results not only within the same experiment, but also between experiments, considering increasing complexity, heterogeneous vehicles, random speeds and different private cost functions. The tables show the average and standard deviation for the proposed metrics considering all 25 runs.

The Table 21 presents the first experiment variation of the lane change scenario. First, we can notice the same pattern of results as in the previous scenarios. As the SUMO model does not guarantee the right of way because some vehicles had to wait to pass through the junction, it presented the highest average travel time; on the other hand, the TLS model controls the passage of vehicles through the traffic light cycles and improves traffic flow. Our model, on the other hand, presents the shortest travel time and the highest average speed, which allowed for greater traffic flow.

We can do the same analysis as before for each lane change scenario presented in the Tables 22, 23 and 24. Likewise, our model keeps the speed close to the road limit and with a low standard deviation, even considering more complex scenarios. That led to a decrease in average travel time by maintaining a continuous traffic flow.

Table 21 – Comparison of models in the lane change homogeneous vehicles scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**PC**	
Statistical	AVG	STD	AVG	STD
Average Travel Time (s)	34.52	6.30	29.98	3.09
Average Speed (m/s)	14.22	2.81	16.56	0.08
Average Waiting Time (s)	2.35	3.37	0.00	0.00
Arrived Vehicles	20.24	3.28	23.36	6.04
Flow (veh/h)	607.20	98.47	700.80	181.22

Source: Research data.

Table 22 – Comparison of models in the lane change heterogeneous vehicles scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**PC**	
Statistical	AVG	STD	AVG	STD
Average Travel Time (s)	38.23	6.87	29.98	3.54
Average Speed (m/s)	11.77	3.35	16.50	0.12
Average Waiting Time (s)	3.29	2.00	0.00	0.00
Arrived Vehicles	17.84	3.02	22.08	4.67
Flow (veh/h)	535.20	90.70	662.40	140.16

Source: Research data.

Table 23 – Comparison of models in the lane change heterogeneous vehicles with random speeds scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Model	**SUMO**		**PC**	
Statistical	AVG	STD	AVG	STD
Average Travel Time (s)	35.38	6.38	29.80	3.01
Average Speed (m/s)	13.92	3.00	16.40	0.17
Average Waiting Time (s)	2.16	2.24	0.00	0.00
Arrived Vehicles	17.72	2.35	20.84	4.00
Flow (veh/h)	531.60	70.63	625.20	119.90

Source: Research data.

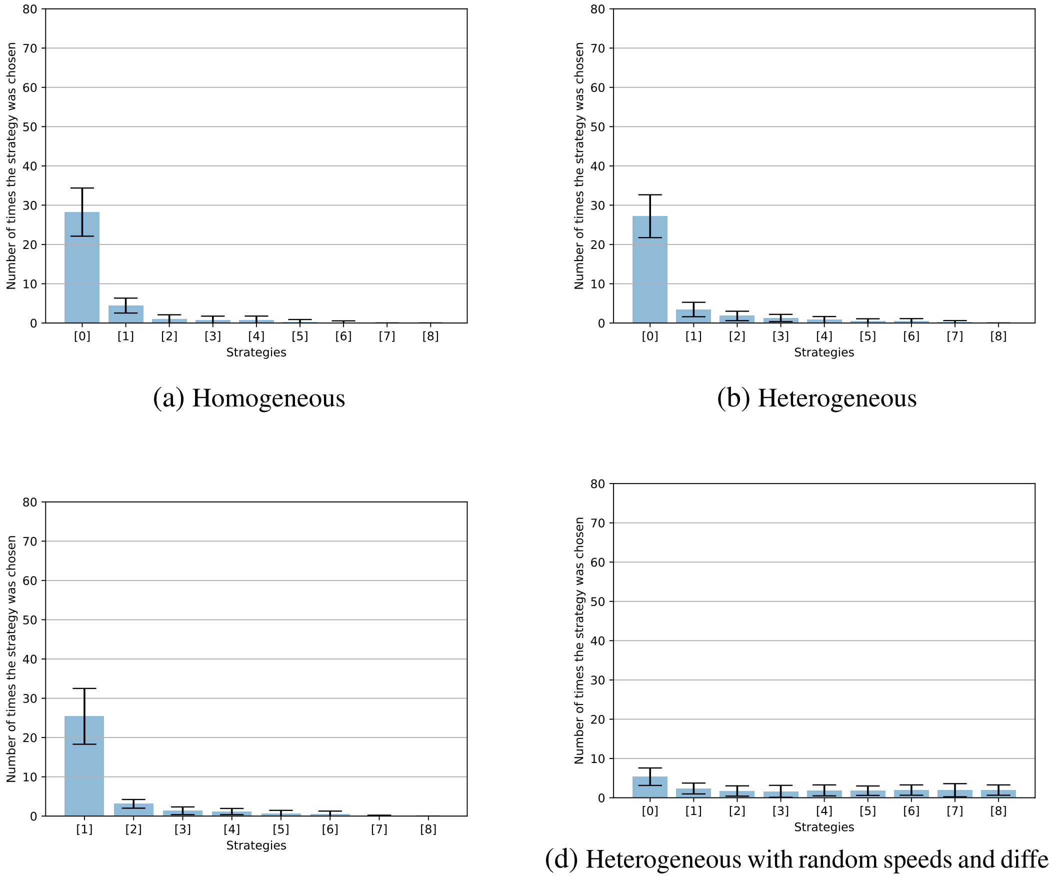
Table 24 – Comparison of models in the lane change heterogeneous vehicles with random speeds and different private cost functions scenario. For each model, the table shows the average and standard deviation for each proposed metric.

Algorithm	**SUMO**		**PC**	
Statistical	AVG	STD	AVG	STD
Average Travel Time (s)	31.23	5.36	30.65	4.12
Average Speed (m/s)	15.88	1.01	15.80	0.40
Average Waiting Time (s)	1.05	2.22	0.00	0.00
Arrived Vehicles	13.40	4.00	14.20	3.77
Flow (veh/h)	402.00	120.00	426.00	113.25

Source: Research data.

The summary statistics for the strategies chosen during 25 simulation runs is displayed in Figure 53. The blue bars indicate the average frequency of strategy selection, while the inner bars show the standard deviation. It can be observed that most of the selected strategies were those with the lowest cost. In this case, we have a decrease in the amount of Heterogeneous with random speeds and different private cost functions because we reduced the flow of vehicles and fewer vehicles were used in the simulation.

Figure 53 – Number of times that one strategy was chosen in the lane change scenario



