Southern Illinois University Carbondale
OpenSIUC

Dissertations | Theses and Dissertations

6-1-2021

Automated Algorithmic Machine-to-Machine Negotiation for Lane Changes Performed by Driverless Vehicles at the Edge of the Internet of Things

Ellie Lovellette
Southern Illinois University Carbondale, elovellette@gmail.com

Follow this and additional works at: https://opensiuc.lib.siu.edu/dissertations

Recommended Citation

Lovellette, Ellie, "Automated Algorithmic Machine-to-Machine Negotiation for Lane Changes Performed by Driverless Vehicles at the Edge of the Internet of Things" (2021). *Dissertations*. 1895.
https://opensiuc.lib.siu.edu/dissertations/1895

This Open Access Dissertation is brought to you for free and open access by the Theses and Dissertations at OpenSIUC. It has been accepted for inclusion in Dissertations by an authorized administrator of OpenSIUC. For more information, please contact opensiuc@lib.siu.edu.

AUTOMATED ALGORITHMIC MACHINE-TO-MACHINE NEGOTIATION FOR LANE CHANGES PERFORMED BY DRIVERLESS VEHICLES AT THE EDGE OF THE INTERNET OF THINGS

by

Ellie Lovellette

M.S., Southern Illinois University Edwardsville, 2011

A Dissertation
Submitted in Partial Fulfillment of the Requirements for the
Doctor of Philosophy Degree

School of Computing
in the Graduate School
Southern Illinois University Carbondale
May 2021

DISSERTATION APPROVAL

AUTOMATED ALGORITHMIC MACHINE-TO-MACHINE NEGOTIATION FOR LANE CHANGES PERFORMED BY DRIVERLESS VEHICLES AT THE EDGE OF THE INTERNET OF THINGS

by

Ellie Lovellette

A Dissertation Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in the field of Computer Science

Approved by:

Dr. Henry Hexmoor, Chair

Dr. Dennis Bouvier, Co-chair

Dr. Chun-Hsi Huang

Dr. Dimitrios Kagaris

Dr. Robert LeAnder

Dr. John Matta

Graduate School
Southern Illinois University Carbondale
March 4, 2021

AN ABSTRACT OF THE DISSERTATION OF

Ellie Lovellette, for the Doctor of Philosophy degree in Computer Science, presented on March 4, 2021, at Southern Illinois University Carbondale.

TITLE: AUTOMATED ALGORITHMIC MACHINE-TO-MACHINE NEGOTIATION FOR LANE CHANGES PERFORMED BY DRIVERLESS VEHICLES AT THE EDGE OF THE INTERNET OF THINGS

MAJOR PROFESSOR: DR. H. HEXMOOR

This dissertation creates and examines algorithmic models for automated machine-to-machine negotiation in localized multi-agent systems at the edge of the Internet of Things. It provides an implementation of two such models for unsupervised resource allocation for the application domain of autonomous vehicle traffic as it pertains to lane changing and speed setting selection.

The first part concerns negotiation via abstract argumentation. A general model for the arbitration of conflict based on abstract argumentation is outlined and then applied to a scenario where autonomous vehicles on a multi-lane highway use expert systems in consultation with private objectives to form arguments and use them to compete for lane positions. The conflict resolution component of the resulting argumentation framework is augmented with social voting to achieve a community supported conflict-free outcome. The presented model heralds a step toward independent negotiation through automated argumentation in distributed multi-agent systems. Many other cyber-physical environments embody stages for opposing positions that may benefit from this type of tool for collaboration.

The second part deals with game-theoretic negotiation through mechanism design. It outlines a mechanism providing resource allocation for a fee and applies it to autonomous vehicle traffic. Vehicular agents apply for speed and lane assignments with

sealed bids containing their private feasible action valuations determined within the context of their governing objective. A truth-inducing mechanism implementing an incentive-compatible strategyproof social choice functions achieves a socially optimal outcome. The model can be adapted to many application fields through the definition of a domain-appropriate operation to be used by the allocation function of the mechanism.

Both presented prototypes conduct operations at the edge of the Internet of Things. They can be applied to agent networks in just about any domain where the sharing of resources is required. The social voting argumentation approach is a minimal but powerful tool facilitating the democratic process when a community makes decisions on the sharing or rationing of common-pool assets. The mechanism design model can create social welfare maximizing allocations for multiple or multidimensional resources.

ACKNOWLEDGMENTS

I would like to extend my sincerest gratitude to my advisor Dr. Henry Hexmoor, for the generosity with his ideas and for letting me set my own pace, and to Dr. Dennis Bouvier for pushing me and reminding me what is at stake.

DEDICATION

To my husband and my mother, who kept asking when I'd be done.

TABLE OF CONTENTS

ABSTRACT i

ACKNOWLEDGMENTS iii

DEDICATION iv

LIST OF TABLES ix

LIST OF FIGURES x

LIST OF ALGORITHMS xii

Introduction 1

1 Cyber-Physical System Agents in the Internet of Things 4

2 Driverless Vehicles 8

2.1 Social Impact of Driverless Vehicles 8

2.1.1 Safety 8

2.1.2 Congestion and Infrastructure Utilization 10

2.1.3 The Environment 10

2.1.4 Driver Health and Time 11

2.1.5 Impact on Other Industries 11

2.2 Technological Evolution 12

2.3 Legal Considerations and Barriers 14

2.4 Deployment 16

2.5 VANET Self-Governance 17

I Automated Multi-Objective Machine-to-Machine Argumentation with Social Voting Among Autonomous Vehicles on a Multi-lane Highway 20

3 Abstract Argumentation for Smart Autonomous Agents in the Internet of Things 21

3.1 Abstract Argumentation 21

3.2 Argumentation Frameworks and Extensions 23

3.3 Argumentation Frameworks and the Acceptability of Arguments 25

3.4 Genesis of Arguments 29
3.4.1 Argument Structure 29
3.4.2 Agent Objectives 32
3.5 Social Abstract Argumentation 34
4 Multi-Objective Argumentation Among Driverless Vehicles with Social Voting . 36
4.1 An Expert System for Lane Selection for Autonomous Vehicles on a Multi-lane Highway 37
4.1.1 Preliminaries 37
4.1.2 A Bird's Eye View 38
4.1.3 The Environment 39
4.1.4 The Density of Agents 40
4.1.5 Formulating Arguments and the Feasibility of Actions 40
4.1.6 Arbitrating Conflict 42
4.1.6.1 The Argumentation Framework 43
4.1.6.2 Objective-Related Voting 44
4.1.6.3 The Semantic Framework and Conflict Resolution 44
4.2 Example Scenario 47
4.3 Empirical Model 53
4.3.1 Environment and Agent Characteristics 53
4.3.1.1 Agent Objectives 54
4.3.1.2 Objective-Based Best Action Derivation 55
4.3.2 Voting and Resolving Ties 59
4.4 Simulation 60
4.4.1 Settings 60
4.4.2 Test Runs 64
4.4.3 Time Complexity 66
4.4.4 Additional Features, Future Extensions and Avenues for Optimization 67

4.5 Conclusion 68

II Automated Multi-Objective Machine-to-Machine Negotiation Among Autonomous Vehicles on a Multi-lane Highway Based on Mechanism Design 70

5 Game-theoretic Mechanism Design for Smart Autonomous Agents in the Internet of Things 71

5.1 Games, Strategies, Payoffs, and Equilibria 71

5.2 Mechanism Design, Dominant Strategies, Strategyproofness, and Incentive-Compatibility 75

6 Game Theoretical Mechanism for Multi-Objective Argumentation Among Driverless Vehicles 78

6.1 Lane and Speed Allocation Mechanism for Driverless Vehicles on a Multi-lane Highway 79

6.1.1 Preliminaries 79

6.1.2 A Bird's Eye View 80

6.1.3 The Environment 81

6.1.4 The Density of Agents 82

6.1.5 The Feasibility of Actions 82

6.1.6 Agents, Preferences, Strategies, and Outcomes 84

6.1.7 Conflict and Conflict-Free Action Allocation 88

6.1.8 Prices and Payouts 90

6.1.9 Lying as a Strategy 91

6.1.10 The Mechanism 92

6.2 Example Scenario 95

6.3 Empirical Model 102

6.3.1 Environment and Agent Characteristics 102

6.3.2 Agent Objectives, Atomic Action Derivation, and Action Ranking . 103

6.4 Simulation 106
6.4.1 Settings 106
6.4.2 Happiness and Patience 107
6.4.3 Bid Reduction to Conflict Subset 109
6.4.4 Test Runs 112
6.4.5 Time Complexity, Optimization, and Scalability 116
6.4.6 Additional Features and Future Extensions 119
6.5 Conclusion 122
7 Applications 124
8 Conclusion 126
REFERENCES 130
VITA 148

LIST OF TABLES

2.1 Automated Driving Levels as Defined by SAE International Standard J3016, June 2018 16

4.1 Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - Determination of Votes for Conflicted Arguments X, Y, and Z 50

4.2 Empirical Model - Lane Characteristics of a Simulated Three-Lane Highway . 54

4.3 Simulation 10,000 Full Cycle Test Runs, Positional Conflict Frequency and Speed Deviation Results 65

4.4 Simulation 10,000 Full Cycle Test Runs, Happiness of Travel Time and Emission Level Agents Results 65

6.1 Mechanism Design for Lane and Speed Allocation Among Driverless Vehicles on a Highway Example - Conflict-Free Allocation Candidates 99

6.2 Empirical Model - Lane Characteristics of Simulated Five-Lane Highway . . . 102

6.3 Simulation 100,000 Full Cycle Test Runs, Conflict-Only Allocation Results . . 113

6.4 Simulation 100,000 Full Cycle Test Runs, Speed and Happiness Results 114

6.5 Simulation 100,000 Full Cycle Test Runs, Results for Happiness and Speed for EL and TT Agents in Mixed and Homogeneous Traffic 116

6.6 Simulation 100,000 Full Cycle Test Runs, Allocation Results in Mixed and Homogeneous Traffic 116

LIST OF FIGURES

3.1 An Illustration of a Toulmin Style Argument 30

3.2 Components of the Toulmin Model of Argumentation 30

3.3 Resulting Argumentation Framework from Intersection Example 31

4.1 The Life Cycle of Argument Formation, Conflict Resolution, and Action Enactment 41

4.2 Feasibility of Lane Changes Example 42

4.3 Social Abstract Argumentation Among Driverless Vehicles on a Highway Example 47

4.4 Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - Attack Graph with Argument Votes 49

4.5 Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - Set of Winning Arguments After Conflict Reduction 51

4.6 Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - NetLogo Simulation Results Round 1 52

4.7 Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - NetLogo Simulation Results Round 2 53

4.8 NetLogo Simulation of Traffic in 11 Lanes One-Way - Vehicles Shown immediately Following Lane Changes in the Argumentation Round 61

4.9 NetLogo Simulation of a 5 Lane Road 64

6.1 Mechanism Design for Lane and Speed Allocation Among Driverless Vehicles on a Highway Example 95

6.2 Mechanism Design for Lane and Speed Allocation Among Driverless Vehicles on a Highway Example - Potential Positions at the End of the Time Slice . . . 98

6.3 Mechanism Design Among Driverless Vehicles on a Highway Example - NetLogo Simulation Results 101

6.4 NetLogo Simulation of a 5 Lane Road 111

6.5 Simulation 100,000 Full Cycle Test Runs, Total Allocations per Traffic Density 113

6.6 Simulation 100,000 Full Cycle Test Runs, Size of Allocations per Traffic Density 118
6.7 NetLogo Simulation Completed 1,000,000 Time Slice Test Run for Traffic Density of 25% 120
6.8 NetLogo Simulation Completed 1,000,000 Time Slice Test Run for Traffic Density of 50% 121
6.9 NetLogo Simulation Completed 1,000,000 Time Slice Test Run for Traffic Density of 75% 122

LIST OF ALGORITHMS

1 Feasible Action Derivation 41

2 Conclusion Derivation Rule Sets for the Personal Travel Time and Global Emission Level Objectives 56

3 Conclusion Derivation Rule Sets for the Local Lane Congestion Objective . 58

4 Conclusion Derivation Rule Sets for All Agents With $Patience \leq 0$ 63

5 Feasible Atomic Action Derivation 104

6 Feasible Action Override for All Agents With $Patience \leq 0$ 109

INTRODUCTION

Following the arc of automation, it is plain to see that the future of ground transportation belongs to autonomous vehicles. As they join the Internet of Things, they will disrupt and reshape the transportation sector forever. Inevitably, self-driving automobiles will eventually largely and possibly completely replace manually operated vehicles on roadways. This burgeoning transportation revolution will affect how and how safely people and goods reach a destination.

Driven by competition, mechanisms for vehicle autonomy are a current hot topic of research. A lot of work is being done to provide basic prerequisites for safely placing an autonomous automobile on the road - to ensure superior sensing capabilities, navigation accuracy, and infallible collision avoidance. However, as the number of these traffic agents increases and is eventually expected to eclipse the number of human-operated vehicles on the road, the need for interoperability mechanisms rises as well. Such sophisticated unsupervised algorithmic components are the topic of this work.

The beginning of this dissertation investigates the upcoming progression of the transportation sector from its current form to its future incarnation fused with the Internet of Things. For background, Chapter 1 provides a brief introduction to fundamental constructs, familiarity with which this work requires, like the Internet of Things, cyber-physical system agents, and edge computing.

Technological progress is often dependent on the concurrent evolution of the underlying legal frameworks governing the field, especially in high-stakes domains like vehicle traffic. For that reason, Chapter 2 briefly examines the current legal and technological constraints in the transportation sector as they pertain to vehicle autonomy. The chapter also lists reasons for the future adoption of self-driving automobiles outside of the evolution of technical capabilities and the subsequent ripple effect adoption could have on industries and population. This chapter outlines the need for automated collaborative fair negotiation mechanisms and suggests algorithmic methods to create systems for real-time arbitration.

The remainder of this work is broken down into two parts exploring two different approaches to machine-to-machine negotiation as they relate to resource sharing. These advanced algorithmic models whose foundations lie in argumentation and game theory find practical applications in transportation allowing autonomous traffic agents to self-organize on the road.

Part I applies lessons learned from online debates to create a mechanism for the democratic assignment of resources in a distributed multi-agent system. Chapter 4 outlines, demonstrates, and analyzes an approach for lane selection for autonomous vehicles on a multi-lane highway rooted in and extending social abstract argumentation the theoretical fundamentals of which are presented in Chapter 3. The lane selection model is built on vehicle expert systems and awards decision making powers to all agents in the local cluster. It is an elegant, minimal, and efficient approach to unsupervised negotiation for portions of a divisible asset (the road). The algorithmic implementation and simulation results show it can provide a self-organizing, persistent, local system facilitating communication and cooperation among autonomous vehicles on a multi-lane highway.

Part II of this work presents an approach, which builds a strategyproof incentive-compatible negotiation model in reverse from a socially beneficial outcome. The outcome maximizes the collective payout and is derived from ranked private bids for lane positions and speed settings submitted by autonomous vehicles on a multi-lane highway. The system's theoretical fundamentals based on mechanism design are laid out in Chapter 5 while the empirical model is outlined, demonstrated, and analyzed in Chapter 6. By ensuring the collective desirability of the equilibrium outcome and the truthfulness of submitted information the system also provides an avenue for monetization. The implementation of the algorithm efficiently prunes the search tree and is shown to be suitable for real-time use in highly dynamic environments. The presented mechanism can work with a single divisible or multiple and/or multidimensional resources.

While the presented models are applied to autonomous vehicle traffic, minor domain-appropriate modifications deliver equivalent algorithmic approaches with diverse practical applications in other Internet of Things areas beyond transportation as addressed in Chapter 7. Chapter 8 summarizes the presented work and discusses further applications and extensions.

CHAPTER 1

CYBER-PHYSICAL SYSTEM AGENTS IN THE INTERNET OF THINGS

As the size and price of electronic components dwindle, manufacturers fortify diverse machines with the ability to join the smart, online networks. These and other advances in technology have produced a myriad of smart devices permeating all contexts of life. Such devices can be found in all enterprise sectors like consumer and home, healthcare and life sciences, retail, hospitality, public safety, military, IT and network, energy, infrastructure, industrial and manufacturing, supply chain, agriculture, education, governance, and transportation. From small personal consumer devices like wearable fitness trackers, smart watches and smart phones, through household items like smart thermostats and refrigerators, through wind turbines and generators, through smart vehicles, to smart grids and smart cities - smart cyber-physical systems are everywhere. These machines are equipped with different capabilities depending on their use and have varying degrees of autonomy. They form the Internet of Things.

The term *Internet of Things* (IoT) was coined at the turn of the century [7] and its definition has continued to evolve with time and technology. Today IoT is used to describe a system where the Internet is connected to the physical world via ubiquitous sensors and actuators. It is the network of interconnected smart devices seeking to automate and improve business processes and various aspects of day-to-day life. The resulting pervasive network is comprised of numerous and heterogeneous machine agents equipped with sensors and the ability to perceive context.

IoT agents can be modeled using the Belief-Desire-Intention model (BDI) [121] where *beliefs* represent the agent's view of the world; the agent's objectives are represented by *desires*, and the actions the agent performs in order to achieve its objectives are *intentions*. An event queue records sensed environment conditions and the agent's beliefs and goals are converted into a set of plans, which prescribe the agent's behavior. The BDI model combines sensed data and action plans to support reasoning and decision making so actions can be taken to satisfy an agent's objectives. The rationality of an agent's behavior in the context of shared resources in a multi-agent system supports collaboration. Agents must have the ability to communicate with one another in order to make decisions on behalf of users or systems and to proactively take action. Due to the heterogeneity of its connected devices, ensuring inter-operability becomes the main concern for the rapidly expanding Internet of Things [129].