(a) Homogeneous

(b) Heterogeneous

(c) Heterogeneous with random speeds

(d) Heterogeneous with random speeds and different private cost functions

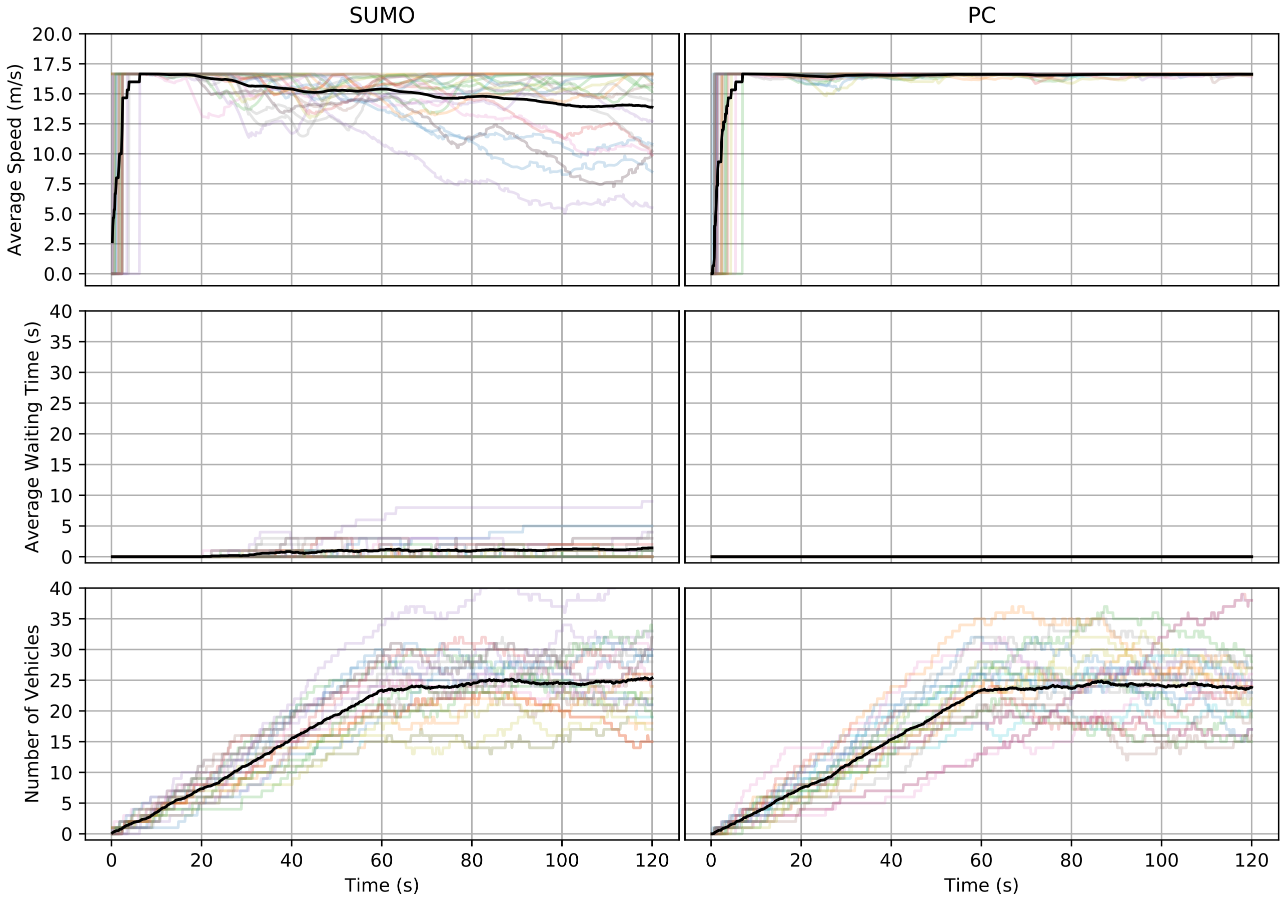
Source: Elaborated by the author.

The graphs in the Figures 54, 55, 56 and 57 show the results for average speed, average waiting time, and the number of vehicles over the experiment for the lane change experiments for all models. In this instance, we present the experiments over time to provide a more comprehensive analysis of the results presented in the previous tables.

First, as we analyzed before, there are several colored lines and a bold black line in the graphs. The colored lines of each graph in the background represent one specific run, and the black lines represent the mean of all runs. Although each experiment has its particularities, it is possible to notice the same pattern throughout the four experiments, homogeneous, heterogeneous, heterogeneous with random speeds, and heterogeneous with different private cost functions and random speeds for each model.

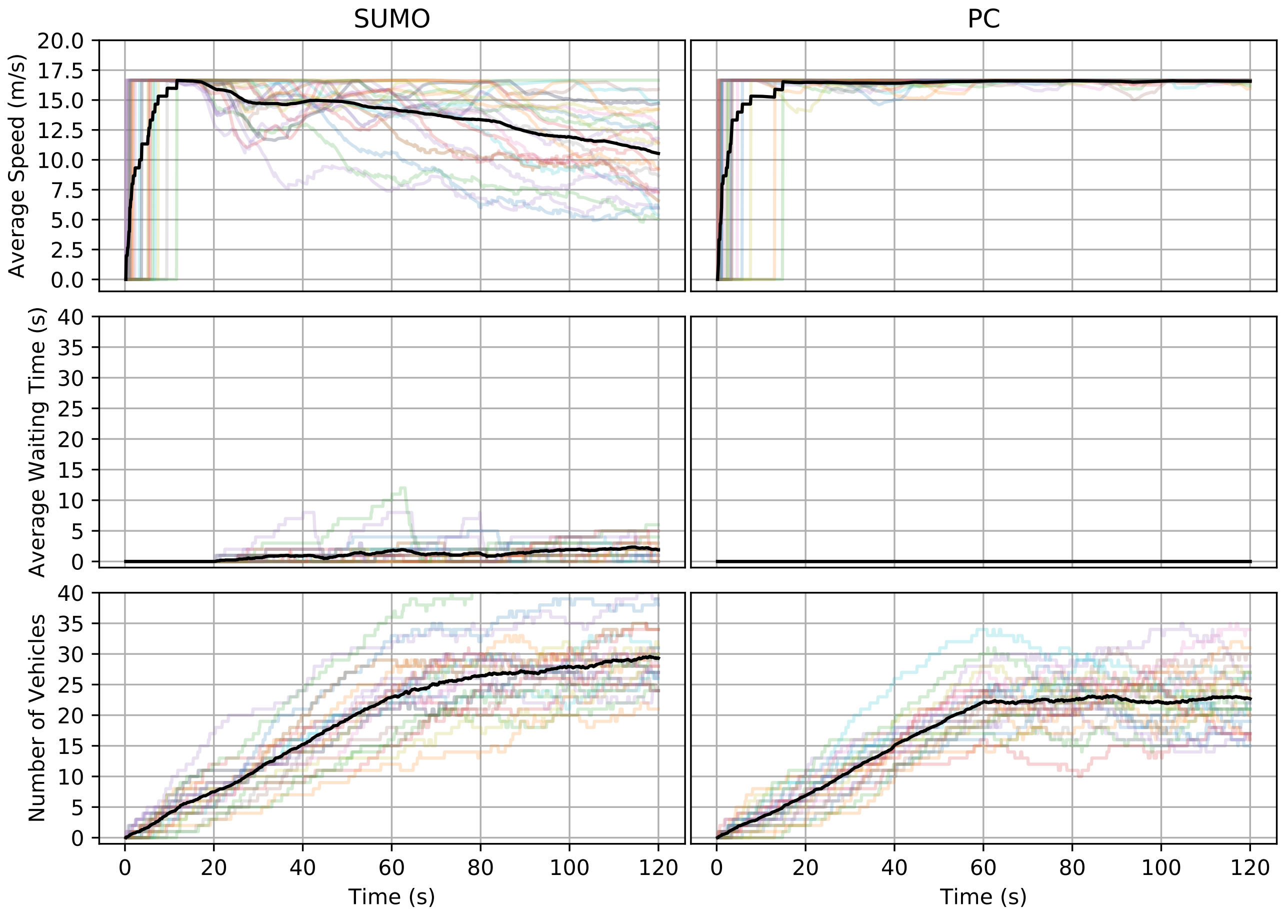
With regards to our PC-based model, there is no waiting time, as seen in the previous tables. This is achieved by adjusting the speed profile of the vehicles, which guarantees the right of way. As a result, both the average speed and the number of vehicles remain stable over time.

Figure 54 – Comparison of models in the lane change homogeneous vehicles scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



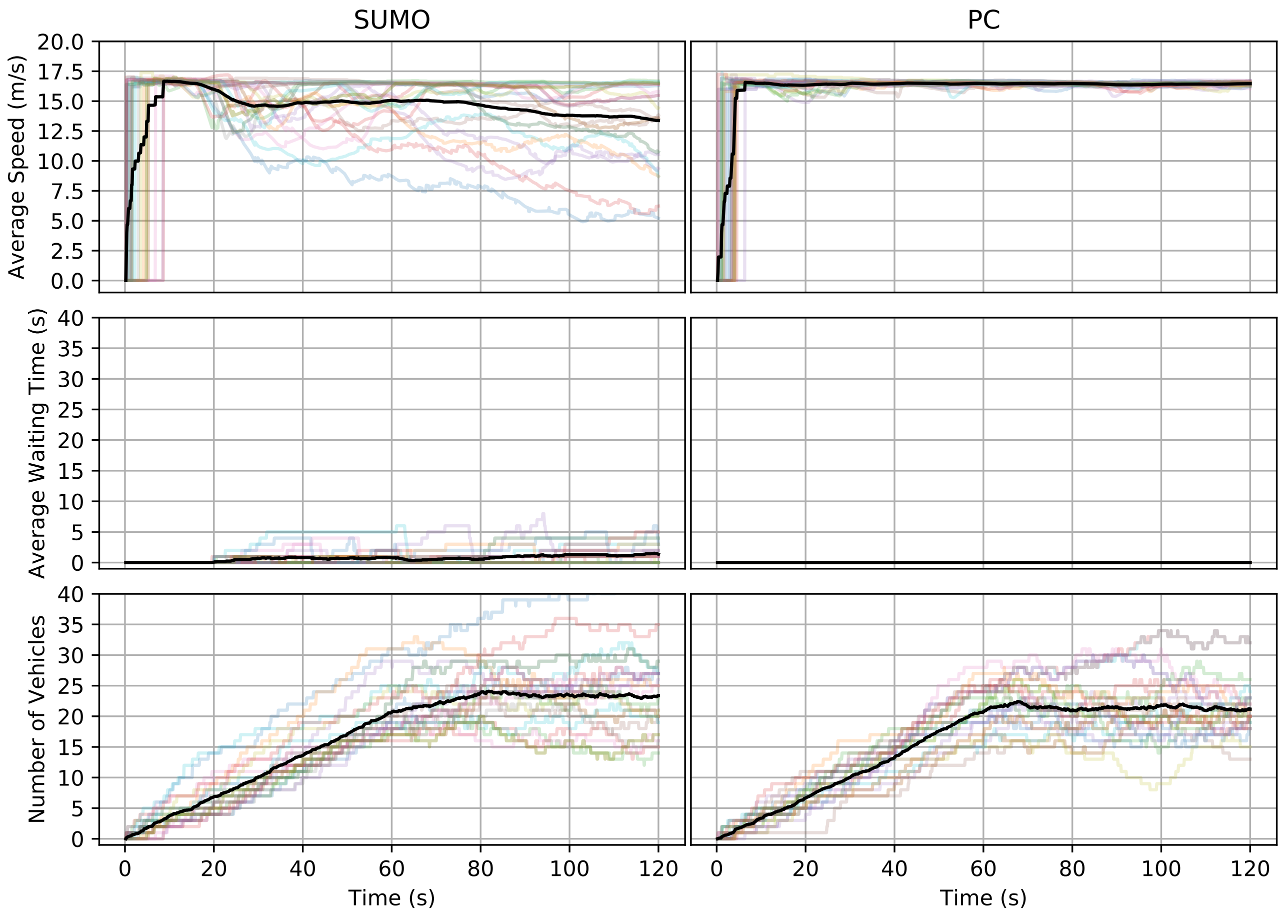
Source: Elaborated by the author.

Figure 55 – Comparison of models in the lane change heterogeneous vehicles scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



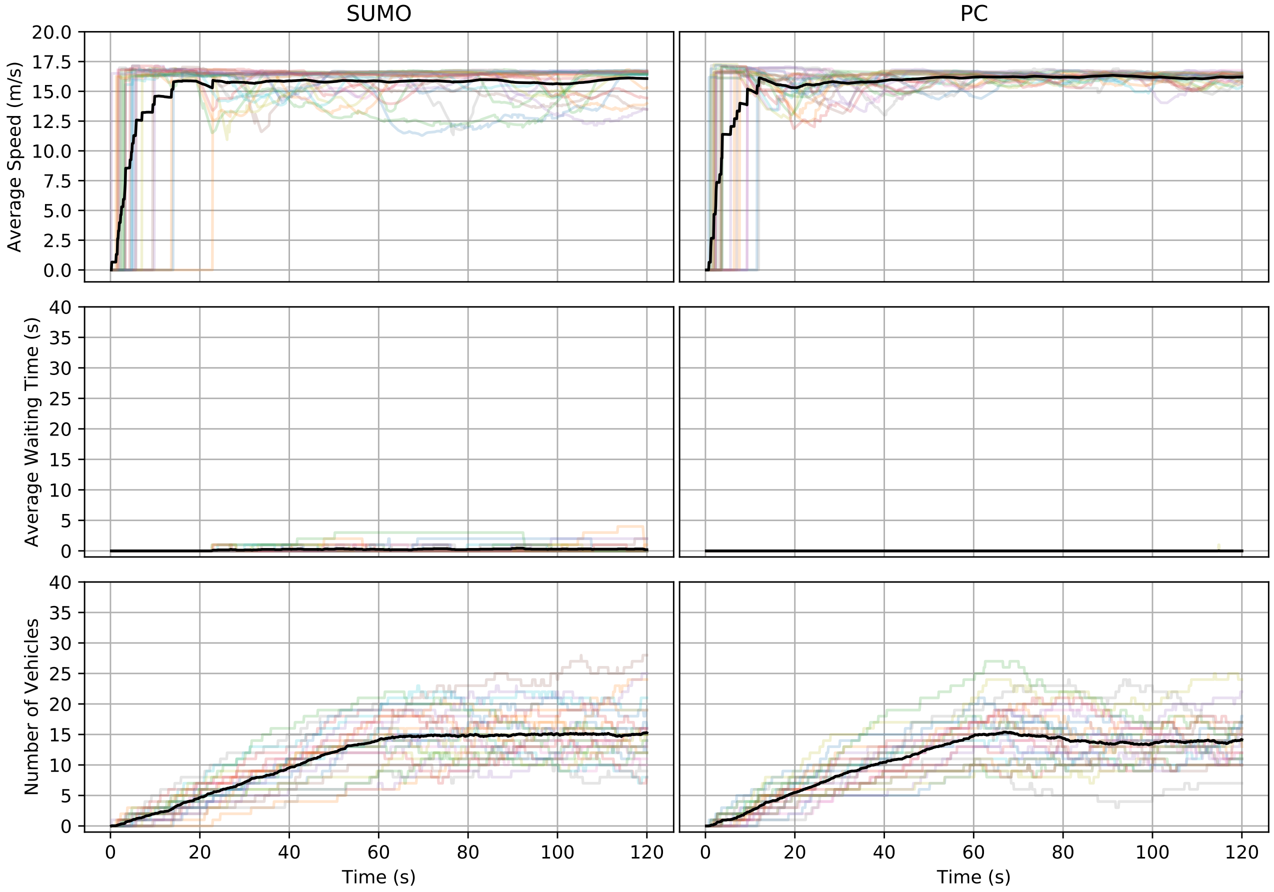
Source: Elaborated by the author.

Figure 56 – Comparison of models in the lane change homogeneous heterogeneous vehicles with random speeds scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



Source: Elaborated by the author.

Figure 57 – Comparison of models in the lane change heterogeneous vehicles with random speeds and different private cost functions scenario using the average speed, average waiting time, and the number of vehicles metrics during all experiment time.



Source: Elaborated by the author.

6.5 Final Remarks

When we analyze the overall profile across all scenarios, we can see that the behavior of all methods is very similar regardless of complexity. However, in many cases, the flow of vehicles decreases as the complexity of the scenario increases. In every scenario, significant enhancements were achieved with regards to the flow of vehicles, reducing the travel time and increasing the average speed.

In the case of the Crossroad 1-lane scenario, our approach showed an average improvement in the traffic flow of 34.63% and 25.50% compared to the SUMO and TLS approaches, respectively. It was also possible to achieve an improvement in travel time that decreased by an average of 69.29% and 55.07% compared to SUMO and TLS.

In the Crossroad 2-lanes scenario, our approach showed an improvement in the traffic flow in average of 12.89% and 15.10% compared to the SUMO and TLS approaches, respectively. Additionally, it was also possible to improve the travel time, which decreased by an average of 62.19% and 51.58% compared to SUMO and TLS.

In the Ramp merge scenario, our approach showed a significant improvement in the traffic flow on average of 86.69% and 46.62% compared to the SUMO and TLS approaches respectively. Additionally, a improvement in travel time was achieved, which decreased on average by 73.74% and 48.58% compared to SUMO and TLS.

In the Lane Change scenario, our approach showed an average improvement of 5.97% compared to SUMO. It was also possible to achieve a improvement in travel time that decreased by an average of 1.85% compared to SUMO. In the same scenario but with homogeneous vehicles, the improvements were on average 15.41% for traffic flow and 13.13%. In this case, we noticed a difficulty of the optimizer in dealing with the problem of vehicles with different dimensions and costs. The increase in the complexity of the problem resulted in momentary increases in density on the road during the simulations, which can lead the optimizer to not converge due to vehicle density.

CHAPTER 7

CONCLUSION AND DISCUSSION

The purpose of this thesis was to investigate the resolution of conflicts between Connected and Autonomous Vehicles in a decentralized manner in various dynamic scenarios. Our goal was to present a decentralized method for conflict resolution in dynamic environments, as well as to integrate scenarios involving vehicles of varying dimensions, random speeds, and different private cost functions.