Even though the IoT is currently largely vertical [129] and heterogeneous, the expectation is that in the future smart devices will seamlessly network with each other [93] behind the scenes. For that to happen, autonomous machine IoT agents require algorithms for automated negotiation as they interact in an open environment and either have different and possibly conflicting goals or must parlay with each other to form dynamic alliances. Some applications also require decision making on the operational level, for which automation is vital [126].

IoT devices are autonomous machine agents in multi-agent systems. Automated negotiation in multi-agent systems based on argumentation has been found to be superior to other approaches [119]. Various argumentation protocols for negotiation between autonomous agents have been proposed - protocols based on dialogue [5], embedded dialogue [36], plan modification [134], and roles and contexts [73].

Aside from inter-operability, the distributed nature of the IoT, the sheer number of connected devices, and the fact that most smart agents will make decisions and perform actions that directly either positively or negatively affect mostly their locale give rise to issues of scalability and efficiency. An emerging type of computing, called *edge computing* [127], where data are collected, stored and analyzed at the source promises faster, scalable, and more responsive IoT systems. Instead of repeatedly accessing the cloud for analysis of the environment and decision making based on sensed data, smart devices are equipped with some of the capabilities normally delegated to a central authority. No further processing by a human agent or a centralized controller is necessary when agents are able to communicate with each other and have arbitration mechanisms in place to resolve inevitable conflict. The edge computing approach moves intelligence from the cloud to the edge, to smart devices themselves. This enables efficient real-time decision making and with the right negotiation model can mean effective scaled-down arbitration in the affected locality.

In order to take advantage of the benefits of edge computing, smart devices in the IoT have to be equipped with the right physical and algorithmic components and have sufficient computational power to use them. Generally, smart devices, also known as *cyber physical systems* (CPS), are the type of active machine agents able to interact with one another and to proactively make decisions and take action. Each agent is designed to receive sensory data and perform problem solving that produces an output. The output might be a mere perception (e.g., "I am standing in front of an obstacle") or an action to perform (e.g., "go around obstacle"). The problem solving performed by a CPS happens thanks to the embedded *cyber-enabled actors* (CA) - algorithmically controlled mechanisms involving smart networked devices and their decision-making modules [46, 153]. CAs are the proactive components of cyber physical systems. Whereas physical components of CPS (e.g., robots, sensors, and other various devices) are tangible, embodied, and occupy physical space; cyber components are largely intangible, disembodied, and location-independent. In contrast to CPS entities which are viewed as passive objects or things, CAs (also referred to as "agents" or "actors") are active and may behave proactively. As an example, from this perspective, smart vehicles as cyber physical systems are empowered with agent overlays that provide deliberate decision-making capabilities based on actionable intelligence collected from the multitude of smart sensors the vehicles are equipped with.

The world is currently experiencing an inexorable proliferation of pervasive and ubiquitous CAs. Cyber-enabled actors may be embodied agents equipped with sensors and actuators performing automated tasks (e.g., different deployed security systems, robot swarms) or disembodied agents inhabiting parts of the Internet and monitoring and initiating automated actions (e.g. Twitterbots). In isolation, the scope of such CAs may be rather narrow and their functionality could remain restricted to independent actions serving larger systems. Naturally, there is mounting effort to incorporate more advanced cyber-enabled actors into interconnected, complex networks. Invariably, these endeavors bring along the need for mechanisms that monitor and ascertain congruence among disparate actions of a group of CAs thus creating a collaborative environment for actors. Having more advanced smart agents able to sense and collect relevant data about the environment and equipped with information processing and decision making capabilities means that information processing, decision making, and conflict resolution can be successfully moved to the system's edge making use of the advantages of edge-centric computing - exploiting the computational, storage and communication power of modern smart devices while pushing the frontier of computing applications away from a centralized authority and redistributing it to the edge of the network [48]. This will improve scalability, reduce issues of message relay delay, simplify additional required infrastructure and provide a local distributed computing environment that will in turn improve real-time performance [123]. Fast and reliable performance is especially important in rapidly changing environments like vehicle traffic.

The vast populations of CAs will soon dwarf human population. Cyber actors must make decisions on behalf of humans in order to advance the trajectory toward automation. This results in a large number of machine-to-machine interactions and a great volume of data that if effectively wielded can positively contribute to society. Collaborative settings for collections of CAs must be designed. One such particularly interesting high stakes collaborative environment in the IoT is in the transportation sector. To achieve autonomy, especially in high flow transportation where there is no room for error like ground vehicle traffic, the system of negotiation must be highly regulated to meet safety standards.

CHAPTER 2

DRIVERLESS VEHICLES

The proliferation of smart devices has now reached all sectors. In transportation, advanced navigation and driver assist systems are already in place. As for automatic driving, research in that field began in Japan in the late 1970s [139], followed by similar projects in Europe and the United States in the 1980s [94, 35]. Since the 1990s research in autonomous vehicles (AV) has been extensive. Today, automatic driving is a hot topic as the transportation sector is industriously working to advance the technology towards complete autonomy. The enormous potential for profit of the yet unconquered driverless vehicle market drives innovation. Fierce competition creates the underlying sense of urgency to be the first to bring a deployment-ready fully autonomous vehicle to market that will convince the public that it is safe to adopt this new technology. As of December 2020 over 40 corporations were actively and aggressively pursuing vehicle automation [23]. It is evident that in all likelihood, the future of ground transportation largely belongs to self-driving vehicles and most researchers agree that AVs will change the main paradigm of transportation [131]. This change will also have a ripple effect through other industries and society as a whole.

2.1 SOCIAL IMPACT OF DRIVERLESS VEHICLES

Some key areas where autonomous vehicles have the potential to have a positive social impact and useful employment include traffic safety, congestion easing, increased road utilization and decreased parking demand, reduced emissions and fuel consumption, increased mobility, and reduced travel time.

2.1.1 Safety

At the end of the last century, researchers posited that about 90% of all traffic accidents were caused by human failure - through fatigue or inattention at the wheel [132]. This figure, arrived at through a literature survey, has since been repeatedly confirmed by road statistics collected worldwide. A recent study by the U.S. Department of Transportation (USDOT) found that 94% of vehicle accidents nationally happen due to human error [107]. Human motor vehicle operators commit errors due to distracted, drowsy, or otherwise impaired driving. Equally, they cause accidents due to driving too fast for the road conditions or curves, misjudging distance or the speeds of others, making wrong assumptions about the actions of others, performing illegal maneuvers, or overcompensating. Another USDOT report pointed out that over 66% of deaths on U.S. roadways are caused by distracted, drowsy, or drunk driving and speeding [108]. A study from Europe found that even with unimpaired drivers, minor timing, speed, direction, distance, or other human errors may produce major outcomes [136].

According to the World Health Organization there were 1.35 million road traffic deaths globally in 2016 and the number is trending up [109]. In addition, for every death from a road crash there are at least 23 non-fatal injuries requiring hospitalization and emergency room attendance. According to the same report, road traffic injuries are the leading cause of death for children and young adults aged 5-29 years. Even with a United Nations-wide road safety strategy ("A Partnership for Safer Journeys", launched on 28 February 2019) and with the diminished traffic volume caused by the Coronavirus pandemic, the United Nations' Sustainable Development Goal 3.6 [99] adopted in 2016 aiming to halve the number of global deaths and injuries from road traffic accidents by the end of 2020 was not met and its fulfilment target date was postponed by a full decade to 2030 [110], hinting at how insurmountable the problem is thought to be.

It is plausible to assume that most of the lives lost on roadways could be saved by technology as autonomous vehicles would not be plagued by the issues of human drivers that caused the accidents in the first place. AVs have the potential to dramatically reduce the chances of accidents by replacing problematic human behaviors behind the wheel with sophisticated software algorithms for decision making and operation, and sensing superior to human senses. Taking the biggest risk factor, the human operator, out of the control loop should greatly reduce danger on the road.

2.1.2 Congestion and Infrastructure Utilization

Aside from a dramatic impact on safety, a well-regulated autonomous vehicle fleet promises to have a number of other positive side effects on traffic. Machine agents can calculate better routes and will, as a consequence, ease congestion. Often, congestion forms due to the human driver perception-response time. Decreasing that response time by replacing the human driver promises to have a significant positive impact on congestion.

Personal vehicle sharing or fleets of driverless cabs providing transportation or small freight services can diminish the need for individuals to own a personal vehicle, further easing congestion. Traffic conditions and congestion can also be improved by vehicle platooning, which in addition will also save commute time.

As adoption widens, it will become safer to increase speed limits, which will lead to further time savings, and decrease spacing between vehicle distances, which will place more vehicles safely on the road without straining traffic. Platooning and placing vehicles closer together can improve traffic flow at intersections in urban areas by decreasing perception-response time [84].

Driverless vehicles will also have an impact on parking. Parking spaces can be relocated to further locations since the car can drop its occupants off and then park itself, or it can simply return to its original location. Using such parking strategies could potentially eliminate parking demand up to 90% [155]. Efficient structured parking for-a-fee mechanisms can be employed where demand for parking spots remains high.

2.1.3 The Environment

In 2016, the transportation sector became the biggest polluter in the United States [1]. Ground transport in particular has an extensive carbon footprint due to the number of vehicles, their overuse, the age and maintenance of the fleet, and pollution through emissions from burning fossil fuels. The move toward hybrid and fully electric vehicles will have a positive environmental impact through the reduction of emissions and particulate matter from brake systems. Employing electric vehicle technology in AVs will amplify the positive effect. Fewer vehicles on the road because of car sharing or placing vehicles closer together and platooning will also reduce gasoline usage and reduce emission levels. It is estimated that autonomous vehicles have the potential to reduce greenhouse gas emissions by 40–60% [68].

2.1.4 Driver Health and Time

The reduction of greenhouse gasses, particulate matter, nitrogen oxides, carbon monoxide and others through the mass deployment of efficient AVs will improve air quality and have a positive effect on health, especially in big cities. Additionally, drivers not actively operating a car will be able to dedicate the saved commute time to rest or other activities. As a result they will enjoy better health, since commuting 11 or more miles to work has been shown to raise and cause spikes in blood pressure, raise blood sugar and cholesterol levels, increase anxiety, increase the risk of depression, worsen posture and cause backaches, worsen cardiovascular fitness and sleep, and decrease happiness and life satisfaction [65]. Not having to dedicate the commute time to manually operating a vehicle would also allow people to live further away from their place of work if they wish to do so, which would positively affect cities struggling with housing issues.

2.1.5 Impact on Other Industries

Wide adoption of driverless vehicles will also have a ripple effect through industries outside of the transportation sector. A reduction on the demand for parking will affect city planning and infrastructure. A reduction in accidents will lead to a reduction in cost for insurance claims and premiums and will also reduce medical claims related to traffic accidents. Disabled and senior individuals, precluded from driving and relegated to specialty transport services, public transit, and assistance from others, will be able to regain a degree of freedom and mobility they currently do not enjoy. Police departments can redirect efforts currently dedicated to policing roads to more pressing concerns like crime and public safety. Without overly straining the infrastructure more autonomous public transit vehicles can serve communities that do not have adequate access to transportation.

2.2 TECHNOLOGICAL EVOLUTION

The aforementioned positive impacts will have to wait until AVs are technologically ready to be deployed in numbers. The main areas of outstanding reliable technological solutions required for the mass deployment of fully autonomous driverless vehicles are in sensing, data processing, perception, decision making, vehicle control, and in security and vehicle communication [131].

Aside from superior sensing and control capabilities, AVs must be outfitted with sophisticated algorithmic components. These components differ in purpose. Some are used for sensor fusion and data filtering, others for object detection, classification and tracking, others for journey and trajectory planning, yet others for communication. Communication technologies include vehicle-to-vehicle (V2V) and vehicle-to-infrastructure, among others. The architecture of Vehicular Adhoc Networks (VANET) has been expanded to allow long-range (over 1 km) communication allowing the exchange of information among moving vehicles, but, also within the context of an Intelligent Transportation System (ITS), the exchange of information between moving vehicles and fixed infrastructures, pedestrians with personal devices and other local IoT entities [34]. V2V communication specifically is used for real-time safety, other vehicle identification, exchanging trajectory information, spacing, and lane changing.

Lane changing in particular, is an essential but exceedingly challenging task that has been studied from many different angles. A behaviorally sound lane-changing model is not even available for the traditional environment [2]. For connected and autonomous vehicles, examination of the topic is also ongoing. The largest body of research comes from engineering and concentrates on the mechanics of vehicle control during individual lane maneuvers and on lane change behavior and trajectory planning [27, 72, 38, 146, 15, 37, 151, 81, 64, 79, 103, 75, 51]. Other popular areas of interest include verifying the safety of maneuvers [114, 100, 63] and lane change scheduling [117, 148, 9]. These models are mainly concerned with longitudinal and lateral movement mechanics and the safety of the subject vehicle but tend to ignore the effect of lane-changing on neighboring vehicles and on the flow of local traffic. On the algorithmic level, some conceptual collaborative models for lane changes have been proposed. Some of them concentrate on cooperative sensing (the collaborative sharing of situational awareness to facilitate lane changes [90, 31]) or on creating lane assignments based on destination to increase traffic throughput [31, 25]. Others study cooperative control - cooperation with immediately neighboring vehicles in the current and desired lanes to facilitate a lane change request while avoiding collisions and minimizing the effect on following vehicles [10, 149, 102, 101, 82].

Lane-changing decision modelling for both human drivers and smart vehicles is usually rule-based, utility-based, or game theoretical. Work concerning algorithmic lane changing mechanisms for autonomous and connected vehicles stemming from game theory is still sparse [71]. One framework to model information flow and driver behavior in a connected environment uses different acceleration models to capture the underlying dynamics of car-following behavior and evaluates whether a lane change is beneficial through the acceleration of both the subject vehicle and its surrounding vehicles [135]. Another model uses game theory to model mandatory lane changing decisions [2]. An approach formulating lane-changing decision-making as a differential game takes the acceleration of the preceding and following vehicles into consideration to optimize the joint cost and determine a unique and continuous path for the vehicle actuators to track [150]. In a game of incomplete information vehicles use gap selection and interaction though turn signals and lateral moves. Through others' reactions to these cues the model learns in real time to determine optimal timing and acceleration for lane changes [154]. Another approach uses a cooperative game of transferable utility in which when finding gaps, vehicles exchange right-of-way for payment, gaining time in the process [83]. This last model in particular shows that a natural extension rooted in mechanism design can be found, in which lane changes can be monetized and all players are winners. Part II of this work presents such a model. Additionally, part I outlines a model based on abstract argumentation augmented with social voting to deliver a democratic assignment of requested lane changes on a multi-lane road.

It is evident that even for seemingly mundane and straightforward but in reality exceedingly difficult and essential tasks like lane changing, interoperability among vehicles sharing the road is a major concern. Reliable mechanisms must be put into place to facilitate communication, negotiation and cooperation among driverless vehicles following their respective objectives and making decisions on actions that will affect others. Some modern cars are already equipped with advanced driver assistance systems like collision detection and warning, collision avoidance, emergency braking, anti-lock braking, adaptive cruise control, automatic navigation, night vision, blind spot monitoring, automatic parking, driver drowsiness detection, lane centering, lane departure warning, stabilization, traction control, communication systems and a myriad of other sensoring, warning, and assistance systems. Newer mid- to high range vehicle models include more and more of these systems so the industry is indeed rapidly heading towards autonomy. According to the European Commission, driver assistance technology prepares the way for future automation [29]. Even so, driverless vehicles are still kept off the road.

2.3 LEGAL CONSIDERATIONS AND BARRIERS

Complete autonomy of vehicles is still prohibited by the Geneva [141] and Vienna [42] Conventions on Road Traffic. But technological evolution and competition among countries for potential future profits will inevitably create challenges to existing legal constraints. In the United States alone, as of March 2020, 35 states had enacted some form of legislation pertaining to autonomous vehicles [106]. And even though public health concerns stalled legislative actions in 2020, the trend is to try to put regulations in place in anticipation of and in preparation for full automation in the future. The U.S. is in dire need of a federal licensing framework for AVs, including nationally recognized standards for liability, security, and data privacy. Countries like Japan, China and Singapore are currently working on national laws. There is also still a need for the development of a universal safety assessment [124].

Determinations need to be made on how this new industry will integrate with legislation in other areas - motor vehicle operating laws, impaired driving laws, insurance laws, driver licensing and liability, vehicle inspection, traffic rules, AV certification standards, etc. Even though each AV is estimated to provide between $2,000 and close to $5,000 per year in societal benefits [44], a number of outstanding concerns need to be addressed. Some of them are purely technical - like vehicle interoperability especially in environments where driverless cars are mixed with vehicles operated by humans; the reliability of sensors in inclement weather; the ability to recognize human signals in the event of failure of signals the car's sensors recognize; protecting autonomous vehicle software from security breaches (hacking); data collection and privacy; the availability and accuracy of navigation (GPS) data. Legal concerns have been raised - who is at fault in the case of an accident - is it the operator, the software developer, the manufacturer, or the owner; also, how well do vehicles crossing state and county borders adapt to local driving laws that can have small differences but may bring important consequences, like turning right on red? Other concerns are socio-economic - the eroding of driver education and skill levels as technology takes over; effect on employment levels in the transportation sector, the personal injury law sector, and the driver's education sector among others; effects on the gasoline industry and the electrical grid; cost of sufficiently equipped autonomous vehicles; competition among manufacturers and potential cost cutting measures that will keep makers competitive; and most importantly, successful, willful and wide-spread adoption of the technology by humans.