Our adaptation of the Probability Collectives method was able to effectively resolve conflict resolution problems in dynamic crossroad 1-lane, crossroad 2-lane, on-ramp merge, and lane change scenarios using a unified solution. We conducted experiments using well established methods in a simulation tool as benchmarks to evaluate the efficiency of our approach. This has significant implications for ITS and urban mobility, generating a direct impact on increasing traffic flow, reducing vehicle travel time and reducing pollutant emissions.

While this study has made a contribution to the ITS area, there are also limitations. We noticed some limitation regarding the SUMO simulator, which was the main tool used in this work, and also some limitations in our proposed method. However, these limitations provide a valuable opportunity for future research to build upon and address these gaps.

The first limitation concerns the calculation of the future trajectory of vehicles. In SUMO it is not possible to know the future trajectory, because the vehicles act according to the car-following and lane-changing models. Our proposed method effectively addressed many of the limitations commonly encountered with simplified maps, such as the ones we use in our experiments. However, for more complex studies it may be interesting to use larger maps, such as city sections, and in this case it will be necessary to adapt our trajectory calculation proposal to improve efficiency and avoid problems at junctions, connections between different roads and a large number of routes.

Another important issue regarding the simulator is the use of emergency braking when two vehicles are very close. As we use the car-following model itself to guide the vehicles at times when our strategy execution manager is not activated, this could lead to some proximity situations in which braking was activated. This was mostly noticed in lane change scenarios.

Regarding our proposal, future studies can explore different traffic situations such as overtaking and different types of structures in a road network. And also consider using sections of real cities and even performing a simulator calibration for that particular region with real data as performed by Poli *et al.* (2021). Investigating negotiation algorithms in more complex and realistic scenarios would be a major advancement for the field of ITS.

In this work, we simplified the negotiation protocol process by performing all negotiations within a single simulation time step. However, it would be very interesting to carry out the negotiation in such a way that each communication iteration is run in a simulation step. This would increase the complexity of the problem, as vehicles would be negotiating while driving the road. The negotiations might become invalid if they take too long to complete. According to Poli *et al.* (2021), the longer negotiation time results in a higher rejection rate of proposals. In such cases, it is likely that the trajectory time interval should be longer than the 10 seconds we set in our experiments. This would provide vehicles with sufficient time to negotiate effectively.

One can also add a network simulator in this negotiation protocol to check the influence of the network on the method. In which problems of delay, packet size, communication radius and signal occlusion can be evaluated.

Comparing decision-making methods, especially those that involve randomness, can be challenging. Implemented methods can introduce errors if not performed by the same author, making it important to establish a benchmark for comparison. This thesis makes a small but valuable contribution by presenting a way to compare our proposed method with established methods in the simulation tool, providing detailed information on vehicle parameters and scenarios. In this way, other authors can compare themselves with our method using the simulation tool as a reference point.

BIBLIOGRAPHY

ANTOINE, N.; BIENIAWSKI, S.; KROO, I.; WOLPERT, D. Fleet assignment using collective intelligence. 02 2004. Citation on page 58.

AREM, B. van; DRIEL, C. J. G. van; VISSER, R. The impact of cooperative adaptive cruise control on traffic-flow characteristics. **IEEE Transactions on Intelligent Transportation Systems**, v. 7, n. 4, p. 429–436, 2006. Citation on page 32.

AZIMI, R.; BHATIA, G.; RAJKUMAR, R. R.; MUDALIGE, P. Stip: Spatio-temporal intersection protocols for autonomous vehicles. In: **2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)**. [S.l.: s.n.], 2014. p. 1–12. Citation on page 35.

BEHRISCH, M.; BIEKER-WALZ, L.; ERDMANN, J.; KRAJZEWICZ, D. Sumo – simulation of urban mobility: An overview. In: . [S.l.: s.n.], 2011. v. 2011. ISBN 978-1-61208-169-4. Citation on page 62.

BENGTSSON, H. H.; CHEN, L.; VORONOV, A.; ENGLUND, C. Interaction protocol for highway platoon merge. In: **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. [S.l.: s.n.], 2015. p. 1971–1976. ISSN 2153-0009. Citation on page 39.

BROGGI, A.; BERTOZZI, M.; FASCIOLI, A.; CONTE, G. **Automatic Vehicle Guidance: the Experience of the ARGO Vehicle**. Singapore: World Scientific, 1999. ISBN 9810237200. Citation on page 27.

BUCKMAN, N.; PIERSON, A.; SCHWARTING, W.; KARAMAN, S.; RUS, D. Sharing is caring: Socially-compliant autonomous intersection negotiation. In: **2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2019. p. 6136–6143. Citation on page 52.

CHANKONG, V.; HAIMES, Y. Y. **Multiobjective decision making: theory and methodology**. [S.l.]: Courier Dover Publications, 2008. Citation on page 143.

CHEN, L.; ENGLUND, C. Cooperative intersection management: A survey. **IEEE Transactions on Intelligent Transportation Systems**, v. 17, n. 2, p. 570–586, 2016. Citations on pages 29, 33, 34, and 53.

CRISMAN, J.; THORPE, C. Scarf: A color vision system that tracks roads and intersections. **IEEE Trans. on Robotics and Automation**, v. 9, n. 1, p. 49 – 58, February 1993. Citation on page 27.

DEB, K.; AGRAWAL, S.; PRATAP, A.; MEYARIVAN, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: SCHOENAUER, M.; DEB, K.; RUDOLPH, G.; YAO, X.; LUTTON, E.; MERELO, J. J.; SCHWEFEL, H.-P. (Ed.). **Parallel Problem Solving from Nature PPSN VI**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. p. 849–858. ISBN 978-3-540-45356-7. Citation on page 143.

DONGXIN, L.; QIQIGE, W.; WENBO, C.; HUILONG, Y.; XIAOPING, D. A priority tree based coordination method for intelligent and connected vehicles at unsignalized intersections. **IET Intelligent Transport Systems**, v. 15, n. 8, p. 1053–1063, 2021. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/itr2.12082>. Citations on pages 35, 53, 78, and 79.

DRESNER, K.; STONE, P. Multiagent traffic management: a reservation-based intersection control mechanism. In: **Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.** [S.l.: s.n.], 2004. p. 530–537. Citation on page 34.

____. Multiagent traffic management: An improved intersection control mechanism. In: **Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems**. New York, NY, USA: Association for Computing Machinery, 2005. (AAMAS '05), p. 471–477. ISBN 1595930930. Available: <https://doi.org/10.1145/1082473.1082545>. Citation on page 34.

EIERMANN, L.; SAWADE, O.; BUNK, S.; BREUEL, G.; RADUSCH, I. Cooperative automated lane merge with role-based negotiation. In: **2020 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2020. p. 495–501. Citation on page 36.

ELLIOTT, C. M.; TALLANT, G.; DOGAN, A. On probability collectives for distributed control allocation. In: **2017 IEEE Aerospace Conference**. [S.l.: s.n.], 2017. p. 1–11. Citation on page 45.

ENGLUND, C.; CHEN, L.; PLOEG, J.; SEMSAR-KAZEROONI, E.; VORONOV, A.; BENGTSSON, H. H.; DIDOFF, J. The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles. **IEEE Wireless Communications**, v. 23, n. 4, p. 146–152, August 2016. ISSN 1536-1284. Citations on pages 28 and 39.

ENGLUND, C.; CHEN, L.; VINEL, A.; LIN, S. Y. Future applications of vanets. In: ____. **Vehicular ad hoc Networks: Standards, Solutions, and Research**. Cham: Springer International Publishing, 2015. p. 525–544. Available: <https://doi.org/10.1007/978-3-319-15497-8_18>. Citations on pages 27 and 28.

ERDMANN, J. Sumo's lane-changing model. In: BEHRISCH, M.; WEBER, M. (Ed.). **2nd SUMO User Conference**. Springer Verlag, 2015. (Lecture Notes in Control and Information Sciences, v. 13), p. 105–123. Available: <https://elib.dlr.de/102254/>. Citation on page 78.

FAA, U. **Introduction to TCAS II, Version 7.1**. 2011. Citation on page 29.