Legal, economic, and ethical considerations remain outside of this work's current scope but it is a virtual certainty that the upcoming changes to the legal framework governing traffic and the rapid advancement of smart vehicle technology will eventually, and probably in the foreseeable future, place autonomous vehicle agents on the road thus forcing the re-legislation of traffic to allow for self-governance.

2.4 DEPLOYMENT

Table 2.1 lists the 5 automated driving levels as defined by the Society of Automotive Engineers (SAE).

Table 2.1: Automated Driving Levels as Defined by SAE International Standard J3016, June 2018

Level 0	No Automation	Zero autonomy, the driver performs all driving tasks
Level 1	Driver Assistance	Vehicle is controlled by the driver, but some driver assist features may be included in the vehicle design
Level 2	Partial Automation	Vehicle has combined automated functions, like acceleration and steering, but the driver must remain engaged with the driving task and monitor the environment at all times
Level 3	Conditional Automation	Driver is necessary, but is not required to monitor the environment. The driver must be ready to take control of the vehicle at all times with notice
Level 4	High Automation	The vehicle is capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle
Level 5	Full Automation	The vehicle is capable of performing all driving functions under all conditions. The driver may have the option to control the vehicle

Many current and new vehicles incorporate level 2 and 3 technologies. Level 4 tech was offered by Tesla's *Autopilot*, which employed automated steering and acceleration in limited conditions but after a fatal crash in 2016 its deployment was delayed. Many other companies are currently testing level 4 vehicles but level 5 autonomy is still out of reach pending numerous technological and algorithmic improvements and many years for development and testing.

The current predictions [86] for market penetration are that assuming level 5 vehicles become commercially available in the 2030s, their initial price point and limitations will keep them at a maximum of 2% of the vehicle fleet. As the technology improves, consumer confidence increases, and prices gradually drop, by 2045, level 5 AVs could account for half of new automobile sales and 40% of vehicle travel. Without mandates, cost and consumer preferences could mean that market saturation would take several decades. Thus, most impacts like the reduction of traffic and parking congestion, independent mobility, improved safety, and pollution reduction, are projected to only become significant when autonomous vehicles become common and affordable, probably in the 2050s to 2060s. It is estimated that by the 2060s, up to 80% of travel could be in AVs [86]. Full market saturation means that the vehicle fleet will govern itself.

2.5 VANET SELF-GOVERNANCE

Level 5 autonomous vehicles entering traffic in the near future underscores the need to create infallible arbitration mechanisms for negotiation among them. Vehicular AdHoc Networks (VANET) will use these mechanisms to communicate and self-organize based on common safety, optimal road usage, individual and shared objectives, availability and utilization of resources, as well as social capital and utility. Regimented versions of this are found in cooperative adaptive cruise control [128, 61] where autonomous vehicles employ distance measurements and communication with other traffic participants to determine parameters for safe vehicle following and lane changes, and in platooning techniques [74, 62] where autonomous vehicles follow a lead car and join and leave a traffic platoon (convoy) at will.

Reliable self-governance in a multi-agent system housed by the IoT starts with the agents themselves. Due to their nature, autonomous vehicular agents will require systems for fast computation and decision making based on sensed data, as well as short and efficient communication pathways with other agents they are sharing the road with or are connected to through their local Intelligent Transportation System (ITS). In order to make collaboration more efficient, communication and decision making authority can be delegated to the local vehicle clusters themselves instead of relying on a central authority. In other words, the edge of the IoT network containing the local VANET partition will self-organize.

Nevertheless, this collaboration is not seamless and devoid of conflict. Collectively, smart vehicles (i.e., driverless and cyber-enabled) are a class of cyber actors in a high-stakes arena that exhibit potential compatibility issues among decisions in cases of e.g., lane selection, platooning, cruising speed control, right of way determination, parking assignment etc. Individually, a fully autonomous level 5 AV will decide on its preferred actions and strive to implement them. A single agent's behavior however, may have direct impact on its closest peers and conflict can arise. Agents may then have to edit their intended behavior in favor of e.g., the collective good. Reminiscent of Isaac Asimov's Three Laws of Robotics [8], to assure the safety and predictability of agent behavior, there is a need for the creation of systems of rules and policies that embody desired objectives and govern all AV actions. It is conceivable that there will be decisions made by agents that will be unpopular, at odds with human needs, or rife with ethical issues. For instance, a driverless vehicle must determine the best course of action when confronted with an unavoidable collision [49] or an AV may have to abandon its claim to a lane position if that would greatly impede remaining traffic.

Algorithmically, in order to codify negotiation between agents, a driverless vehicle traffic system needs to be represented with the appropriate model. A game theoretic model can be used to implement such a system. As a game strategy, the action plan a vehicle follows may be competitive or cooperative. In the case of competition on the road, e.g., fastest possible delivery of goods, leading to self-serving adversarial behavior of a vehicle or platoon, the model can mimic a zero-sum game. Thus other agents in traffic may seek to limit the adverse actions taken by an agent or a group of agents because of competition or in service of the greater good. In a non-zero sum game setting, a more general representation can be posited with a social welfare maximizing game theoretic model. In such games of coordination, strategy profile selection by players yields synergistic effects [60]. Coordination, cooperation, and negotiation improve payoffs for all or most players as the model strives to maximize social welfare or reach a democratic decision. Inevitably, this will put vehicles at odds with one another and create a framework where bids/arguments are continuously exchanged and arbitrated.

Since advanced autonomous smart agents' need to proactively take action will, as a matter of course, lead to conflicts with other agents in the proximity, arbitration mechanisms for self-governance must offer conflict resolution. One technique that offers itself to this scenario is automated argumentation. It is a multi-agent system approach that views a pool of intended actions of heterogeneous agents as a collection of abstract arguments. The model then uses these arguments to identify possible congruence concerns and to arrive at a satisfactory resolution of conflicts as viewed from the perspectives of all participating agents. Another approach for arbitration suited to this scenario is mechanism design. It is a game theoretic model in which a game is constructed backwards from the desired social utility maximizing outcome. The mechanism accepts ranked bids from all local agents, arbitrates conflict in favor of a common objective, and delivers a resource allocation that is beneficial for the group as a whole. This dissertation presents the theoretic groundwork and prototyped implementation for these two self-governance models within the application domain of driverless vehicles.

Part I

Automated Multi-Objective Machine-to-Machine Argumentation with Social Voting Among Autonomous Vehicles on a Multi-lane Highway

CHAPTER 3

ABSTRACT ARGUMENTATION FOR SMART AUTONOMOUS AGENTS IN THE INTERNET OF THINGS

The Internet of Things is a ubiquitous multi-agent system so it is a near certainty that smart objects will have other smart devices in their vicinity and often they will have to interact with them. Each autonomous machine agent uses sensed data from its environment and interprets it in the context of objectives it maintains and their prioritization. This proactive interpretation is a basis for problem solving the result of which is a perception or a preferred action for the agent to perform. When sharing resources, selected preferred actions of different smart agents will inevitably come into conflict and in a multi-agent system conflict has to be resolved in order to avoid undesirable consequences. So even though decisions can be made in isolation, actions cannot always be undertaken until an arbitration mechanism has been engaged. This negotiation process has to be automated and unsupervised to enable agent autonomy as agents interact in an open environment and either have different and possibly conflicting goals or must parlay with each other to form dynamic alliances. Some applications also require decision making on the operational level, for which automation is vital [126].

One way to ensure devices successfully communicate and negotiate with each other to achieve individual or common goals is through argumentation. Various argumentation protocols for negotiation between autonomous agents have been proposed - protocols based on dialogue [5], embedded dialogue [36], plan modification [134], and roles and contexts [73]. Automated negotiation in multi-agent systems based on argumentation has been found to be superior to other approaches [119].

3.1 ABSTRACT ARGUMENTATION

Argumentation is the process in which agents construct, exchange and evaluate interacting and possibly conflicting arguments. These arguments contain sets of premises, corresponding methods of reasoning, and the culminations of conclusions drawn from logical reasoning. When the arguing group is made up of humans, human reasoning leads the process. While negotiation and reasoning among humans have been an inspiration for modeling automated argumentation within groups consisting of machines, customarily human involvement brings underlying politics, implicit agendas, and possible behavior-altering strategic incentives. With those taken out of the equation, cyber-agent arguments are modeled as atomic proclamations, the general idea for which was posited in earlier work [87]. Thus, human level reasoning can be transformed into computational argumentation by abstracting arguments and the relations between them. These abstract argumentation frameworks provide the transition between human argumentation and machine-to-machine (M2M) multi-agent negotiation.

When the internal structure of arguments is disregarded and what is instead taken into account is the relationships between them, argumentation becomes abstract. The foundations were laid by Birnbaum et al. attempting to model the human argumentation process in a computer program with the help of argument graphs [17, 16] but argumentation is widely credited to Dung who in his seminal paper [39] presents a structured framework for the argumentation process, equating it to a special form of logic programming.

Argumentation is one of the oldest research foci and a pervasive topic in various fields. It is a rich and varied discipline which has found wide application in Artificial Intelligence [12, 130]. Argumentation research stemming from the AI field has delivered argumentation framework extensions, dialogue action languages, argument schemes, computational treatments, and established argumentation services. In Philosophy, Toulmins's initially rejected work [138] regarding the structure of arguments has produced the now widely accepted Toulmin Model. More recently, argumentation research from the field of Philosophy has introduced Pollock's inference-graphs and degrees of justification for arguments [76]. Argumentation has also been adapted to many other domains including computational law [40] - to model legal argument in the presence of prevarication tactics employed by an agent acting in a self-serving non-cooperative manner; in machine learning [95], and in multi-agent negotiations [45] - to represent multi-issue deliberations. Argument mining for machine learning and argumentation is currently a topic drawing much interest from a diverse group of research communities [85].

Argumentation can find application in the Internet of Things especially at the edge of the network, where it can facilitate and optimize autonomous negotiation among smart devices. Smart vehicle technology in particular can benefit from such regimented unsupervised negotiation. A general approach can be posited with a game theoretic model to arbitrate among differing positions and promote outcomes achieved through a democratic process. Inevitably, this will pit vehicles at odds and create a automated argumentation framework.

3.2 ARGUMENTATION FRAMEWORKS AND EXTENSIONS

Invariably, independent decision making produces conflict as differing arguments are pit against one another. Conflicts between arguments are designated as "attacks" in abstract argumentation. Argumentation's main objective is to collect contradictions into a common pool and reach a desirable configuration (called an extension) through a process of arbitration.

Dung style argumentation [39] is a well-known model for the abstract argumentation process. An argumentation framework consists of a set of abstract interacting arguments lacking internal structure or specific interpretation, a set of attacks (i.e., contradictions, conflicts) between them, and semantics for evaluating these arguments.

Dung formally defines the admissibility of arguments as one of three possible absolute statuses - *accepted*, *rejected*, and *undecided*. A single attack on an argument is sufficient to automatically retract it. However, this approach does not migrate well for scenarios where arguments are not equal. In most cases, an argument will at least weaken a conflicting argument but will not necessarily negate it completely.

Extensions of Dung's original argumentation frameworks have been introduced to address the lack of levels of relative strength and acceptability of arguments outside of the attack/defend relations and accepted/rejected/undecided status. Acceptability ranks for arguments are introduced by ranking-based semantics [3, 4, 116]. The identity of an argument is irrelevant in the determination of its ranking, only the attacks between it and other arguments count. Thus, an argument can be attacked multiple times by others and is no longer removed, only downgraded in acceptability - the higher the rank of the attacking argument, the greater the downgrade. Defenders of arguments - attacking their attackers - have the opposite effect on the degree of acceptability. In this approach, the set of semantics transforms the argumentation graph of the framework into a ranking on its set of arguments: from the most accepted to the weakest. Further refinements include the ability to, depending on the decision-making situation and context, give dominance to the cardinality or quality of attackers, or assign priorities to arguments and use the resulting partial order to resolve conflicts.

An extension based on game-theory [92] models the argumentation framework as a repeated two-person zero-sum game. Recursive computation and the Minimax theorem determine the weight of an argument by taking into account its attackers and defenders. Bipolar Argumentation Frameworks [22] introduce the notion of "graduality" of argument acceptability by adding a valuation metric that assigns each argument a measure of its value beyond Dung's acceptability. The valuation is based on the value of direct attackers or the type of labeling an argument receives depending on the length of the branch it is in. Attacks can be assigned weights too and they can be used to derive defense, acting as a de-facto preference relation [30]. Attacks having weights allows conflicting arguments to co-exist and to tolerate attacks with total weight below a certain threshold. In these weighted argumentation frameworks [41] the model can set the level of acceptance of some contradiction. Sophisticated argumentation models therefore could profit from the ability to separately or simultaneously attach weights to arguments and attack relations, where appropriate, thus allowing for a graduality in the acceptability of arguments.

Under certain conditions, arguments need to be temporarily excluded from the framework but not be given zero weight permanently. Dynamic argumentation frameworks [125] introduce evidence as a deciding factor for which arguments are currently active, and thus valid, and which arguments are not. As premises in argumentation may change, sets of arguments providing evidence to satisfy the premise are activated and deactivated as appropriate.

When the value of arguments is not sufficiently determined by the structure of the framework they are in, external valuation metrics can be introduced to augment the system. Social Abstract Argumentation (SAA) [43, 80] is an extension where the framework is supplemented with social voting. In SAA anyone can cast a "vote" for arguments in the argumentation pool. Votes are either for or against the argument and the level of its social support is calculated as the difference between votes supporting and opposing it.

Existing extensions have shown that acceptability of arguments can be gradual and ranked based on their attack/defend relations or the arguments themselves. The addition of external metrics can augment the framework in meaningful ways. Borrowing from game theory and modelling negotiation as a multi-round game can create a continuous process from a series of discrete rounds. Depending on the scenario and application domain, the choice of an appropriate argumentation framework extension can greatly enhance a model.

3.3 ARGUMENTATION FRAMEWORKS AND THE ACCEPTABILITY OF ARGUMENTS

Arguments are the building blocks of argumentation. They contain sets of premises, methods for logical reasoning, and conclusions but their internal structure is irrelevant. What is important are the relationships between them as relationships give rise to conflict. Resolving this conflict is achieved though the abstract argumentation process with the help of argumentation frameworks (AF). AFs are represented using binary graphs in which the nodes are arguments and the edges are attacks between pairs of arguments. The edges of the graph are directed arcs indicating that one argument attacks an incident node. Pollock's inference graphs [76] are very similar to graphs produced by depicting Dung's attack relationships. Mathematical particulars concerning the argumentation process are outlined in this section. The Dung model [39] is outlined in definition I.1 and definitions I.2-I.4 list some of the main properties for a set of arguments.

Definition I.1. An *argumentation framework* $F = \langle A, R \rangle$ consists of a set of arguments, A, and a set of binary relations (attacks), R, between them, where A is finite and R is finite and $R \subset A \times A$. A relation between two arguments a_i and a_j is denoted as a binary attack relation (a_i, a_j) such that argument a_i attacks argument a_j, further denoted as $(a_i \curvearrowright a_j)$.

Definition I.2. An argument $a_i \in A$ is *acceptable* w.r.t. a set of arguments S iff for $\forall a_j \in A$: if a_j attacks a_i, then a_j is attacked by S, i.e. $\exists s \in S$ s.t. s attacks a_j.

In essence, to a rational agent an argument (or a set of arguments) is acceptable if the agent can defend it against all attacks. This is assured by having arguments in the set complement protect arguments in the set by attacking possible offending arguments. This is a rather common phenomenon in society. This is how in-groups emerge [115]. An in-group holds steadfast to a set of arguments it finds acceptable and repels others.

Definition I.3. A set of arguments S is *conflict-free* if $\nexists a_i, a_j \in S$ s.t. a_i attacks a_j or a_j attacks a_i.

Definition I.3 posits that there are no attack relations among any pairs of arguments in a conflict-free set. This is less common than the acceptability property but in-groups exhibit this phenomenon as well.

Definition I.4. A conflict-free set of arguments S is *admissible* iff each argument in S is acceptable w.r.t S. The empty set is admissible.

This means that a rational agent maintains a set of arguments that is free from internal contradictions and that can be defended from within against all attacks on it.

Dung formally defines semantics for the acceptability of arguments. These semantics allow the computation of sets of arguments within an argumentation system called *extensions*. The idea behind extensions is to identify sets of arguments that can survive conflict together and collectively represent a reasonable position in a negotiation. An extension can be thought of as a belief in something that can be defended with the arguments in the extension. Thus, extension-based argumentation semantics are used to select sets of arguments according to a predetermined criterion. Definitions I.5-I.8 outline basic extension types.

Definition I.5. A conflict-free set of arguments S is a *stable extension* iff S attacks every argument outside of S.

The stable extension property appears to identify a xenophobic tendency that is an unreasonable fear or hatred of the unfamiliar. This rather strange notion of stability is traced to the long standing Arab-Israeli conflict example elaborated in [16] and draws parallels to the Hatfield-McCoy feud (1863-1891), the account of American folklore that has become a byword for bitterly feuding rival parties in general. The existence of a stable extension in an argumentation framework is not guaranteed.

Definition I.6. A *preferred extension* of an argumentation framework is its maximal admissible set w.r.t. set inclusion.

It can be shown that for each admissible set in an argumentation framework there exists a preferred extension that is its superset. Since the empty set is always admissible, it follows that every argumentation framework possesses at least one preferred extension. Furthermore, every stable extension is a preferred extension but the opposite does not hold true. Preferred extensions represent *credulous semantics* in an argumentation framework - a rational agent's readiness to accept a number of individual beliefs (sets of arguments or hypotheses) each of which does not pose internal conflict and that the agent can defend from within the given set. In this case the agent runs the risk of accepting arguments that belong to different internalized beliefs but may not be acceptable together. A preferred extension is maximal and the agent cannot add any more arguments to it without causing internal conflict.

Definition I.7. A *complete extension* of an argumentation framework is an admissible set S s.t. each argument acceptable w.r.t. S belongs to S.

The complete extension represents a rational agent that believes and internalizes everything it can defend. Every preferred extension is complete (the opposite does not hold) so every argumentation framework will have a a stable extension even if it is the empty set. An agent can maintain more than one complete extension, each corresponding to a different internally contradiction-free viewpoint that can defend itself against attacks.

Definition I.8. A set is a (unique) *grounded extension* if it is the smallest element w.r.t. set inclusion among the complete extensions of an argumentation framework.

It is obvious that every grounded extension is complete and an argumentation framework will always have a unique grounded extension that can be obtained iteratively. A grounded extension starts with all arguments that are attacked by no one and then arguments that are defended directly or indirectly by these unattacked arguments are added. An agent can maintain multiple preferred extensions and the grounded extension is contained in all of them. Grounded extensions represent *skeptical semantics*. A rational agent will only accept an argument if it is compatible with all internalized beliefs (individual sets of arguments or hypotheses represented by extensions). In this case the agent risks losing information by over-restricting the requirements for acceptability of arguments, since beliefs may only intersect minimally or not at all.

Similarly to status assignment in Pollock's inference graphs [113], the status of an argument can be decided using a labeling approach where each argument in the argumentation framework receives a label from a predefined set. The most widely used labeling system is to use a set of three labels $L = \{in, out, undec\}$ [20, 19, 120] but there are alternative approaches [70]. An *in* label signifies that the argument is accepted, an *out* label signals the argument's rejection; and *undec* is assigned when no determination was made, i.e., the status of the argument as related to its acceptability is undecided. The three value labeling can be used to signify an argument's membership in an extension. For a conflict-free extension, arguments in the extension are labeled as *in*, arguments attacked by the extension are *out* and arguments that are neither in the extension, nor are attacked by it are labeled *undec* [11].

3.4 GENESIS OF ARGUMENTS

To take part in negotiation with others an agent must be capable of generating arguments. Once generated, the agent must decide which argument(s) to put forth to use in the argumentation process. Arguments cast by agents are gathered in a common pool. The pool is then used to build an argumentation framework. The resolution component of the argumentation system evaluates attack relations between arguments and finds a conflict-free set. But how are arguments formed in the first place?

3.4.1 Argument Structure

In order to make use of the advantages of edge computing, each agent is expected to perform its own independent reasoning and problem solving that can be approximately modeled as an expert system (ES). In general, an ES can be used for generating inferences as well as for postmortem analysis of conclusions. For the purposes of the work presented here, the focus is on how to generate inferences as arguments and on how to arbitrate among them.

Agents fuse one or more pieces of sensory data to determine an input for reasoning. In the context of smart Internet of Things devices, an agent's first task is to use its ES to periodically generates arguments [69] as atomic abstract entities comprised of sensed data and an output. Similar to the BDI model [121] introduced in Section 1, the internal structure of an argument is modeled as ⟨*sensed condition, then warrant; therefore, recommended action*⟩ as illustrated in Figure 3.1. Arguments built this way follow the Toulmin model of argumentation [138] shown in Figure 3.2.

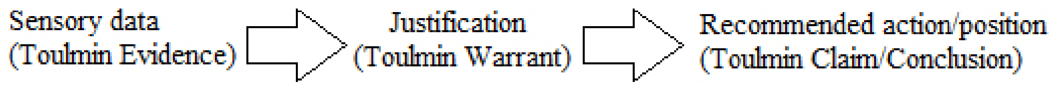

Figure 3.1: An Illustration of a Toulmin Style Argument

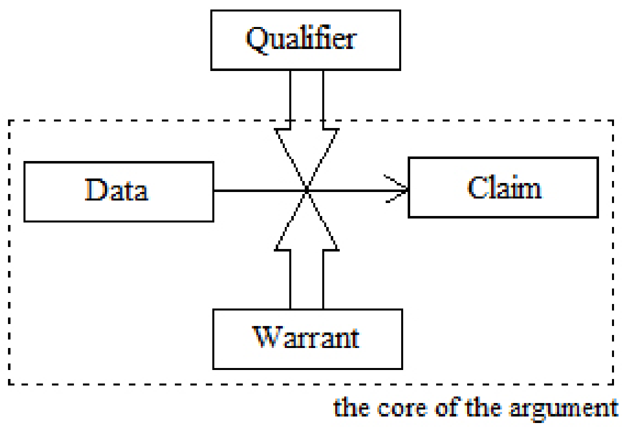

Figure 3.2: Components of the Toulmin Model of Argumentation

For an arbitrary domain, an agent's expert system is typically structured as a set of rules partitioned into subsets that are either in the form ⟨**if** condition-x **then** conclusion-y⟩ for situation assessment purposes or in the form ⟨**if** conclusion-y **then do** action-k⟩ for action selection purposes. A "condition-x" is a combination of conditions sensed in the environment (Toulmin Evidence) while a "conclusion-y" is a perception. An "action-k" is an action to be performed. An inference engine (IF) is the component of the expert system that gathers conditions matching the "condition-x" part of rules. The IF then fires the applicable rules and determines a winning "conclusion-y" (Toulmin Warrant). Inference works on all applicable action rules that arbitrate among competing actions to determine a winning "action-k" (Toulmin Conclusion) to be executed. Running inference on the expert system yields argument components.

Example I.1. *Argument Structure and Acceptability Example*

Consider the following three arguments where warrants are produced as a result of using a model that designates right of way for self-driving vehicles at an intersection:

- a_1 = I am an ambulance; my emergency signal is on; therefore I should go next.
- a_2 = I am a car amid a funeral procession; funeral processions should not be interrupted; therefore I should go next.
- a_3 = I am a police car; there is an emergency vehicle with its signal on; therefore the emergency vehicle should go next.

Irrespective of the theme and content of arguments, in their abstraction the first two arguments are mutually attacking but may be designated with different weights. For example, in argument a_2 the warrant is to avoid interrupting a funeral procession. However, that claim is not as strong in the presence of an emergency vehicle. Thus the weight of argument a_2 may differ from that of a_1. In addition, argument a_3 weakens argument a_2, and by attacking a_2, a_3 defends a_1. The resulting argumentation framework comprised of the argument pool and changing conditions is the basis for autonomous agent problem solving.

$$
\begin{tikzcd}
a_1 \arrow[r, bend left] & a_2 \arrow[l, bend left] & a_3 \arrow[l, bend right]
\end{tikzcd}
$$

Figure 3.3: Resulting Argumentation Framework from Intersection Example

The presented example results in an argumentation framework $F = \langle A, R \rangle$ where $A = \{a_1, a_2, a_3\}$ and $R = \{(a_1 \curvearrowright a_2), (a_2 \curvearrowright a_1), (a_3 \curvearrowright a_2)\}$ as shown in Figure 3.3. It is easy to see that there are two non-empty conflict-free sets - $\{a_2\}$ and $\{a_1, a_3\}$ and the set $\{a_1, a_3\}$ is admissible since it can defend itself from the attack by a_2.

Thus $E = \{a_1, a_3\}$ is (the only) preferred extension, which also makes it complete and grounded. And since a_2 is the only remaining argument outside of E and there is an attack relation between it and set, then E is also a stable extension.

With arguments created by the IF the agent must then select the best one(s) to cast into the argument pool to compete with other agents' arguments. Should the scenario call for a single argument to be played, the ES performs the selection of the best argument in consultation with the agent's current governing objective.

3.4.2 Agent Objectives

It can be assumed that all machine agents are rational in the sense that they have one or more objectives, selfish or altruistic ones, which they strive to satisfy. Agents with similar objectives may form coalitions and bring forth arguments that strengthen their shared position. When objectives of agents compete, the argument pool becomes more adversarial.

In Example I.1 the contradictions among arguments are entirely within the same objective context (cross the intersection) as arguments arise from reasoning with an expert system that is bound to an objective. Occasionally, the same arguments may arise in multiple objective contexts. Less commonly, attacks between a pair of arguments in one objective framework may reoccur in another. However, attacks between arguments that belong to different objective contexts do not exist.

A set of objectives can create inconsistencies for argumentation. Since an agent's objectives are often a compatible set for the welfare of the agent, it might be possible to prioritize among them. Therefore, the winning arguments from competing objectives would inherit priorities of their corresponding objective. Otherwise, we can use multi-objective techniques such as the multi-objective particle swarm [28]. Of course, an agent can select a single governing objective and just make decisions within its guidelines.

As previously illustrated, arguments are byproducts of reasoning as in expert systems [69]. The ES of an agent is able to infer and analyze context sensed from the environment and generate possible actions. Feasible actions pertaining to a specific objective can be contrasted in the context of the objective. The relative contribution of an action toward an objective can be used to give it a relative higher precedence over lower valued actions. An agent's objectives themselves will have different priorities, which will help decide the best next action to take. This quantitative valuation helps the ES determine the optimal action to take within the context of an objective and across objectives as shown in Example I.2.

Example I.2. *Agent Objectives Example*

Consider a driverless delivery vehicle which has three objectives and is travelling on a local two-lane road:

- o_1 = Arrive at venue X in the fastest time possible.
- o_2 = Safely deliver the wedding cake to venue X.
- o_3 = Pick up 3 bags of confectioners sugar.

In this case objective o_2 will clearly take precedence over o_1, and both of them dominate o_3. After all, delivering a wedding cake intact is more important that not being late but potentially ruining the cake; and supplies can be picked up after the successful delivery.

Assume that during its travels the autonomous delivery vehicle finds itself behind a slower moving automobile and there is oncoming traffic a short distance away. The agent then considers two potential actions. It can overtake the vehicle but due to the proximity of oncoming traffic that maneuver would require the agent to make a fast change into the opposite traffic lane, rapidly accelerate by 15 miles per hour and then quickly move back into its lane. Alternatively, the agent can wait out approaching vehicles and then make a more leisurely overtake maneuver. Both actions can be considered within the context of both objectives o_1 and o_2. The former action would be the front-runner to satisfy objective o_1 and the latter would be the best action chosen within the context of o_2. Since the second objective is the dominating goal, the vehicle will elect to maintain speed and perform overtaking maneuvers later when it can do so without risking the safety of the cargo.

Having determined the next best action for itself in accordance with its objective(s), the agent plays one or more arguments which are used to support its right to implement the action. Arguments thus guide an agent's activity in the world. Once generated, each argument can be labeled as an atomic entity. Agents then cast their arguments into the argumentation pool where they are maintained indexed by the environmental contexts of their genesis primarily identified by the objective of the agent that created them. The competition among arguments is only meaningful when they pertain to a common purpose. As a whole, in the pool, there will be a large number of arguments and since they are generated by different agents, inconsistencies and contradictions are unavoidable. The process of argumentation is largely a process of identifying a group of arguments that are the most compatible for producing an objective. This is achieved by resolving conflict within the argumentation framework by using the provided semantics to get to a desirable extension. Sometimes the semantics of the AF can be augmented by external measures.

3.5 SOCIAL ABSTRACT ARGUMENTATION

Social Abstract Argumentation (SAA) [43, 80] is an external valuation metric which extends the classical Dung model with the inclusion of social voting. It was inspired by the chaos of social networks and the lack of well-structured debate and interaction between users seeking more meaningful exchange online. Unlike the original Dung approach where arguments are either accepted or rejected outright, SAA achieves degrees of acceptability of arguments by augmenting acceptability information gained from the argumentation framework with an additional social support metric.

In its simplest form SAA allows a group of participants and spectators to influence the outcome of a debate by casting “for” and “against” votes. Votes can be cast for and against both arguments and attacks. The model can also be extended for more elaborate voting. Since social acceptance of an argument in the form of votes represents popular opinion but most issues do not have universal support, SAA allows for the alteration of argument strength beyond the original absolute accepted/rejected status.

There is no reason why SAA cannot be adapted for and utilized with groups of machine agents in the IoT. It can allow sufficiently equipped autonomous machines with sensing and computational capabilities to self-organize and self-arbitrate either in a completely distributed manner or by involving an arbitration agent at the edge. By adding voting to the argumentation framework, agents who authored arguments with a high degree of social support can be authorized to take the actions they competed for the right to implement by casting these arguments. The remainder of agents can be assigned default actions. Thus, properly modified SAA can provide a means for the unsupervised democratic resolution of conflict within the argumentation framework. When this discrete process is repeated at regular time intervals, much like a multi-round game, it can govern a continuous system without requiring any human involvement.

A distributed multi-round SAA enhanced argumentation framework prototype for the AV domain, which provides a vote gathering function and a semantic framework that is able to evaluate votes, calculate social support, resolve ties, reduce the conflict set and determine winners and losers is presented next.

CHAPTER 4

MULTI-OBJECTIVE ARGUMENTATION AMONG DRIVERLESS VEHICLES WITH SOCIAL VOTING

The model outlined in this chapter is structured as a multi-round game governed by an argumentation mechanism. Each autonomous vehicle on the road only has knowledge of its own objective(s) according to which it adjusts its travel behavior. Clearly, when sharing the road within a community, vehicles cannot make maneuver decisions in isolation since the resulting actions will affect others. Thus, in every round of the game, an agent planning to implement intended actions like lane changes must seek approval to do so from its local peer cluster. The community potentially affected by these actions collectively decides whether to approve or prohibit them. An agent's planned actions may be in conflict with the desired actions of others. Achieving a democratic resolution of conflict present in the system for the game round can be facilitated by the use of social voting where each participant in the affected area casts votes and influences the decisions of others. A vote is a strategy played by the agent casting it. The vote is chosen with the help of an expert system and an inference engine in accordance with current conditions and the set of objectives the agent maintains. The strategy can be individual and selfish or collaborative and selected in view of a coalition. Either way, when used with the proper argumentation mechanism, voting facilitates the conflict-free implementation of actions and enables the collaborative use of sharable resources. Social Abstract Argumentation makes a community-driven driverless vehicle lane change mechanism possible. The presented model was implemented and verified in NetLogo.

4.1 AN EXPERT SYSTEM FOR LANE SELECTION FOR AUTONOMOUS VEHICLES ON A MULTI-LANE HIGHWAY

4.1.1 Preliminaries

To delineate the general case of application of the presented argumentation model for lane selection, it is assumed that every AV utilizing the portion of the road governed by the system has been granted access to said road in exchange for its agreement to participate in the argumentation process and to abide by its decisions. Extraneous players like priority agents or non-autonomous vehicles are assumed to be absent. While the algorithm can be extended to accommodate such agents, this is currently out of scope for this work.

There is no consideration given to the propagation of information; how vehicles build and maintain network links and how long it takes to transfer arguments and votes along those links is irrelevant for this work. Physical constraints will limit the local network size. Keeping the agent cluster small is beneficial for scalability and is appropriate since effects of lane change maneuvers are limited to agents in the immediate proximity. Also, consensus on public data like spatial positions, speed and vehicle characteristics is implied, as it can be easily verified and as dishonesty about situational awareness can be penalized by the system, agents have no incentives to try to falsify this data.

For any degree of autonomy, driverless vehicles have to be outfitted with advanced sensors and enough computational power to quickly evaluate their environment and react to it. To support the proposed lane selection argumentation mechanism an agent will only need to be additionally equipped with the expert system algorithm. The argumentation process is modeled as a multi-round game designed to be performed in a fully distributed manner, as arguments are public. However, the process can also be modified to employ the help of an arbiter. In the case where an arbiter is present (e.g., a call tower with coverage over the portion of road) or designated (a vehicle in traffic), non-arbiter agents will only use the inference engine component of the expert system, while the arbiter will run the conflict resolution component and communicate the results to the players. Regardless of the mode of arbitration, the system keeps interaction, conflict identification and resolution to the edge of the IoT and distributes and localizes it within a multi-agent subset of manageable size. Working within a limited group of agents improves communication reliability and speed, while performing sensing, computation, and decision-making at the source improves real-time performance, which is paramount for intensely dynamic high-stakes environments like vehicle traffic.

4.1.2 A Bird's Eye View

In the presented model autonomous vehicle (AV) agents with private travel objectives share a multi-lane road. The road is partitioned in segments, which include a finite number of vehicles. The process unfolds in rounds.

During every round, each AV agent makes a decision on actions pertaining to lane selection that, when taken, will advance its main objective (although vehicles may maintain an arbitrarily large number of objectives, for simplicity here one main objective is used as the basis for decision making). A driverless agent's expert system analyzes environmental data, its private data, and the agent's current state to derive the currently feasible actions. From this set of possibilities and in consultation with its current governing objective, the inference engine component of the ES selects the best action to take next and creates an argument to support the agent's request to implement it. An AV formally expresses its action plan by announcing it to its environment. This announcement is made by casting the public argument supporting the intended action into the argumentation pool of the local peer cluster. The argument is a de facto bid for a lane assignment containing the vehicle identification, projected spatial position and other data. The local community collectively makes a decision whether the planned action an agent is asking to implement is to be allowed or prohibited. All AVs in the affected multi-agent group cast a *for* or *against* vote on all arguments played by their peer cluster, including their own. Some bids will be in conflict and thus in jeopardy of rejection by the community. Conflict is resolved by the vote tally and, where necessary, by the concession preferences of involved agents. Vehicles whose arguments were accepted, are given the go ahead to implement their intended actions. Those, whose bids were rejected, implement a preset default action. After approved actions have been taken, the process restarts and repeats cyclically.