GACIARZ, M.; AKNINE, S.; BHOURI, N. Automated negotiation for traffic regulation. In: KOCH, F.; GUTTMANN, C.; BUSQUETS, D. (Ed.). **Advances in Social Computing and Multiagent Systems**. Cham: Springer International Publishing, 2015. p. 1–18. ISBN 978-3-319-24804-2. Citations on pages 39 and 73.

GHOLAMHOSSEINIAN, A.; SEITZ, J. A comprehensive survey on cooperative intersection management for heterogeneous connected vehicles. **IEEE Access**, v. 10, p. 7937–7972, 2022. Citations on pages 33 and 34.

GOTTSCHALK, S. Separating axis theorem. Technical Report TR96-024, Department of Computer Science, UNC Chapel Hill, 1996. Citation on page 70.

GOTTSCHALK, S.; LIN, M. C.; MANOCHA, D. Obbtree: A hierarchical structure for rapid interference detection. In: **Proceedings of the 23rd annual conference on Computer graphics and interactive techniques**. [S.l.: s.n.], 1996. p. 171–180. Citation on page 69.

HAN, Z.; NING, C.; WEI, Y. Mopso for bim: a multi-objective optimization tool using particle swarm optimization algorithm on a bimbased visual programming platform. **Hello, Culture**, p. 39–51, 2019. Citations on pages 143 and 147.

KARAGIANNIS, G.; ALTINTAS, O.; EKICI, E.; HEIJENK, G.; JARUPAN, B.; LIN, K.; WEIL, T. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. **IEEE Communications Surveys Tutorials**, v. 13, n. 4, p. 584–616, Fourth 2011. ISSN 1553-877X. Citation on page 33.

KENNEY, J. B. Dedicated short-range communications (dsrc) standards in the united states. **Proceedings of the IEEE**, v. 99, n. 7, p. 1162–1182, 2011. Citation on page 31.

KHAYATIAN, M.; MEHRABIAN, M.; ANDERT, E.; DEDINSKY, R.; CHOUDHARY, S.; LOU, Y.; SHIRVASTAVA, A. A survey on intersection management of connected autonomous vehicles. **ACM Trans. Cyber-Phys. Syst.**, Association for Computing Machinery, New York, NY, USA, v. 4, n. 4, aug 2020. ISSN 2378-962X. Available: <https://doi.org/10.1145/3407903>. Citations on pages 27, 28, 29, 33, and 34.

KLISCHAT, M.; DRAGOI, O.; EISSA, M.; ALTHOFF, M. Coupling sumo with a motion planning framework for automated vehicles. In: . [S.l.: s.n.], 2019. Citation on page 65.

KRAJZEWICZ, D.; ERDMANN, J. Road intersection model in sumo. In: . [S.l.: s.n.], 2013. p. 212–220. Citation on page 78.

KRAJZEWICZ, D.; ERDMANN, J.; BEHRISCH, M.; BIEKER, L. Recent development and applications of SUMO - Simulation of Urban MObility. **International Journal On Advances in Systems and Measurements**, v. 5, n. 3&4, p. 128–138, December 2012. Citation on page 62.

KRAUSS, S. **Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics**. 1998. Citation on page 63.

KRAUSS, S.; WAGNER, P.; GAWRON, C. Metastable states in a microscopic model of traffic flow. **Phys. Rev. E**, American Physical Society, v. 55, p. 5597–5602, May 1997. Available: <https://link.aps.org/doi/10.1103/PhysRevE.55.5597>. Citation on page 63.

KULKARNI, A.; TAI, K. Probability collectives: A distributed optimization approach for constrained problems. In: . [S.l.: s.n.], 2010. v. 86, p. 1–8. Citations on pages 43 and 48.

KULKARNI, A. J.; TAI, K. Solving constrained optimization problems using probability collectives and a penalty function approach. **International Journal of Computational Intelligence and Applications**, v. 10, n. 04, p. 445–470, 2011. Citation on page 43.

____. A probability collectives approach for multi-agent distributed and cooperative optimization with tolerance for agent failure. In: ____. **Agent-Based Optimization**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 175–201. Citation on page 43.

KULKARNI, A. J.; TAI, K.; ABRAHAM, A. **Probability Collectives: A Distributed Multi-Agent System Approach for Optimization**. [S.l.]: Springer Publishing Company, Incorporated, 2015. ISBN 3319159992. Citations on pages 43, 45, 47, 48, 49, 55, 59, and 143.

LAMÉ, G. **Examen des différentes méthodes employées pour résoudre les problèmes de géométrie**. Librairie scientifique J. Hermann, 1818. Available: <https://books.google.com.br/books?id=FjMDAAAAQAAJ>. Citation on page 69.

LEVIN, M. W.; REY, D. Conflict-point formulation of intersection control for autonomous vehicles. **Transportation Research Part C: Emerging Technologies**, v. 85, p. 528–547, 2017. ISSN 0968-090X. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X17302735>. Citations on pages 35, 53, and 79.

LISSAC, A.; DJAHEL, S.; HODGKISS, J. Infrastructure assisted automation of lane change manoeuvre for connected and autonomous vehicles. In: **2019 IEEE International Smart Cities Conference (ISC2)**. [S.l.: s.n.], 2019. p. 173–180. Citation on page 79.

LIU, C.; LIN, C.-W.; SHIRAISHI, S.; TOMIZUKA, M. Distributed conflict resolution for connected autonomous vehicles. **IEEE Transactions on Intelligent Vehicles**, v. 3, n. 1, p. 18–29, 2018. Citations on pages 29 and 40.

LOPEZ, P. A.; BEHRISCH, M.; BIEKER-WALZ, L.; ERDMANN, J.; FLÖTTERÖD, Y.-P.; HILBRICH, R.; LÜCKEN, L.; RUMMEL, J.; WAGNER, P.; WIESSNER, E. Microscopic traffic simulation using sumo. In: **The 21st IEEE International Conference on Intelligent Transportation Systems**. IEEE, 2018. p. 2575–2582. Available: <https://elib.dlr.de/127994/>. Citation on page 62.

MAKSIMOVSKI, D.; FESTAG, A.; FACCHI, C. A survey on decentralized cooperative maneuver coordination for connected and automated vehicles. In: . [S.l.: s.n.], 2021. p. 100–111. Citations on pages 27, 28, and 29.

MALIK, F. M.; KHATTAK, H. A.; ALMOGREN, A.; BOUACHIR, O.; DIN, I. U.; ALTAMEEM, A. Performance evaluation of data dissemination protocols for connected autonomous vehicles. **IEEE Access**, v. 8, p. 126896–126906, 2020. Citation on page 27.

MENG, Y.; LI, L.; WANG, F.-Y.; LI, K.; LI, Z. Analysis of cooperative driving strategies for nonsignalized intersections. **IEEE Transactions on Vehicular Technology**, v. 67, n. 4, p. 2900–2911, 2018. Citations on pages 39 and 52.

MORIDPOUR, S.; SARVI, M.; ROSE, G. Lane changing models: A critical review. **Transportation Letters: The International Journal of Transportation Research**, v. 2, p. 157–173, 07 2010. Citation on page 37.

NASH, J. The bargaining problem. **Econometrica**, v. 18, n. 2, p. 155–162, 1950. Available: <https://EconPapers.repec.org/RePEc:ecm:emetrp:v:18:y:1950:i:2:p:155-162>. Citation on page 38.

NICHTING, M.; HEß, D.; SCHINDLER, J.; HESSE, T.; KöSTER, F. Space time reservation procedure (strp) for v2x-based maneuver coordination of cooperative automated vehicles in diverse conflict scenarios. In: **2020 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2020. p. 502–509. Citation on page 40.

NUNEN, E. van; KWAKKERNAAT, R.; PLOEG, J.; NETTEN, B. Cooperative competition for future mobility. **Intelligent Transportation Systems, IEEE Transactions on**, v. 13, n. 3, p. 1018–1025, 2012. Citation on page 27.

PHILIPPE, C. **Reliable and safe control Navigation for Autonomous Vehicles in Dynamic Urban Environments**. Phd Thesis (PhD Thesis), 06 2020. Citations on pages 40 and 73.

PHILIPPE, C.; ADOUANE, L.; TSOURDOS, A.; SHIN, H.-S.; THUILOT, B. Probability collectives algorithm applied to decentralized intersection coordination for connected autonomous vehicles. In: **2019 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2019. p. 1928–1934. Citation on page 40.