4.1.3 The Environment

Without restricting generality, consider the following simplified example setup.

The environment is a simulated multi-lane expressway with each lane having characteristics pertaining to (i) *maximum speed*, (ii) *minimum speed*, (iii) and *emission level*. For right-hand traffic, lanes closer to the left highway border have higher allowed speed ranges. As allowed speeds grow, so do emission levels.

The vehicle agents themselves are defined as having (i) *current speed*, (ii) *preferred speed*, (iii) a Boolean *amenability rating*, and (iv) a level of *objective emphasis*. The amenability rating refers to the willingness of a vehicle to cede its claim to a projected lane position to another vehicle whilst in conflict with said vehicle. The objective emphasis attribute denotes the level of prioritization a vehicle currently places on its (one) main objective. The preferred speed and the objective emphasis determine the vehicle's target lane and speed. The governing objective and the emphasis the vehicle places on it have a direct effect on the way agents make decisions and how they perceive the decisions of others.

In traffic, each vehicle maintains a required safety distance (buffer zone) between one and two car lengths from the vehicle directly ahead of it as currently defined by the system.

4.1.4 The Density of Agents

The buffer zone requirement puts a natural upper bound on the number of vehicles that can travel in each lane.

Lemma I.1. The maximum number of vehicles n of uniform length c allowed on a single lane segment of length L is $\left\lfloor \frac{L}{3c} \right\rfloor \leq n \leq \left\lfloor \frac{L}{2c} \right\rfloor$.

Proof. The space that an agent occupies in the lane is its own length c plus the required safety distance $d = xc$, where $x \in [1, 2]$. For n agents $nc + nxc \leq L$ or $n \leq \frac{L}{c(1+x)}$. Since n is a positive integer, $n = \left\lfloor \frac{L}{c(1+x)} \right\rfloor$. And since $x \in [1, 2]$, $\left\lfloor \frac{L}{3c} \right\rfloor \leq n \leq \left\lfloor \frac{L}{2c} \right\rfloor$. □

4.1.5 Formulating Arguments and the Feasibility of Actions

The expert system (ES) for lane selection among smart vehicles developed here utilizes the Toulmin style form of argument formation presented in 3.4.1. The ES of an agent is able to infer and analyze context sensed from the environment and determine the optimal next action to take. It embodies two main components. The first one is the inference engine.

The inference engine component of the expert system is responsible for determining the best potential course of action determined from sensed data and for formulating an argument to support it. As seen in Figure 4.1, environmental conditions and the outcome of the previous argumentation round are evaluated to determine all current feasible actions that can be taken by the vehicle.

Agents have five potential atomic actions: (1) *move up one lane*, (2) *move down one lane*, (3) *maintain speed*, (4) *decelerate*, and (5) *accelerate*. Note that "move up" signifies moving to a faster lane (the left lane in right-hand traffic). Similarly, "move down" means a move to a slower lane. The conditional rule sets pertaining to these five actions consider system priority, basic physical constraints, and argumentation constraints. The feasible action derivation mechanism is shown in Algorithm 1.

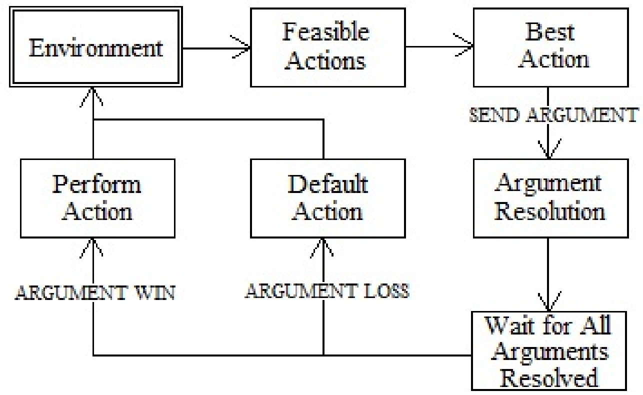

Figure 4.1: The Life Cycle of Argument Formation, Conflict Resolution, and Action Enactment

---

Algorithm 1: Feasible Action Derivation

---

if LaneUp in bounds **and** LaneUp not occupied **then**
    $\mathit{CanMoveUp}$ = true;
if $\mathit{LaneDown}$ in bounds **and** $\mathit{LaneDown}$ not occupied **then**
    $\mathit{CanMoveDown}$ = true;
$\mathit{CanMaintainSpeed}$ = true;
if $\mathit{SpaceBehind}$ not occupied **or** ($\mathit{SpaceBehind}$ occupied **and** $\mathit{CurrentSpeed}$ of $\mathit{CarBehind} < \mathit{CurrentSpeed}$ of myself) **then**
    $\mathit{CanDecelerate}$ = true;
if $\mathit{SpaceAhead}$ not occupied **or** ($\mathit{SpaceAhead}$ occupied **and** $\mathit{CurrentSpeed}$ of $\mathit{CarAhead} > \mathit{CurrentSpeed}$ of myself) **then**
    $\mathit{CanAccelerate}$ = true;

An agent can move up if the lane above is within the bounds of the environment and the spatial position it seeks to occupy is not obstructed. Not obstructed means the desired space is not currently occupied by another agent and there are no other agents within a cone that maintains a safe space so that potential collisions with decelerating vehicles ahead and accelerating vehicles behind in the target lane are avoided. Similar conditions of bounds and availability apply to the action of moving down. As illustrated in Figure 4.2, the vehicle in the middle lane is directly blocked from moving up a lane by another vehicle in the position it is seeking to occupy, and its move down is obstructed by a vehicle in the safety cone, so both lane changes are not feasible at this moment in time.

Acceleration and deceleration are contingent upon the availability of space ahead or behind coupled with the speed of blocking vehicles. Naturally, all decisions are preempted by a collision avoidance mechanism.

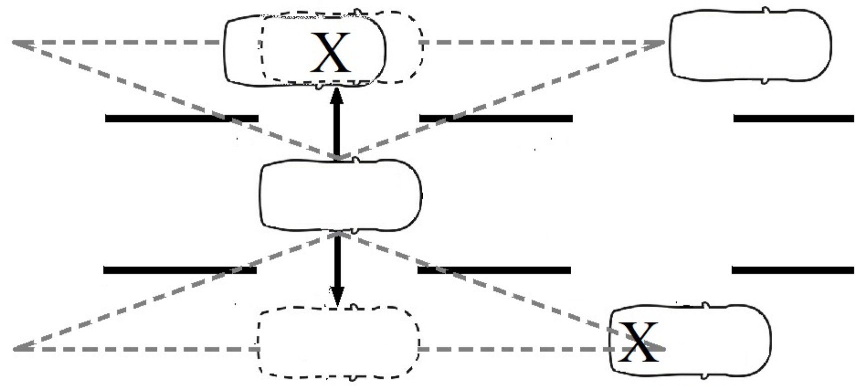

Figure 4.2: Feasibility of Lane Changes Example

From the feasible actions available to the agent, rule sets determined by the governing objective arrive at the (single) next best action to take. The inference engine generates an argument to represent the request to receive approval to implement the selected action. The bid becomes public when the ES submits it into the argumentation pool where it might have to compete with the bids of other agents with potentially conflicting interests. With the played argument an agent stakes a claim over the projected spatial position it seeks to occupy should the chosen action be successfully implemented from the current position the agent is in. The vehicle will be at odds with other vehicles within the current road configuration vying for the same position. Before a decision can be made on who wins a conflicted position, the system needs to resolve all conflict. This is achieved by the second component of the expert system.

4.1.6 Arbitrating Conflict

The inter-agent argumentation resolution component of the expert system employs the use of augmented social abstract argumentation. The argumentation framework and its corresponding semantic framework, the particulars of which are laid out in this section, identify conflicting arguments and use the social support of each argument determined through voting to arrive at a conflict-free set of winner and loser agents for the round of the game.

4.1.6.1 The Argumentation Framework

As arguments are played into the pool, the argumentation framework modeling their relationships can be built.

Definition I.9. An *argumentation framework* $F = \langle A, R, V_a \rangle$ consists of a set of arguments, A, a set of relations between arguments, R, and a set of votes for the arguments, V_a, where:

- A is finite. An argument $a_i \in A$ is a pair $a_i = \langle i, p_i \rangle$ composed of a vehicle i, and the very next highway position p that the vehicle is projected to occupy based on the best action generated by its inference engine;
- R is finite. A relation between two arguments a_i and a_j is denoted as a binary attack relation (a_i, a_j) such that argument a_i attacks argument a_j (meaning that p_i and p_j are incongruent), further denoted as $(a_i \curvearrowright a_j)$;
- V_a is finite. Each vote $v \in V_a$ takes the form $v_{a_i} = \langle a_i, v_+, v_- \rangle$ where $a_i \in A$, and v_+ denotes a vote for argument a_i, while v_- denotes a vote against argument a_i.

It should be noted that the position p_i can be comprised of an x and y coordinate in the plane or as (x, y, z) should/when vehicle lanes become available in 3-dimensional space.

Conflicts are formed when two arguments are found to be in competition for the same global position.

Definition I.10. The *set of binary attack relations between arguments*, R, is comprised of all conflicting arguments in the set of arguments, A. The inference engines of vehicles i and j have determined the next best positions, respectively, as p_i and p_j. Upon detection of conflict, the corresponding arguments are paired into an argument relation and added to the existing set R of relations between arguments:

$$\forall a_i \in A \iff \exists a_j \in A \text{ s.t. } p_i \equiv p_j \Rightarrow R = R \cup \{(a_i, a_j), (a_j, a_i)\}.$$

4.1.6.2 Objective-Related Voting

In every argumentation round all agents in the designated local group cast votes to determine the corresponding social support for each of the arguments in the pool. Each agent confers with its own expert system to determine if another's planned action would negatively affect its next move. Thus, for each argument in the $a_i \in A$ there exists social support $v_{a_i} = \langle a_i, v_+, v_- \rangle$, that contains all of the approval votes v_+ and the rejection votes v_- cast for a_i.

Definition I.11. The *set of argument votes* V_a registers the social approval or rejection of any given vote cast in the argumentation round. Every argument a_i has a corresponding $v_{a_i} = \langle a_i, v_+, v_- \rangle$ which encompasses the social support for a_i:

$$\forall a_i \in A \; \exists v_{a_i} \in V_a \text{ s.t. } a \equiv v_{a_i}.$$

All vehicles will naturally generate "against" votes for bids fully or partially coinciding or interfering with the position they are bidding on and cast "for" votes for their own bid. The remaining voting decisions are made through the perspective of the governing objective.

4.1.6.3 The Semantic Framework and Conflict Resolution

A semantic framework S is introduced to model the argumentation framework described in Definitions I.9-I.11.

Definition I.12. A *semantic framework* $S = \langle \Psi, \Gamma, \neg \rangle$, consists of a vote evaluation function, Ψ, a conflict reduction function, Γ, and a unary negation operator, $\neg$, where for two arguments $a_i, a_j \in A$,

$$\neg(a_i \curvearrowright a_j) \equiv (a_j \curvearrowright a_i).$$

A function that converts v_{a_i} to a corresponding value of social approval $v_{+/-}$, gives the social support for a given argument a_i. The difference between the votes for (v_+) and against (v_-) an argument a_i impacts $v_{+/-}$ such that $v_{+/-} = v_+ - v_-$.

Definition I.13. A *vote evaluation function*, Ψ, accepts votes for an argument a_i and returns a scalar value of support $v_{+/-}$ for a given argument a_i,

$$v_{+/-} = \Psi(v_{a_i}).$$

To resolve ties among arguments who garner the same level of social support, the amenability rating attribute of the agents owning the conflicting arguments must be taken into consideration. Amenability is a Boolean value of either *true*, to indicate that the agent postulating an argument is willing to concede if tied with another agent positing a conflicting argument, or *false*, where the agent will not concede to another agent in the event of a tie. If both agents have the same amenability value, then there is no winner, and both would lose as a result.

Definition I.14. A *tie resolution function*, T, considers the argument relations $r_i, r_j \in R$ and the amenability levels of their aggressing arguments, denoted as c_{r_i} and c_{r_j}, respectively, to resolve a tie in social support between the two arguments. The loser, or losers, of the operation are returned into a set ℓ, s.t.

$$\ell = T(r_i, r_j) = \begin{cases} r_j & \textit{if } c_{r_i} = false \textit{ and } c_{r_j} = true, \\ r_i & \textit{if } c_{r_i} = true \textit{ and } c_{r_j} = false, \\ r_i, r_j & \textit{if } c_{r_i} \equiv c_{r_j}. \end{cases}$$

Finally, a conflict reduction function is necessary to resolve all attack relations identified within R. The function reduces the set of relations R to a conflict-free set R_r with the help of the set of argument votes V_a using the process outlined in Definition I.15.

Definition I.15. A *conflict reduction function*, Γ, accepts a set of argument votes, V_a, and a set of relations, R, and reduces R to a conflict-free set of relations, R_r, with $R_r \subseteq R$. The vote evaluation function, Ψ, determines the social support for each pair of conflicting arguments a_i and a_j involved in the attack relations r_i and r_j in which a_i and a_j are the respective aggressor. The relation whose aggressor has garnered less social support is removed from the set. Should the social support for both aggressor arguments end in a tie, the resolution function, T, is utilized to determine if either agent will collaborate and abandon their claim to the desired position. Applying the conflict reduction function repeatedly for all conflict pairs delivers a conflict-free set $R_r = \Gamma(R, V_a)$:

$$\forall r_i \in R \iff \exists r_j \in R \text{ s.t. } r_i \equiv \neg r_j \Rightarrow$$

$$\Rightarrow R_r = R \setminus \{ \begin{cases} r_j & \textit{if } \Psi(v_{a_i}) > \Psi(v_{a_j}), \\ r_i & \textit{if } \Psi(v_{a_i}) < \Psi(v_{a_j}), \\ T(r_i, r_j) & \textit{if } \Psi(v_{a_i}) \equiv \Psi(v_{a_j}) \end{cases} \}.$$

Regardless of the level of social support, an agent who posits an argument that is unconflicted is allowed to carry out the action corresponding to that argument. Conversely, if an agent posits an argument which is in conflict, any level of social support short of the maximum in the conflicted group will deliver a loss and the agent will be forced to implement a default action for the round. Agents implement their approved actions and after a short adjustment period the process can begin again.

In a discrete environment, this model can deliver resource allocation as a one time act or be repeated as necessary. In a continuous setting like vehicle traffic, as applied here, the process takes place in rounds so a loss of projected position in one round has no long term effect on the chances of an agent to progress towards its preferred lane position and speed setting.

Each round of this model will result in the set of winning arguments being, at the very least, a preferred extension (the maximal admissible set).

4.2 EXAMPLE SCENARIO

Consider the following simplified scenario where 6 autonomous vehicle agents on a 3-lane highway are travelling at certain speeds in accordance with their main objectives. Of these vehicles, three prioritize travelling in lanes with minimal congestion (V, W, X), two focus on minimizing their personal travel time (Y, Z), and one has the reduction of global emissions as an objective (U). At the beginning of the round, the inference engines of all agents derive the best next actions for themselves to take. Vehicles V, W, and Y are to maintain speed, Vehicle U is to accelerate, X is to move down a lane, while Z is to move up. The current configuration and intended actions are illustrated in Figure 4.3. As a result of the recommended lane changes, both Vehicle X and Vehicle Z bid for a coinciding (or overlapping) position that is directly ahead of Vehicle Y.

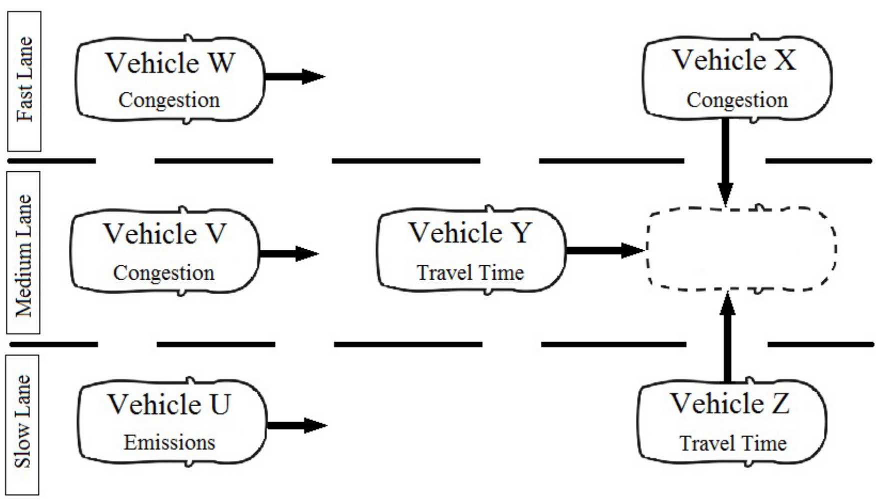

Figure 4.3: Social Abstract Argumentation Among Driverless Vehicles on a Highway Example

Recall that the argument an agent plays in support of its bid to implement its selected action contains the projected next highway position it aims to occupy. For simplicity, in this example scenario the arguments are simply labeled with their corresponding vehicle identification. Thus, the set of all arguments, A, generated by the vehicles within the scenario is:

$$A = \{U, V, W, X, Y, Z\}$$

The pool of submitted bids results in the argumentation framework for the round. Vehicle X currently gains the most utility from arguing for a configuration that both vehicle Y and vehicle Z are at odds with. No other disputes over imminent positioning changes are present within this set of vehicles for this round of argumentation. Arguments generated by X, Y, and Z will pairwise be in conflict and the attacks between them will make up the set of binary relations R.

$$R = \{(X \curvearrowright Y), (X \curvearrowright Z), (Y \curvearrowright X), (Y \curvearrowright Z), (Z \curvearrowright X), (Z \curvearrowright Y)\}$$

The existence of conflict thus clearly established, it is clear that not all agents will gain approval for their planned actions. Since the action tied to an argument can only be implemented once all conflict in the argumentation pool is settled and a determination is made whether the vehicular agent is allowed to proceed, the argument resolution component of the system will be activated.