PLOEG, J.; SEMSAR-KAZEROONI, E.; MEDINA, A. I. M.; JONGH, J. F. C. M. de; SLUIS, J. van de; VORONOV, A.; ENGLUND, C.; BRIL, R. J.; SALUNKHE, H.; ARRUE, A.; RUANO, A.; GARCIA-SOL, L.; NUNEN, E. van; WOUW, N. van de. Cooperative automated maneuvering at the 2016 grand cooperative driving challenge. **IEEE Transactions on Intelligent Transportation Systems**, v. 19, n. 4, p. 1213–1226, April 2018. ISSN 1524-9050. Citations on pages 28, 37, and 39.

POLI, F.; DENIS, B.; MANNONI, V.; BERG, V.; MARTíN-SACRISTáN, D.; GARCIA-ROGER, D.; MONSERRAT, J. F. Evaluation of c-v2x sidelink for cooperative lane merging in a cross-border highway scenario. In: **2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)**. [S.l.: s.n.], 2021. p. 1–5. Citations on pages 37 and 134.

PRITCHETT, A. R.; GENTON, A. Negotiated decentralized aircraft conflict resolution. **IEEE Transactions on Intelligent Transportation Systems**, PP, n. 99, p. 1–11, 2017. ISSN 1524-9050. Citation on page 38.

RIOS-TORRES, J.; MALIKOPOULOS, A. A. Automated and cooperative vehicle merging at highway on-ramps. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 18, n. 4, p. 780–789, 2016. Citation on page 36.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Prentice Hall, 2010. (Prentice Hall series in artificial intelligence). ISBN 9780136042594. Available: <https://books.google.com.br/books?id=8jZBksh-bUMC>. Citation on page 32.

SANTOS, T. C. dos; WOLF, D. F. Automated conflict resolution of lane change utilizing probability collectives. In: **2019 19th International Conference on Advanced Robotics (ICAR)**. [S.l.: s.n.], 2019. p. 623–628. Citation on page 40.

____. Bargaining game approach for lane change maneuvers. In: **2019 19th International Conference on Advanced Robotics (ICAR)**. [S.l.: s.n.], 2019. p. 629–634. Citation on page 38.

SCHAFFER, J. Multiple objective optimization with vector evaluated genetic algorithms. In: . [S.l.: s.n.], 1985. p. 93–100. Citation on page 143.

SCHEPPERLE, H.; BöHM, K.; FORSTER, S. Towards valuation-aware agent-based traffic control. In: **Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems**. New York, NY, USA: Association for Computing Machinery, 2007. (AAMAS '07). ISBN 9788190426275. Available: <https://doi.org/10.1145/1329125.1329349>. Citation on page 79.

SISLAK, D.; VOLF, P.; PECHOUCEK, M.; SURI, N. Automated conflict resolution utilizing probability collectives optimizer. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 41, n. 3, p. 365–375, 2011. Citations on pages 40, 49, 56, and 143.

TALEBPOUR, A.; MAHMASSANI, H. Influence of connected and autonomous vehicles on traffic flow stability and throughput. **Transportation Research Part C: Emerging Technologies**, Elsevier Limited, v. 71, p. 143–163, Oct. 2016. ISSN 0968-090X. Citation on page 33.

THORPE, C.; HEBERT, M.; KANADE, T.; SHAFER, S. Vision and navigation for the carnegie-mellon navlab. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 10, n. 3, p. 362 – 373, May 1988. Citation on page 27.

THRUN, S.; MONTEMERLO, M.; DAHLKAMP, H.; STAVENS, D.; ARON, A.; DIEBEL, J.; FONG, P.; GALE, J.; HALPENNY, M.; HOFFMANN, G.; LAU, K.; OAKLEY, C.; PALATUCCI, M.; PRATT, V.; STANG, P.; STROHBAND, S.; DUPONT, C.; JENDROSSEK, L.-E.; KOELEN, C.; MARKEY, C.; RUMMEL, C.; NIEKERK, J. van; JENSEN, E.; ALESSANDRINI, P.; BRADSKI, G.; DAVIES, B.; ETTINGER, S.; KAEHLER, A.; NEFIAN, A.; MAHONEY, P. Stanley: The robot that won the darpa grand challenge: Research articles. **J. Robot. Syst.**, John Wiley and Sons Ltd., Chichester, UK, v. 23, n. 9, p. 661–692, Sep. 2006. ISSN 0741-2223. Available: <http://dx.doi.org.ez67.periodicos.capes.gov.br/10.1002/rob.v23:9>. Citation on page 27.

TIENTRAKOOL, P.; HO, Y.-C.; MAXEMCHUK, N. F. Highway capacity benefits from using vehicle-to-vehicle communication and sensors for collision avoidance. In: **2011 IEEE Vehicular Technology Conference (VTC Fall)**. [S.l.: s.n.], 2011. p. 1–5. Citation on page 32.

TOLEDO, T.; ZOHAR, D. Modeling duration of lane changes. **Transportation Research Record**, v. 1999, n. 1, p. 71–78, 2007. Citation on page 71.

URMSON, C.; ANHALT, J.; BAGNELL, D.; BAKER, C.; BITTNER, R.; CLARK, M. N.; DOLAN, J.; DUGGINS, D.; GALATALI, T.; GEYER, C.; GITTLEMAN, M.; HARBAUGH, S.; HEBERT, M.; HOWARD, T. M.; KOLSKI, S.; KELLY, A.; LIKHACHEV, M.; MCNAUGHTON, M.; MILLER, N.; PETERSON, K.; PILNICK, B.; RAJKUMAR, R.; RYBSKI, P.; SALESKY, B.; SEO, Y.-W.; SINGH, S.; SNIDER, J.; STENTZ, A.; WHITTAKER, W. R.; WOLKOWICKI, Z.; ZIGLAR, J.; BAE, H.; BROWN, T.; DEMITRISH, D.; LITKOUHI, B.; NICKOLAOU, J.; SADEKAR, V.; ZHANG, W.; STRUBLE, J.; TAYLOR, M.; DARMS, M.; FERGUSON, D. Autonomous driving in urban environments: Boss and the urban challenge. **J. Field Robot.**, John Wiley and Sons Ltd., Chichester, UK, v. 25, n. 8, p. 425–466, Aug. 2008. ISSN 1556-4959. Available: <http://dx.doi.org.ez67.periodicos.capes.gov.br/10.1002/rob.v25:8>. Citation on page 27.

WALDOCK, A.; CORNE, D. Multi-objective probability collectives. In: **Applications of Evolutionary Computation**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 461–470. Citation on page 49.

WANG, Y.; CAI, P.; LU, G. Cooperative autonomous traffic organization method for connected automated vehicles in multi-intersection road networks. **Transportation Research Part C: Emerging Technologies**, v. 111, p. 458–476, 2020. ISSN 0968-090X. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X19305686>. Citation on page 79.

WOLPERT, D. H.; STRAUSS, C. E.; RAJNARAYAN, D. Advances in distributed optimization using probability collectives. **Advances in Complex Systems**, World Scientific, v. 9, n. 04, p. 383–436, 2006. Citations on pages 43, 45, and 47.

YOUSEFI, S.; MOUSAVI, M. S.; FATHY, M. Vehicular ad hoc networks (vanets): Challenges and perspectives. In: **2006 6th International Conference on ITS Telecommunications**. [S.l.: s.n.], 2006. p. 761–766. Citation on page 31.

ZADEH, L. Optimality and non-scalar-valued performance criteria. **IEEE Transactions on Automatic Control**, v. 8, n. 1, p. 59–60, 1963. Citation on page 49.

ZHAO, Z.; WANG, Z.; WU, G.; YE, F.; BARTH, M. J. The state-of-the-art of coordinated ramp control with mixed traffic conditions. In: **2019 IEEE Intelligent Transportation Systems Conference (ITSC)**. [S.l.: s.n.], 2019. p. 1741–1748. Citation on page 36.

ŠIšLáK, D.; VOLF, P.; PěCHOUčEK, M. Agent-based cooperative decentralized airplane-collision avoidance. **IEEE Transactions on Intelligent Transportation Systems**, v. 12, n. 1, p. 36–46, 2011. Citation on page 38.