Votes are cast in consultation with the agents' own objectives and with regard to positional interference from others. In general, congestion agents (V, W, X) will approve the arguments of others as long as they are not driving in or entering their current or intended lane. Global emissions agents (U) will only approve of other vehicles moving down or driving in lanes coinciding with or lower than the one they have deemed acceptable for themselves. Travel time agents (Y, Z) will approve any argument except those that impede their ability to accelerate, maintain speed, or move to a different lane.

For the example shown in Figure 4.3, the corresponding attack graph representing all arguments in set A, their relations contained in set R, and the votes cast in relation to them, are depicted in Figure 4.4. The set V_a of social support tuples $v_{a_i} = \langle a_i, v_+, v_- \rangle$ for each argument is:

$$V_a = \{\langle U, 6, 0\rangle, \langle V, 4, 2\rangle, \langle W, 4, 2\rangle, \langle X, 3, 3\rangle, \langle Y, 2, 4\rangle, \langle Z, 2, 4\rangle\}$$

The social support will help extract a conflict-free set of arguments S from the argumentation framework with the help of the defined semantics. S will contain the set of winning arguments. To begin with, $S = \{\emptyset\}$.

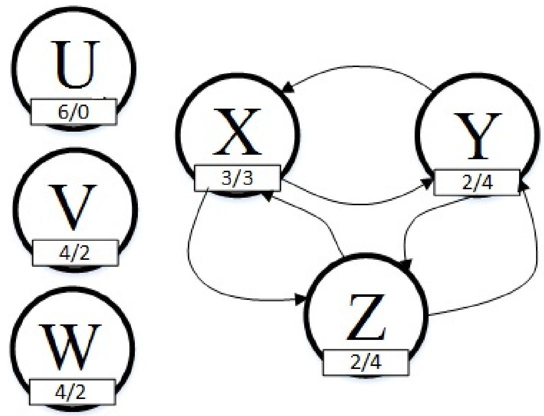

Figure 4.4: Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - Attack Graph with Argument Votes

Vehicle U's best action for the current round is to accelerate in its lane. Its chosen action is not in conflict with any of the actions the remaining 5 agents plan to take - vehicle U is not looking to enter lanes congestion focused vehicles V, X, and W are currently in, and it will not hinder travel time focused vehicles Y and Z, so they all cast approval votes for U's argument. Argument U thus receives 6 *for* votes (including its own). There are no attack relations the argument is involved in so it is absorbed in the conflict-free argument set, $S = \{U\}$, essentially being labeled *in*. Being in the reduced set means that agent U will be given approval to implement its chosen action at the end of the round.

Vehicle V posits an argument according to which it plans to remain in its lane. It will receive two votes of disapproval - by vehicle U because V is in a higher emission lane than U approves of, and by vehicle X because it congests the lane X intends to be in. V has no attack relations with U or others, so argument V is acceptable with respect to S and can be labeled *in* and absorbed into the conflict-free set, making $S = \{U, V\}$.

Similarly agent W's decision to remain in its current lane garners a negative vote from emissions vehicle U, since W is not driving in or moving to a lower emission lane and from congestion vehicle X, because W is congesting X's current lane. W's argument is fully unconflicted, and as a result, $S = \{U, V, W\}$.

The vote tally and reasoning for the conflicting arguments X, Y, and Z is shown in Table 4.1.

Table 4.1: Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - Determination of Votes for Conflicted Arguments X, Y, and Z

Vehicle (Objective)	Vote for X	Vote for X Reasoning	Vote for Y	Vote for Y Reasoning	Vote for Z	Vote for Z Reasoning
U (Emission)	V_+	X is moving to a lower emission lane	V_-	Y is in a higher emission lane than I approve and it's not moving down	V_-	Z is moving to a higher emission lane
V (Congestion)	V_-	X will congest my lane	V_-	Y is congesting my lane	V_-	Z will congest my lane
W (Congestion)	V_+	X is moving out of my lane	V_+	Y is not hindering me	V_+	Z is not hindering me
X (Congestion)	V_+	I am X	V_-	Y is taking my desired position	V_-	Z is taking my desired position
Y (Travel Time)	V_-	X is taking my desired position	V_+	I am Y	V_-	Z is taking my desired position
Z (Travel Time)	V_-	X is taking my desired position	V_-	Y is taking my desired position	V_+	I am Z

For conflicted arguments, the social support represents the graduality of their acceptance by the system. The vote evaluation function and the conflict reduction function presented in Definitions I.13 and I.15 will reduce the set of attack relations R down to a conflict-free set $R_r = \Gamma(R, V_a)$. With X gathering the most social support, the reduced set thus contains two remaining attack relations $R_r = \{(X \curvearrowright Y), (X \curvearrowright Z)\}$. The aggressor X will be absorbed into the *in*-label (winner) set of arguments $S = \{U, V, W, X\}$ seen in Figure 4.5.

Agents that are the aggressor of attack relations contained within $R \setminus R_r$ are labeled

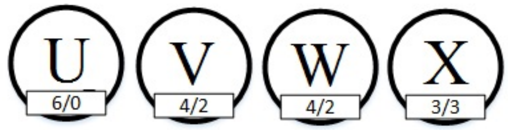

Figure 4.5: Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - Set of Winning Arguments After Conflict Reduction

out (in this example the losing agents Y and Z) and are forced to amend their choice of best action to the automatically accepted state of "decelerate." The remaining agents either have no conflicting argument relations or were elected as the top choice among their attack subsets and will be approved for their selected actions. Having reached the end of the argumentation round, vehicles can now proceed to implement their approved actions. After a short adjustment period, the next round of sensing the environment, deriving best actions, argument submission, voting, conflict rectification, and action enactment will start again.

Note: For this example, the set of winning arguments $S = \{U, V, W, X\}$ is conflict-free as it has no internal conflict. It is also an admissible set since through R_r it can defend itself from all attacks against it. It is the maximal admissible set, which makes it a preferred extension. S is also a stable extension because it attacks all arguments $A \setminus S$ outside of it.

The illustrative example presented here was implemented and verified in a NetLogo simulation as seen in Figure 4.6. The thicker lines stemming from vehicle avatars visually represent next best actions. Horizontal lines parallel to road markings represent maintaining speed (pointing forward) or deceleration (pointing backward), slightly elevated horizontal lines signal acceleration, vertical lines correspond to lane changes as intended actions. A thin vertical line represents a just implemented lane change. A red patch of pavement signifies a conflicted position.

The next round (seen in Figure 4.7) of argumentation puts vehicle W at odds with vehicle Y. Vehicle W posits an argument reflecting its wish to remain in the fast lane and

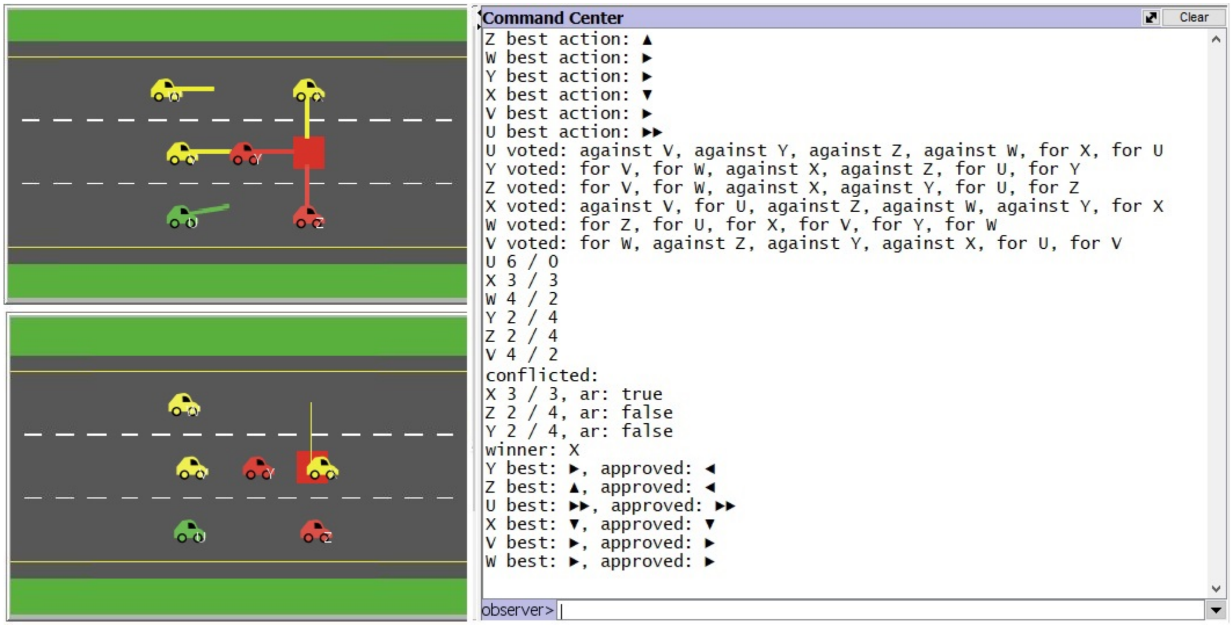

Figure 4.6: Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - NetLogo Simulation Results Round 1

maintain its speed as it currently has the best congestion level and no change is necessary. Having lost the position directly ahead to vehicle X in the previous round and now having an opening in the lane above, vehicle Y puts forth an argument for moving up to the fast lane. Agent W is opposed to that move since it would increase the congestion in its lane and be in direct conflict with its desired position. Both Y and W's arguments garner the same social support from the group and the outcome is decided based on the amenability levels of each agent. In the same round, vehicle Z is not able to move up a lane because it is blocked by vehicle X so it chooses to accelerate until in subsequent rounds it is able to move up to the middle lane.

In the current model, each agent votes in isolation on each argument posited by another agent. In the first round, as seen in Table 4.1 vehicle W voted for vehicle X moving out of its lane but it also voted for the arguments of vehicles Y and Z, which were in direct competition with vehicle X for the middle lane position. Positive votes for Y and Z could have sabotaged the ability of vehicle X to change lanes, even though X leaving

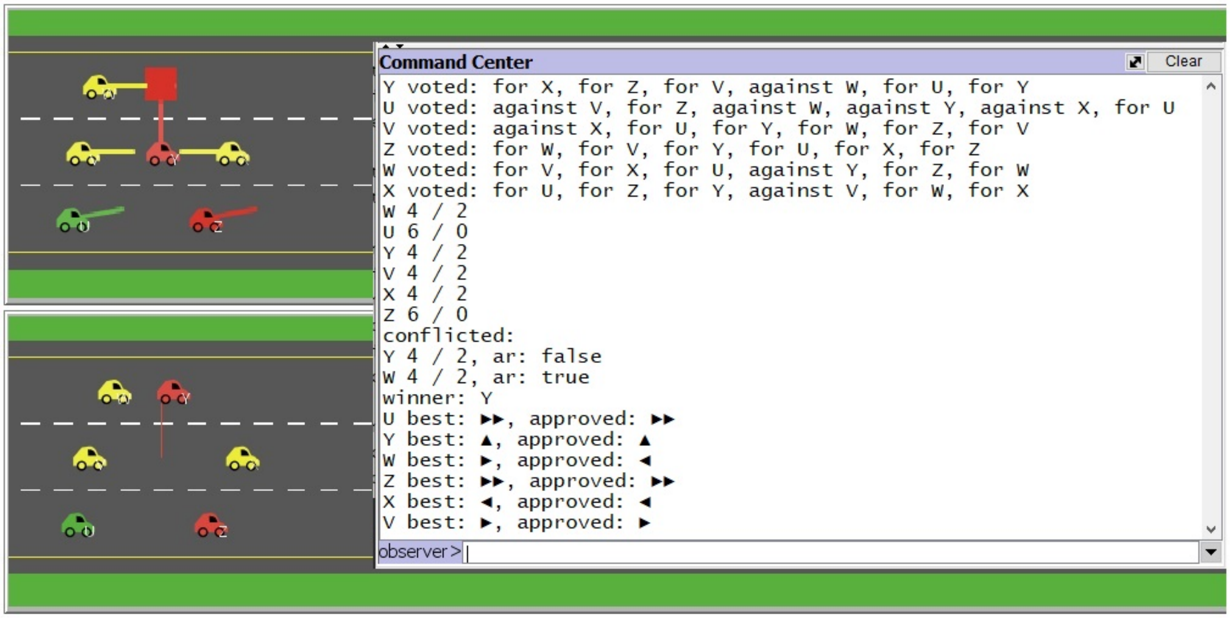

Figure 4.7: Social Abstract Argumentation Among Driverless Vehicles on a Highway Example - NetLogo Simulation Results Round 2

the lane would have been preferred by W in accordance with its congestion objective. In the following round vehicle V votes for vehicle Y leaving the lane they currently share but it also votes for vehicle W, since W is not currently negatively affecting it. Casting positive votes for both W and Y even though vehicle V has a vested interest in vehicle Y winning the argument is not the best strategy for the agent. An added layer of strategy during the voting process where the agent considers the conflict relations that may directly affect it before it casts its votes is a possible future improvement of the model.

4.3 EMPIRICAL MODEL

A test bed was created to implement and verify the argumentation model.

4.3.1 Environment and Agent Characteristics

As outlined before, the simulated expressway has lanes with minimum and maximum speed characteristics and an emission level attribute correlated to the allowed speed range for the lane. For simplicity, these features are assigned valid values between 1 and 10 with 1 being the lowest emission level or speed and 10 being the highest. For a simulated three-lane highway, as shown in Table 4.2, the top lane, Lane 3, is characterized by the maximum overall speed and also the maximum emission level. These indicators decrease as lanes change down. Lane 1 is the slowest and most emission-friendly lane. These conditions were chosen to simulate and simplify real world multi-lane expressway structure.

Table 4.2: Empirical Model - Lane Characteristics of a Simulated Three-Lane Highway

	Emission Levels (1-10)	Maximum Speed (1-10)	Minimum Speed (1-10)
Lane 3	10	10	7
Lane 2	7	6	4
Lane 1	4	3	1

The attributes of the vehicle agents take the following simplified legal values - current speed and preferred speed are valued between 1 and 10, the objective emphasis attribute takes values between 1 and the number of lanes being simulated with the higher values commensurate with the importance the vehicle assigns to its governing objective. The amenability rating indicator is a Boolean.

A design choice was made regarding the laws governing the travel of vehicles between lanes. As trajectory generation and tracking are out of scope for this algorithmic model, it is assumed that once a vehicle starts a lane change maneuver the movement will be largely perpendicular to the vehicle's current position.

4.3.1.1 Agent Objectives

In this test bed, agents maintain objectives which may be in one of three priority modes: reduce global emissions (EL), minimize local lane congestion (CL), or minimize personal travel time (TT). The local lane congestion objective is concerned with reducing the congestion of the vehicle's current lane. The personal travel time objective aims to attain and if possible exceed the agent's preferred speed thus minimizing travel time. Finally, the global emission objective's target is to reduce the global emission levels generated by all vehicles on the road.

By maintaining an objective, AVs are working against some or all of the remaining objectives. For example, a vehicle focused on minimizing its travel time will likely select to drive in a faster lane than an agent focused on minimizing its emissions would choose, since faster speeds come with higher emissions. Thus, the action of choosing a faster lane taken to satisfy a travel time objective adversely affects the global emission levels and therefore works against the emissions reduction objective. The tendency of agents focused on reducing emissions to choose the slowest lane and the inclination of vehicles interested in reducing travel time to choose the fastest lane result in higher congestion levels in both the slow lane and fast lane, respectively, and therefore work against the congestion reduction objective.

4.3.1.2 Objective-Based Best Action Derivation

AV agents independently derive all currently feasible actions based on the sensed data as seen in Algorithm 1. Given the availability of actions from the current position, the agent's expert system consults the dominant objective in order to determine which one of these actions is the best next move. That determination is made outside of the context of what agents in the immediate vicinity want to do. The agent is only interested in what action brings it closer to satisfying its own goal. The decision derivation mechanism follows the Toulmin model and the structure presented in 3.4.1. Example arguments which exhibit possible conclusions and actions pertaining to each objective are:

- For the Personal Travel Time Objective: a_1 = Since my current lane has a maximum speed limit below my adjusted preferred speed and the lane above is available; then faster speed in the lane up is possible and will reduce travel time; therefore, move up one lane.

- For the Global Emission Objective: a_2 = Since my lane has a minimum speed limit above my emission adjusted preferred speed and the lane below is open; then a move to a slower lane is possible and will reduce emissions; therefore, move down one lane.

- For the Local Lane Congestion Objective: a_3 = Since I've reached my preferred speed, and the lane above is open and has lower relative congestion than my current lane; then moving up a lane will benefit my local lane congestion level and satisfy my adjusted preferred speed requirement; therefore, move up one lane.

This decision making process is streamlined in the form of rule sets, which each objective contributes to the ES so that, when fired, they extract the best one of the set of currently feasible actions. These rule sets are outlined in Algorithms 2 and 3.

Algorithm 2: Conclusion Derivation Rule Sets for the Personal Travel Time and Global Emission Level Objectives

---

Variables: $TargetLane$ is the lane for which $TargetSpeed$ falls within lane speed limits.
For agents with Personal Travel Time objective
$TargetSpeed = PreferredSpeed + ObjectiveEmphasis$.
For agents with Global Emission Level objective
$TargetSpeed = PreferredSpeed - ObjectiveEmphasis$.
$TargetSpeed$ has a forced lower bound of 1 and upper bound of 10.

set $BestAction$ = maintain speed
if $CurrentLane = TargetLane$ **then**
  if $CurrentSpeed < TargetSpeed$ **and** $CanAccelerate$ **then**
    set $BestAction$ = accelerate;
  if $CurrentSpeed > TargetSpeed$ **and** $CanDecelerate$ **then**
    set $BestAction$ = decelerate;
else
  if $LaneMaximumSpeed > CurrentSpeed$ **and** $CanAccelerate$ **then**
    set $BestAction$ = accelerate;
  if $LaneMaximumSpeed < CurrentSpeed$ **and** $CanDecelerate$ **then**
    set $BestAction$ = decelerate;
  if $CurrentLane < TargetLane$ **and** $CanMoveUp$ **then**
    set $BestAction$ = move up;
  if $CurrentLane > TargetLane$ **and** $CanMoveDown$ **then**
    set $BestAction$ = move down;

TT and EL vehicles derive the next best action with the same algorithm but the underlying variable values are determined differently. For these two types of vehicles, the motivation behind travelling into a higher or lower lane are dependent on the adjusted speed of the vehicle, which is comprised of the preferred speed and the objective emphasis. For the TT objective, the higher emphasis the vehicle places on its goal of shortening time, the more willing it is to favor speeds higher than its preferred setting. For the EL objective, vehicles are willing to drive slower than their preferred speed in order to travel in lanes of lower emission rates. In other words, the priority a vehicle places on its governing objective directly impacts how willing the agent is to ignore its own preferred speed.

Congestion Level (CL) vehicles take a two-fold approach to satisfying their objective. First, the agent strives to attain its preferred speed in its target lane. Unlike TT and EL agents though, which aim to maintain their target lane and emphasis adjusted preferred speed (target speed) once reached, when a CL agent achieves its preferred speed, it considers the lanes above and below, if within physical bounds, and calculates their relative congestion levels within a set radius. The vehicle then selects the lane with the least congestion as its new target lane and adjusts its target speed by a factor of its objective emphasis. The move to a new target lane may not happen immediately if the community collectively denies the request. Once that target is achieved though, the agent reconsiders local congestion levels again and repeats the process. In the event of congestion level ties, a vehicle would always choose to move to or stay in the faster lane.

In order to not impede other traffic as much as possible, even though an AV's priority is to reach its target lane where it can attempt to achieve its target speed, until it has done so, an agent will strive to abide by the speed restrictions of the lane it currently finds itself in (lines 2.8 - 2.11 and 3.16 - 3.19).

One may note that, as seen in the pseudocode in Algorithms 2 and 3 (lines 2.7 - 2.15 and 3.15 - 3.23), vehicles try to get to their target lane where they can reach their target speed as fast as possible. A vehicle which finds itself in the wrong lane will choose to

Algorithm 3: Conclusion Derivation Rule Sets for the Local Lane Congestion Objective

Variables: $TargetLane$ is the lane for which $TargetSpeed$ falls within lane speed limits $TargetSpeed = PreferredSpeed$. $RelativeLaneCogestion$, $RelativeLaneUpCongestion$, and $RelativeLaneDownCongestion$ are calculated within distance ± 10 car lengths and fall within $[0, 1]$

set $BestAction$ = maintain speed;
if $CurrentLane = TargetLane$ **then**
    if $CurrentSpeed < TargetSpeed$ **and** $CanAccelerate$ **then**
        set $BestAction$ = accelerate;
    if $CurrentSpeed > TargetSpeed$ **and** $CanDecelerate$ **then**
        set $BestAction$ = decelerate;
    if $CurrentSpeed = TargetSpeed$ **then**
        if $LaneDown$ within bounds **and** $RelativeLaneCongestion > RelativeLaneDownCongestion$) **and** ($LaneUp$ not within bounds **or** ($LaneUp$ within bounds **and** $RelativeLaneDownCongestion < RelativeLaneUpCongestion$)) **then**
            set $TargetLane = LaneDown$;
            set $TargetSpeed = max(LaneDownMinumumSpeed, TargetSpeed - ObjectiveEmphasis * (1 - RelativeLaneDownCongestion))$;
        else
            if $LaneUp$ within bounds **and** $RelativeLaneCongestion \geq RelativeLaneUpCongestion$ **and** ($LaneDown$ not within bounds **or** ($LaneDown$ within bounds **and** $RelativeLaneDownCongestion \geq RelativeLaneUpCongestion$)) **then**
                set $TargetLane = LaneUp$;
                set $TargetSpeed = min(LaneUpMaximumSpeed, TargetSpeed + ObjectiveEmphasis * (1 - RelativeLaneUpCongestion))$;
else
    if $LaneMaximumSpeed > CurrentSpeed$ **and** $CanAccelerate$ **then**
        set $BestAction$ = accelerate;
    if $LaneMaximumSpeed < CurrentSpeed$ **and** $CanDecelerate$ **then**
        set $BestAction$ = decelerate;
    if $CurrentLane < TargetLane$ **and** $CanMoveUp$ **then**
        set $BestAction$ = move up;
    if $CurrentLane > TargetLane$ **and** $CanMoveDown$ **then**
        set $BestAction$ = move down;

make the lane switch in the direction of its target lane over adjusting its speed to follow speed restrictions in its current lane. This is due to the inherent blocking nature of reasoning in the best action derivation algorithms. So if a vehicle is travelling below its target speed and below the lane maximum speed, and can either move up to a faster lane or accelerate, it will choose to move up. In this case, because of blocking, "Best Action" is assigned to "Move Up" overwriting the previously assigned "Accelerate" decision. The same is true when an agent has to make a decision in what fashion to travel at a slower speed in a lower lane. Since until vehicles have reached their target lane, lane changes dominate speed adjustments, in order to lessen the potential negative impact on traffic in the new lane, "Move Up" lane changes come with an automatic acceleration factor and "Move Down" lane changes adjust the current speed down.

Through these algorithms, for each round, given the same environmental context, the expert systems of autonomous vehicles prioritizing different objectives will be able to select their top actions and play against others in an attempt to win the right to implement them.

4.3.2 Voting and Resolving Ties

All agent vehicles within the section of highway under consideration view the pool of arguments submitted by all agents in their cluster. SAA requires all agents to vote on what arguments they view most align with their specific objectives and do not hinder their own planned actions. Generally, TT agents will disapprove of other vehicles driving too slow and bidding for a position close enough to obstruct their current one (in case they lose) or obstructing the position they wish to be in (in case they win). CL agents will vote against all bids to enter or remain in their current lane or the lane they aim to be in on the next move. EL agents will only vote for bids to move down a lane or drive in a lane coinciding with or lower than their target lane, which they have deemed acceptable for emission levels and will reject all others.

After the voting, when faced with social support ties, in the general case, the tie resolution function as presented in Definition I.14 is fair and sufficient in resource distribution. However, in the test bed application domain of vehicle traffic on a multi-lane road, in the interest of not wasting resources and impeding progress, the function was amended to include a random tie break in the event of coinciding amenability rating values. This way someone always wins the conflicted position.

These are the abstract concepts relevant to the test bed.

4.4 SIMULATION

The empirical model was implemented and verified with a NetLogo simulation.

4.4.1 Settings

The simulation is fully distributed, as arguments are public, and agents' expert systems run separately and simultaneously, arriving at the same outcome. If desired, the model leaves the possibility open to designate an arbiter that extracts the conflict-free set from the argumentation pool and vote tally and communicates the win/loss decision to the local AV group.

The simulation represents a fairly limited size wrap-around portion of the road, so all vehicles participate in the bidding and the vote. When deployed on prohibitively long stretches of road, the implementation will require some simple additional code to assign vehicles to local subgroups based on their spatial position.

All agents in the simulation have the same physical characteristics. For simplicity, the design choice was made to disregard vehicle performance and structure even though in the real world these features will influence and restrict the speed at which a vehicle can travel, the maneuvers it can perform, and the emission levels it will produce. Adding agent diversity to better approximate real world conditions is a minor extension and does not affect the model algorithmically.

To clearly distinguish between objectives, vehicle avatars are color coded - red for TT, green for EL, and yellow for CL. The number of agents on the road and the number of lanes can be adjusted. The *number-of-lanes* controller in the simulation environment has an upper bound of 5 but the model is easily scaled by adjusting the legal value range of underlying variables to traffic with an arbitrary (but reasonable) number of lanes, e.g., 11 one way, like California State route 22. This case is shown in Figure 4.8.

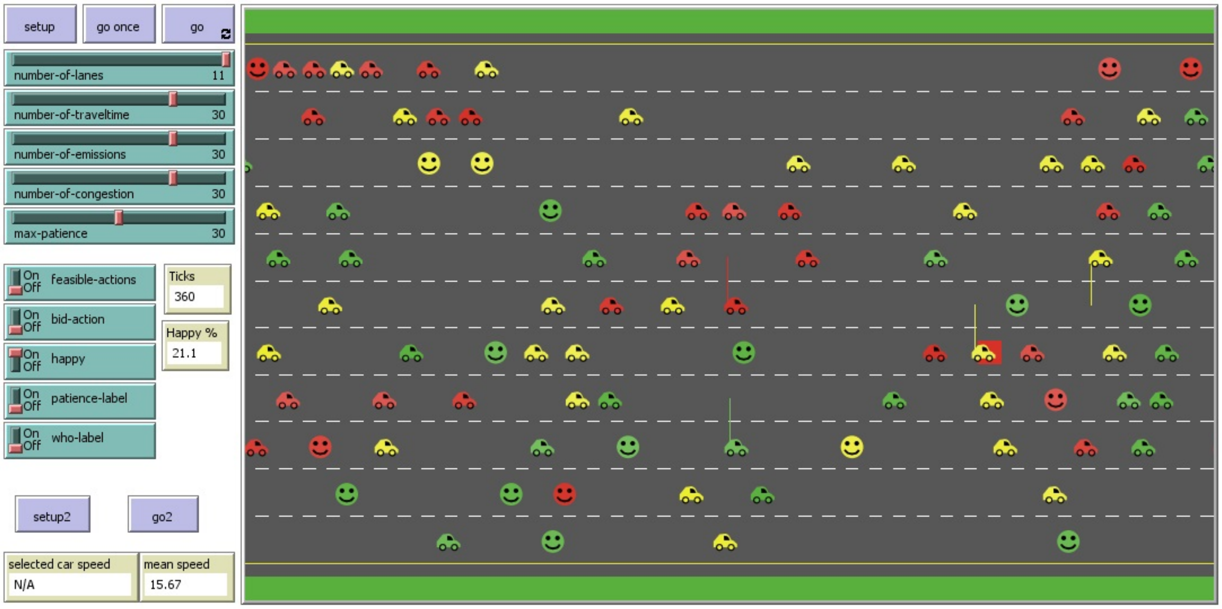

Figure 4.8: NetLogo Simulation of Traffic in 11 Lanes One-Way - Vehicles Shown immediately Following Lane Changes in the Argumentation Round

While the current speed attribute is explicitly defined as the speed at which the vehicle is traveling, for the test bed, the objective emphasis, amenability rating, and preferred speed attributes are randomly generated within the feasible ranges for each vehicle.

Each round of argumentation lasts for 10 time slices (ticks). Deriving best actions, announcing arguments, voting, conflict resolution, and implementation of approved actions all happen in a single tick, while the remaining time in the round is spent progressing forward after adjustments. The model unfolding in a single time slice allows to adjust the total length of the round at will. In general, a full 10-tick round would take a length of several seconds (but the real ideal round length would best be determined in real world conditions), so that agents have the opportunity to make changes often enough but not so frequently that the rate of recurrent computation impedes the vehicle's ability to function and has a detrimental effect on overall traffic.

The simulation also introduces the concept of happiness. Each agent has a target - a lane and speed that satisfy its objective and preference settings, so after reaching that target, the need to make changes should diminish. The *happy* controller shows vehicles that have reached these targets and will now try to maintain them. Such agent avatars can be seen in Figures 4.8 and 4.9 represented by a smiley face. It should be noted that while TT and EL vehicles strive to reach and maintain the happy state, happy CL agents immediately start looking to change to the least congested local lane. So if looking to draw conclusions from the percentage of vehicles in a happy state, one should consider only looking at travel time and emission level vehicles. The proportion of agents that have satisfied and can maintain their objective targets also depends on the amount of vehicles on the road and the relative congestion level they present in relation to the total number of lanes on the road. However, in order to avoid situations in which vehicles are stuck in less than ideal lane/speed conditions for a long time, e.g., travelling in the target lane but at less than the target speed due to a blocking vehicle immediately ahead, the simulation introduces an additional feature for all agents - *patience*.

An agent is assigned a random level of patience between 10 and a user defined level, say 34, as shown in Figure 4.9. This means that a vehicle has patience to last it anywhere between 1 and 3 argumentation rounds. Patience is an integer which decrements with every time slice during which certain conditions are met: (a) the agent is behind a blocking vehicle driving slower than intended, (b) the agent is driving the target speed but not in the target lane, (c) the agent is driving the wrong speed in the wrong lane. Especially in the first case where there is a slower moving vehicle directly ahead, agents are forced to adjust speed down to match the blocking agent in order to avoid a collision. This may drastically change the current speed setting and possibly moving the vehicle further away from its target speed. Once the level of patience of a car dips below positive, a different set of rules (shown in Algorithm 4) preempts the inference engine rule sets from Algorithms 2 and 3.

Algorithm 4: Conclusion Derivation Rule Sets for All Agents With $Patience \leq 0$

---

if $CanAccelerate$ **then**
    set $BestAction$ = accelerate;
else if $CanMoveUp$ **then**
    set $BestAction$ = pass up;
else if $CanMoveDown$ **then**
    set $BestAction$ = pass down;
else
    set $BestAction$ = maintain speed;
reset $Patience$;

The patience derivation rule set introduces two additional best actions - "pass up" and "pass down". These are added for practical purposes relevant to the application domain. A car that has run out of patience will try to avoid disrupting neighboring lanes and look to accelerate in its own lane first. Should that not be possible, the next best thing would be to maneuver to the left into a higher speed lane and overtake the blocking vehicle(s). The next best action would be to overtake on the right (if allowed by local traffic laws). When making regular lane changes, moving to a higher speed lane is accompanied by a half unit acceleration factor while a switch to a lower speed lane comes with a 0.5 deceleration (on a speed scale 1-10). However, such minor changes in speed would not be sufficient when trying to overtake others. Vehicles passing up or down use higher acceleration factors in order to speed up the maneuver and cause minimal disruptions in traffic. Thus, overtake maneuvers momentarily ignore lane speed restrictions for the rest of the round. Patience level for the agent is then reset and the vehicle goes back to its regular decision making algorithm. Of course, the overtake maneuver may be thwarted by a blocking vehicle in the new lane, so success is not guaranteed. Currently, the simulation does not provide agents with the ability to estimate the chance of success of an overtake maneuver before engaging in it. As the traffic configuration continues changing, agents will have the opportunity in the very next round to make decisions to again move them towards their lane and speed target, so failing to pass a blocking vehicle has no long term negative effect.

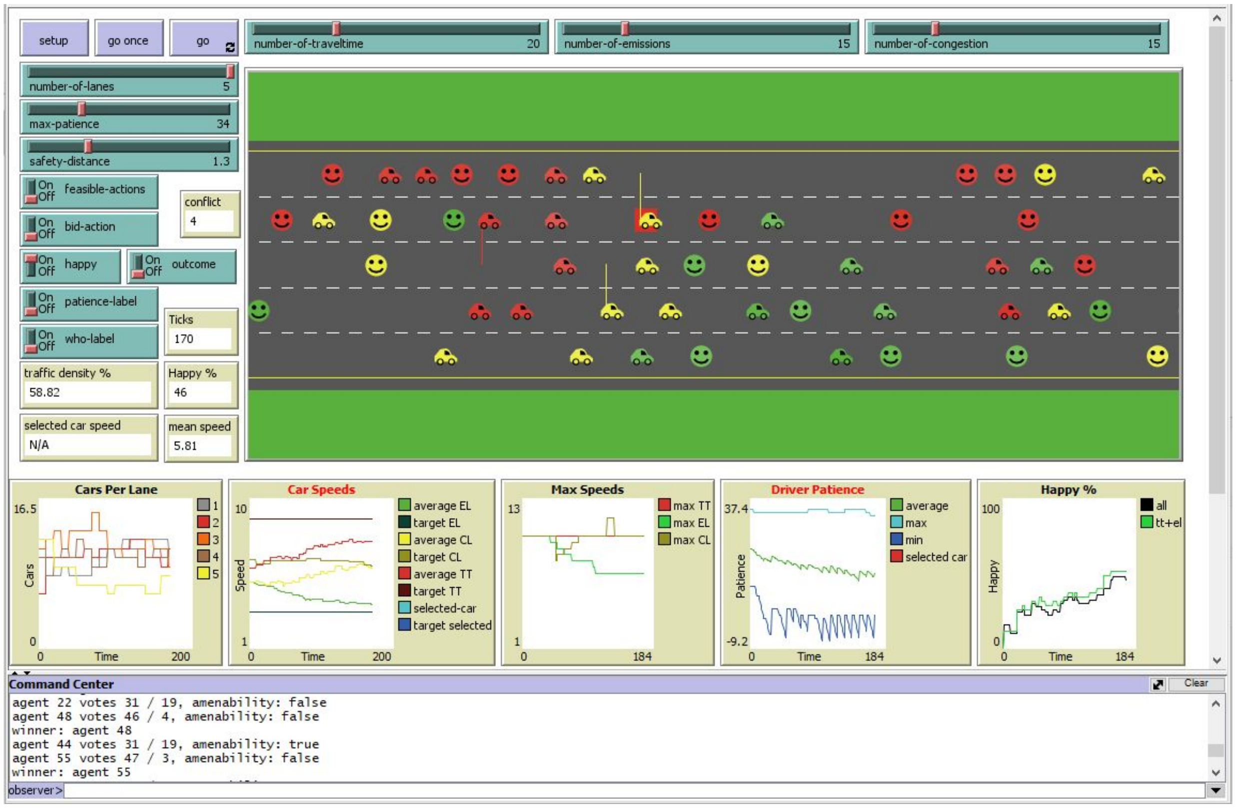

Figure 4.9: NetLogo Simulation of a 5 Lane Road

4.4.2 Test Runs

The simulation was used for test runs of 100,000 time slices (10,000 full rounds) each at different levels of traffic density and buffer zone requirements. The number of vehicles was evenly split among agents prioritizing the 3 different objectives. Results are summarized in Tables 4.3 and 4.4.