APPENDIX A

PROBABILITY COLLECTIVES FOR MULTI-OBJECTIVE

In this appendix, we evaluate the solutions obtained for resolving three benchmark problems in multi-objective optimization using our proposed algorithm in Section 3.1, which is based on (SISLAK *et al.*, 2011) and (KULKARNI; TAI; ABRAHAM, 2015). Our objective is to evaluate the convergence of the algorithm and compare with Non-dominated Sorting Genetic Algorithm (NSGA-II) (also known as NSGA2) proposed by Deb *et al.* (2000), which is able to find the Pareto frontier.

We analyzed three functions: the Chankong and Haimes function which was proposed by Chankong and Haimes (2008), the Schaffer function N. 1 proposed by Schaffer (1985) and the Constr-Ex problem proposed by Han, Ning and Wei (2019).

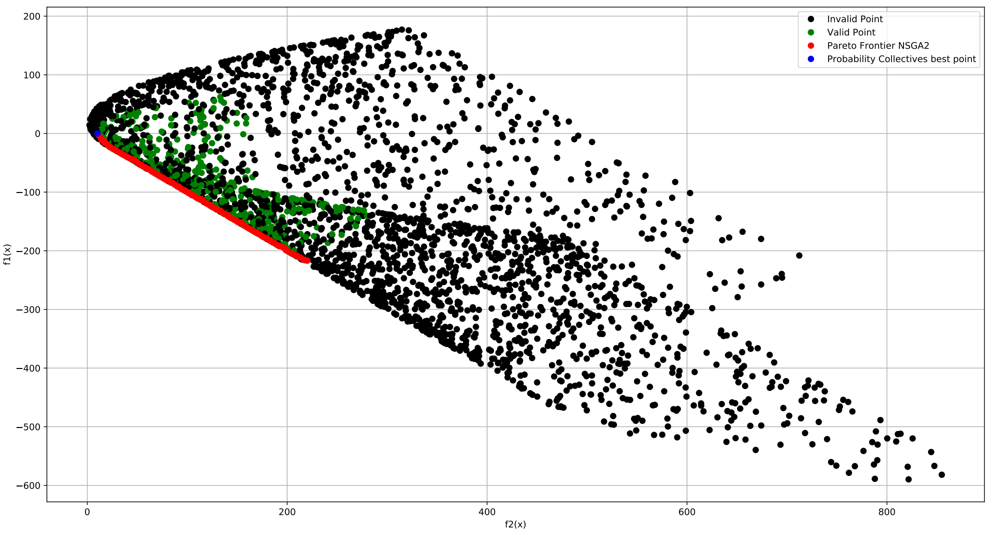
For this study, we use the pymoo Multi-objective Optimization in Python version 0.5.0 tool, which includes an NSGA2 implementation for Python 3. We defined the multi-objective problems and collect the Pareto frontier points which are represented by the red color in the graphs below.

For all functions, after collecting the Pareto frontier points, we run our proposed Probability Collectives for optimizing generic functions presented in Section 3.1 and compare the solution with the NSGA2 Pareto frontier.

A.1 Chankong and Haimes function

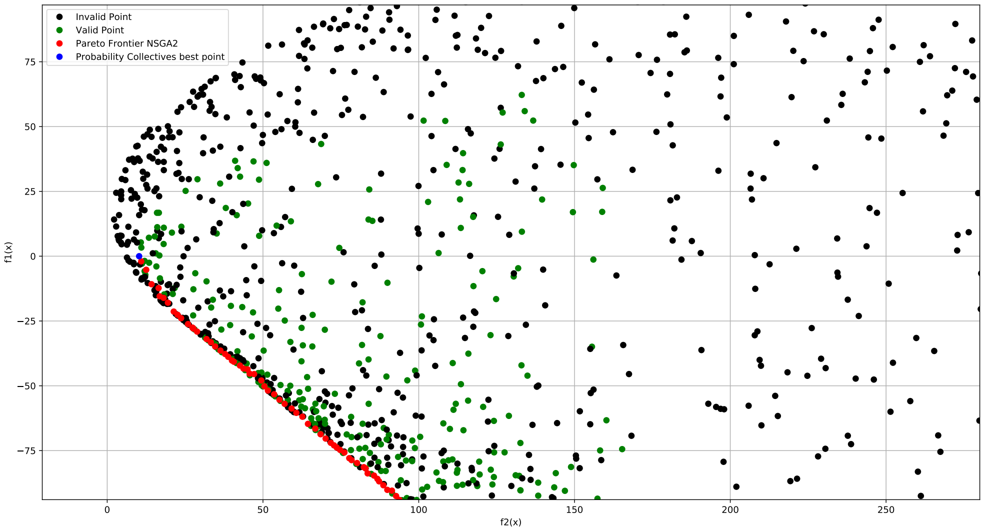
$$
\begin{aligned}
&\text{Minimize} = \begin{cases} f_1(x,y) = 2 + (x-1)^2 + (y-1)^2 \\[2ex] f_2(x,y) = 9x - (y-1)^2 \end{cases} \\[4ex]
&\text{Subject to} = \begin{cases} g_1(x,y) = x^2 + y^2 \leq 255 \\[2ex] g_2(x,y) = x - 3y + 10 \leq 0 \end{cases} \\[4ex]
&\text{Search domain} = -20 \leq x, y \leq 20
\end{aligned}
\tag{A.1}
$$

Figure 58 – The outcome of minimizing the Chankong and Haimes function is shown. Black points denote invalid values, while green points indicate valid values that fulfill the constraints but are not optimal. The red points depict the Pareto frontier calculated with NSGA2, and the blue point represents the optimal result of the optimization carried out by the PC.



Source: Elaborated by the author.

Figure 59 – The figure provides a closer look at the Pareto frontier area of the Chankong and Haimes function



Source: Elaborated by the author.

In this first problem, the algorithm found a point close to the Pareto frontier and coincidentally close to a "knee point", which in this case is a point that has a good cost balance.

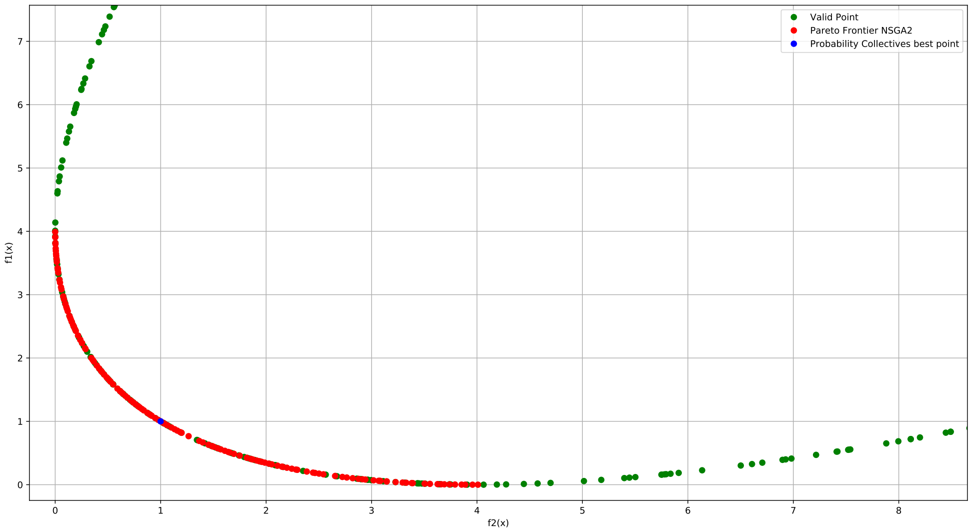
$$\text{Probability Collectives best value} \begin{cases} x = & 10.248385761510693 \\ y = & -0.00068195045497532l9 \end{cases}$$

A.2 Schaffer function N. 1

$$\text{Minimize} = \begin{cases} f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \end{cases} \tag{A.2}$$

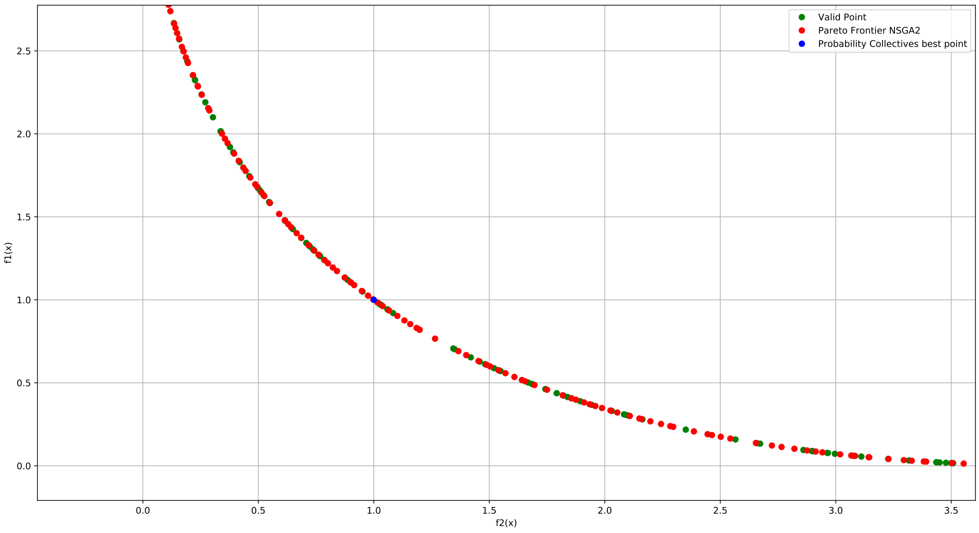
$$\text{Search domain} = -10 \leq x \leq 10$$

Figure 60 – The outcome of minimizing the Schaffer function N. 1 is shown. Green points indicate valid values that fulfill the constraints but are not optimal. The red points depict the Pareto frontier calculated with NSGA2, and the blue point represents the optimal result of the optimization carried out by the PC.



Source: Elaborated by the author.

Figure 61 – The figure provides a closer look at the Pareto frontier area of the Schaffer function N. 1



Source: Elaborated by the author.

The PC algorithm converged to a central point on the Pareto frontier that has the best cost balance between the two functions.

$$\text{Probability Collectives best value}\begin{cases} x = & 0.9999596333093044 \\ y = & 1.000040367505447 \end{cases}$$

A.3 Constr-Ex problem

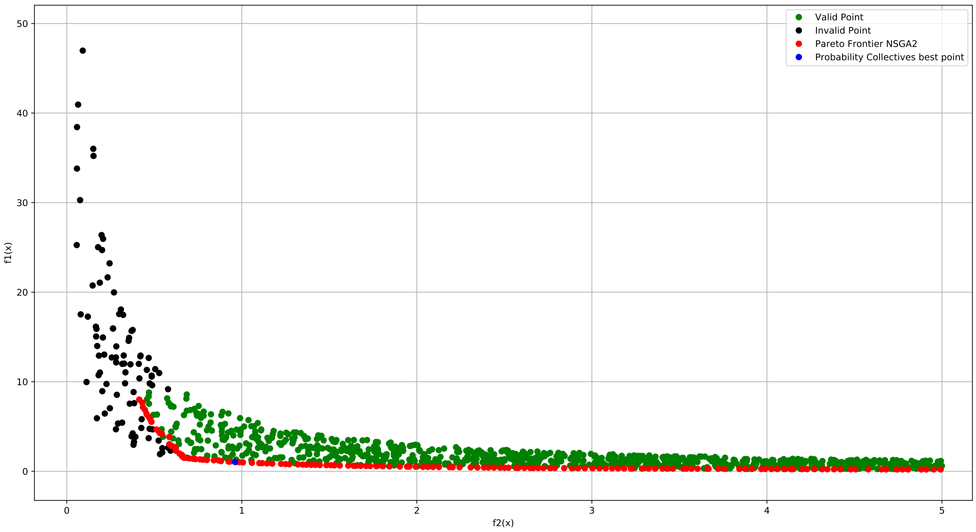
This function was proposed by Han, Ning and Wei (2019).

$$\text{Minimize} = \begin{cases} f_1(x,y) = x \\ f_2(x,y) = (1+y)/x \end{cases}$$

$$\text{Subject to} = \begin{cases} g_1(x,y) = y + 9x \geq 6 \\ g_2(x,y) = -y + 9x \geq 1 \end{cases} \tag{A.3}$$

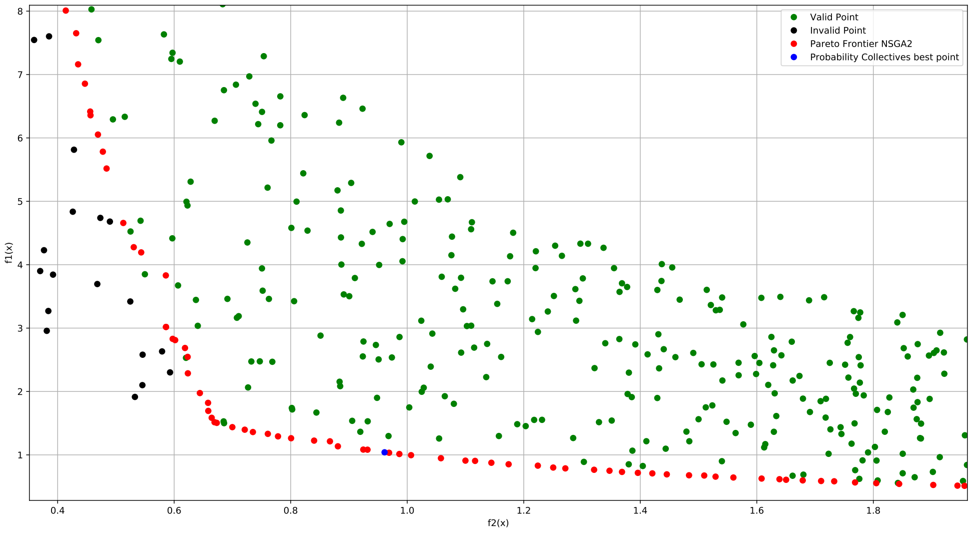
$$\text{Search domain} = \begin{cases} 0.1 \leq x \leq 1 \\ 0 \leq y \leq 5 \end{cases}$$

Figure 62 – The outcome of minimizing the Constr-Ex problem is shown. Black points denote invalid values, while green points indicate valid values that fulfill the constraints but are not optimal. The red points depict the Pareto frontier calculated with NSGA2, and the blue point represents the optimal result of the optimization carried out by the PC.



Source: Elaborated by the author.

Figure 63 – The figure provides a closer look at the Pareto frontier area of the Constr-Ex problem



Source: Elaborated by the author.

In this case, the point was slightly displaced from the center of the Pareto frontier, but the point found by the PC algorithm also has a good balance between costs.

$$\text{Probability Collectives best value} \begin{cases} x = & 0.961140297837971 \\ y = & 1.040432149973398 \end{cases}$$

APPENDIX B

PUBLICATIONS

B.1 Publications related to this thesis.

- T. C. dos Santos, D. R. Bruno, F. S. Osório and D. F. Wolf, "Evaluation of lane-merging approaches for connected vehicles," 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 2019, pp. 1935-1939, doi: 10.1109/IVS.2019.8813802.

- T. C. dos Santos and D. F. Wolf, "Bargaining game approach for lane change maneuvers," 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2019, pp. 629-634, doi: 10.1109/ICAR46387.2019.8981547.

- T. C. dos Santos and D. F. Wolf, "Automated Conflict Resolution of Lane Change Utilizing Probability Collectives," 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2019, pp. 623-628, doi: 10.1109/ICAR46387.2019.8981609.

B.2 Contributions to other works

- D. R. Bruno, T. C. Santos, J. A. R. Silva, D. F. Wolf and F. S. Osório, "Advanced Driver Assistance System Based on Automated Routines for the Benefit of Human Faults Correction in Robotics Vehicles," 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), João Pessoa, Brazil, 2018, pp. 112-117, doi: 10.1109/LARS/SBR/WRE.2018.00029.

- L. A. H. Horita, A. Przewodowski, T. C. d. Santos and F. S. Osório, "Altitude offset constraint for mobile robots' localization," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-6, doi: 10.1109/ITSC45102.2020.9294561.

- SILVA, C. E. ; SANTOS, TIAGO C. ; Gomes, I. P. ; SILVA, JUNIOR A. R. ; WOLF, DENIS F. ; ALVES, R. ; Souza, J. R. . Route Scheduling System for Multiple Self-driving Cars Using K-means and Bio-inspired Algorithms. In: International Conference on Engineering Applications of Neural Networks, 2022, Creta. International Conference on Engineering Applications of Neural Networks, 2022. p. 27-39.

ICMC USP
SÃO CARLOS