At low traffic densities, agents still influence outcomes for their peers through voting but actual positional conflict is rarely present in the system. As the buffer zone widens, conflict is diminished as well. In the simulation settings, the time slice duration and the safety zone cone when determining the feasibility of lane changes were chosen in a way

Table 4.3: Simulation 10,000 Full Cycle Test Runs, Positional Conflict Frequency and Speed Deviation Results

Traffic Density	Buffer Zone	Conflict Sets	Average (Speed - Target Speed)		
			EL agents	CL agents	TT agents
25%	1.0	38	0.00	-0.12	0.16
	1.5	8	0.19	0.28	0.35
	2.0	0	0.00	0.17	0.35
50%	1.0	229	0.09	0.14	0.19
	1.5	24	0.23	0.58	0.35
	2.0	2	0.00	0.56	0.23
75%	1.0	375	0.11	0.71	0.80
	1.5	41	0.15	0.77	0.53
	2.0	11	0.26	0.94	0.59

Table 4.4: Simulation 10,000 Full Cycle Test Runs, Happiness of Travel Time and Emission Level Agents Results

Traffic Density	Buffer Zone	Happiness of EL and TT Agents		
		minimum	average	maximum
25%	1.0	12%	90%	100%
	1.5	7%	70%	93%
	2.0	0%	79%	100%
50%	1.0	0%	76%	97%
	1.5	4%	68%	90%
	2.0	0%	87%	96%
75%	1.0	6%	61%	88%
	1.5	5%	64%	88%
	2.0	6%	59%	82%

that the size of conflict sets was kept small, which additionally drove the total amount of conflict down.

In general agents travelled at speeds around or above their target velocity. For EL and TT vehicles this increase is due mostly to overtake maneuvers and for CL agents, the variability is also due to the moving nature of speed targets as lane congestion conditions determine the next lane and speed goal. The model as designed, achieved satisfactory speed outcomes for all vehicles.

The average level of happiness that TT and EL agents were able to reach was consistently high. Obviously, at lower traffic densities, it is easier to reach and maintain lane and speed targets. At the 25% traffic density level it was possible for all non-CL agents to be simultaneously in a happy state. But even with restricted movement at 75% traffic density and diminished availability of lane positions due to the 2 car lengths buffer zone, the average happiness of travel time and emission level prioritizing agents was above 50%. Naturally, these results do not reflect how many additional vehicles were only marginally removed from their targets as, in the simulation, happiness is an all or nothing indicator.

4.4.3 Time Complexity

Lemma I.2. *The argumentation mechanism runs in $O(n^2)$ time (worst case).*

Proof. In the simulation's current iteration, bid creation takes place simultaneously for all n agents in runtime $O(1)$ for each agent. For n agents (and thus n arguments), AVs simultaneously make voting decisions in runtime $O(n)$.

The distributed conflict resolution takes place at the same time and locally for each agent. If the position the agent is bidding for is unconflicted, the AV immediately knows the action is approved. However, if the position was marked as having more than one contender, the agent must iterate through the conflict set for this position and determine the highest vote-getter to compare to its own bid.

This operations is implemented with nested loops. In the worst case, where all n agents on the road are conflicted there are $\sum_{i=1}^{n}\sum_{j=1}^{n-1}$ iterations to find the argument with the most votes. The loop represented by the second term could see an earlier termination if the agent finds an argument in the set of contenders for the position with higher social support than its own. Additionally, resolving potential ties between the contender with the most social support and the agent's own bid happens in constant time. $\square$

Obviously, n^2 is not ideal. However, this is the general case for application of the resource sharing model in any application domain. In vehicle traffic, where vehicles can advance only one lane at a time by enforcing and strategically selecting the buffer zone requirement when determining if a vacant position is even feasible, the argumentation model can shrink the conflict set down, making conflict resolution computationally trivial. Indeed, in real world conditions, a vacant position of one car length could be coveted by a limited number of agents (the vehicles surrounding the open spot on the road from above, below, ahead and behind which have a feasible trajectory and speed change factor making it possible to obtain that position within the time constraints of the argumentation round). The time slice duration, acceleration/deceleration rate, and buffer zone width can be chosen in a way that the maximal conflict set size is shrunk down to a trivial size, e.g 2-8.

4.4.4 Additional Features, Future Extensions and Avenues for Optimization

The simulation offers additional features like adjusting the required safety distance between vehicles, focusing on a single vehicle, showing identifying characteristics of agents, briefly showing lane change traces, momentarily highlighting conflicted positions in red, showing all feasible and/or the best action for each agent in each round. It also shows informational plots of vehicle density per lane, average target speed vs. actual speed, top speed for the agent type, driver patience, and happiness levels as seen in Figure 4.9.

Additional features specific to the application domain can be added to the simulation while still keeping the integrity of the underlying model. For added flexibility, agents could be given the ability to switch objectives at will or maintain multiple objectives. Mimicking a dynamic argumentation framework, agents can derive the best action for each active objective they possess and play the argument generated for the current governing objective. Conversely, degrees of contribution to each objective can create a de-facto ranking of feasible actions and let the agent choose an argument according to its rank. For progress improvement, agents could sense the feasibility of overtake maneuvers before undertaking them or be given the possibility to maintain speed as a default loss action instead of decelerating. The model is currently implemented as a synchronous process but can be changed to happen asynchronously through agents regulating speed at will within the speed constraints of their lane but announcing intentions to change lanes and forcing a vote from their immediate neighbors. Additionally, agents should be augmented with physical features, and be given the option to join groups and act and vote as a platoon. Inclusions can be made to add obstacles that can be treated as unavailable road portions or to add extraneous vehicles on the road e.g., emergency vehicles for which all cars have to make room. Negotiating such conditions would require the argumentation model to be preempted by a different set of rules and is thus out of the scope for this work.

To improve efficiency and minimize network traffic, vehicles could be asked to vote only on conflicted spacial positions instead of casting a vote for all arguments. Additionally, if agents plan to maintain their status quo, they can be released from the obligation to submit an argument but will still vote on arguments in the pool. Also, if there are no others close by that can be affected by an agent's chosen action, that vehicle can refrain from bidding and voting. The stretch of road governed by the system can be partitioned into subgroups of agents in accordance with which agents are likely to be directly affected by the actions of peers in the local group. For these disjoint agent clusters the argumentation processes can be run in parallel. Any strategy that diminishes the number of arguments in the pool for the round will have a positive effect on system performance.

4.5 CONCLUSION

Part I of this dissertation presented a model for automated multi-objective machine-to-machine argumentation augmented with social voting. This approach delivers a democratic mechanism for conflict resolution and resource sharing through unsupervised machine-to-machine negotiation. The system is fully distributed and benefits from its localized nature and deployment at the edge of the Internet of Things.

A prototyped NetLogo implementation including an example set of objectives governing vehicles' movements and intentions was presented. The expert system is equipped with rules sets to extract feasible actions from the environmental data and determine the most desirable available action to implement next. It demonstrates the capability of the system to self-govern. Features of the prototype are adjustable to fit the deployment scenario.

This model represents a stride toward fully unsupervised negotiation among machines in just about any smart application domain. It augments the ability of interconnected cyber-physical systems to communicate with other agents in the IoT. It affords a fair and organized distributed model of argumentation resulting in democratically approved decisions for a group of agents. It is a simple but powerful tool for unsupervised collaboration and negotiation through communication requiring minimal resources.

Part II

Automated Multi-Objective Machine-to-Machine Negotiation Among Autonomous Vehicles on a Multi-lane Highway Based on Mechanism Design

CHAPTER 5

GAME-THEORETIC MECHANISM DESIGN FOR SMART AUTONOMOUS AGENTS IN THE INTERNET OF THINGS

In a multi-agent system like the Internet of Things, smart devices interact with one another and their decisions affect others. When the moves made by agents working toward an objective are interdependent and pertinent decision attributes include the decisions of others, game theory can be used to formalize the process of reasoning [47]. The need for such arbitration mechanisms becomes increasingly evident as the evolution of technology is causing the world of economic systems to become progressively intertwined with the world of computational systems. Agents in these systems may act in self-interest and change their strategy in an attempt to maximize their own payoff, or may coordinate with others to form coalitions and change strategies as a group to improve the collective utility. The outcomes for all players will be affected by the strategies they employ. As agents behave according to their own objectives, the laws governing the system produce an outcome as a result of these actions and are responsible for the system's success or failure [59]. Mechanism design is the design of directives regulating the interaction between agents such that the system in equilibrium represents a desired outcome as defined by the mechanism designer. The rules may include incentives, mechanisms or laws, while the outcome is often centered around social welfare or profit, or both.

A game theoretical approach based on mechanism design lends itself well to automated negotiation especially when agents must make decisions on how to share a resource.

5.1 GAMES, STRATEGIES, PAYOFFS, AND EQUILIBRIA

In a multi-round game players may maintain private information. In such games of incomplete information players' payoffs depend on their own type (the combination of the private information and preferences of an agent) since there is no information about the types of other players. Each player agent selects a strategy from an available set to play.

The game utility function computes player payoffs given their choice of strategy. For the game or, if the game is played in rounds, for each round, the optimal outcome is an equilibrium so that no player would be better off selecting a different strategy. This dominant strategy solution, where each player has no knowledge of and makes no assumptions about the others and has a unique best strategy independent of the strategies of the rest of the players, may not deliver the optimal payoff for the players but it may lead to an outcome that is desirable. However, games very rarely have a dominant strategy solution and, if it even exists, finding it is a computationally intensive problem. Working backwards, mechanism design, also known as "reverse game theory", aims to construct games with dominant strategy solutions.

Internally, a game consists of players, their strategies, and a way to determine payoffs as formalized in Definition II.1.

Definition II.1. A *game* G is a simple tuple $G = \langle I, S, U \rangle$ where I is a set of players (agents), S is the set of their individual strategies, and U is the utility function.

To participate in the game, each player selects a strategy s_i from a set of available strategies S_i.

Definition II.2. Each participant $i \in I$ in a game $G = \langle I, S, U \rangle$ has an available *strategy set* S_i. S_{-i} denotes the strategy sets of all players other than i.

A strategy is a complete algorithmic approach to game play pre-selected by a player. It governs the actions a game participant undertakes in every potential situation it can face in the course of the game. Thus, each player uses the selected strategy to make moves.

Definition II.3. Let $s = (s_1, s_2, ..., s_I)$ represent a vector of strategies selected by all I players in a game G. The vector s is the game's *strategy profile*. $S = \prod_{i=1}^{I} S_i$ is the set of all agents' strategy sets, i.e., the set of all possible strategy choices for all players or all possible strategy profiles.

The strategy profile of a game determines the outcome for each player and different players may see different outcomes. Each player orders the outcomes according to desirability congruent with its own preferences. The preference ordering is a complete, transitive, reflexive binary relation on the set S. Player i weakly prefers a strategy S_1 over strategy S_2 if it considers the outcome of S_1 to be better or equal to that of S_2. The outcome of the game is determined by the strategies played (selected) by all players.

Definition II.4. In a game G the *utility function* U with $u_i : S \to \mathbb{R}$ computes the payoffs for players given their particular choice of strategy.

Alternatively to the utility function shown in Definition II.4, a payoff in some games may be considered as the cost incurred by a player represented by a *cost function* C with $c_i : S \to \mathbb{R}$. Naturally $u_i(s) = -c_i(s)$. Utility is non-negative and the payoff obtained by each player depends not only on its own strategy but also on the strategies chosen by all other players.

A strategy profile with the property that no agent can obtain a higher utility payoff when unilaterally deviating from the profile is known as a Nash equilibrium [98].

Definition II.5. A strategy profile $s \in S$ is in *Nash equilibrium* if for each player i, the selected strategy $s_i \in S_i$ and all of the other possible strategies $s'_i \in S_i$, the outcome of the original strategy choice is no worse than the outcome of the alternate strategies, i.e. $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$.

A Nash equilibrium is a solution to a non-cooperative game of two or more players where each player knows the equilibrium strategy of the other players and a player can gain nothing by changing their strategy in isolation. Similarly, in a cooperative game a strong Nash equilibrium exists when there is no coalition A of players that can simultaneously change their strategy to improve or retain the outcomes for each of its members. In other words there exists no group of players $A \in I$ with a strategy profile s_A such that the utility of each player in the coalition improves if the group jointly chooses a new strategy s'_A, i.e. $\nexists A$ for which $u_i(s) \leq u_i(s'_A, s_{-A})$ and the inequality is strict for at least one player $i \in A$.

An equilibrium is a form of stability and a competitive game may have multiple equilibria with possibly vastly different payoffs or no equilibria at all. Finding a Nash equilibrium is a PPAD-complete (polynomial parity arguments on directed graphs [112]) problem [32].

The issue with the Nash equilibrium as a solution is that since multiple Nash equilibria might exist, players may not know which one they are strategically playing towards. Also, the implicit assumption is made that each player has complete information about all other players and their preferences which is not usually the case. A more relaxed solution for games of incomplete information (Bayesian games [57]), is the Bayesian-Nash equilibrium in which each player has *beliefs* about the types of all other players and chooses a utility-maximizing strategy based on those assumptions. Sometimes however, players have no knowledge of and make no assumptions about others. If that is the case and at the same time players have a unique best strategy independent of the strategies of others, the game has a dominant strategy solution. The dominant strategy does not necessarily deliver the optimal payoff for everyone but its outcome may be desirable. This is the case with the well-known *Prisoner's Dilemma* [140] game where the dominant strategy for both players is to confess even though the optimal outcome is reached when both players simultaneously lie.

Dominant strategies are considered superior to other strategies, regardless of the strategy choices of other players. Strategic dominance can be strict or weak. Irrespective of other players' strategies a strictly dominant strategy always provides greater utility to the player as compared to all other potential strategies the player can select, while a weakly dominant strategy delivers utility that is at least the same or better than all other potential strategy choices.

Each player choosing their own dominant strategy results in a dominant strategy equilibrium. Any dominant strategy equilibrium is a Nash equilibrium (but the opposite is not true). Games very rarely have a dominant strategy solution, so in its absence, the less restrictive Nash equilibrium is an acceptable and desirable game-theoretic stable solution even though it also may not be optimal for the players. Constructing games with dominant strategy solutions is the aim of mechanism design.

5.2 MECHANISM DESIGN, DOMINANT STRATEGIES, STRATEGYPROOFNESS, AND INCENTIVE-COMPATIBILITY

Generally, rational agents act selfishly in an attempt to maximize their own utility based on their own undisclosed preferences. If the aim of a system is to extract the best collective payoff, agent rationality can be used to design the model backwards. The mechanism designer defines rules governing the interaction between agents such that the system in equilibrium represents a desired outcome usually maximizing social welfare. This desirable outcome is achieved with the help of a social choice function constructed in a way that for every combination of individual player types (or whatever input is relevant for the scenario - preferences, judgements, welfare, etc.) there is a Nash equilibrium with the desired outcome. The social choice function itself aggregates the inputs of all players into a social choice for a single outcome. An incentive-compatible social choice function cannot be strategically manipulated by a player (i.e., no single player can ensure a certain outcome by strategically misrepresenting its type).

The revelation principle of mechanism design [105], posits that any arbitrary mechanism implementing a particular social choice function and its equilibrium outcome (payoffs) can be replicated by an incentive-compatible direct mechanism implementing the same function in which all participants have incentives to reveal information truthfully [91, 96]. In a direct mechanism a player's set of available actions is the set of their possible preferences. A mechanism is characterized as incentive-compatible if every participant achieves the best possible (or at least not worse) outcome just by following their own preferences [104]. The revelation principle a key piece in finding solutions as it narrows the search field making finding a mechanism easier.

Probably the most common game designed is a single shot auction. In such a game players' strategies are represented by the bid they submit to procure a resource. The outcome for a player is 0 if the auction is lost and $v_i - p$ if the auction is won where v_i is the player's valuation of the resource and p is the price at which the resource was won. However, an auction game in which the resource is awarded to the highest bidder and the price of the resource is the value of the bid does not have a dominant strategy solution. In fact, the behavior of players in such a game will be unpredictable as the decision on what bid to submit is made based on what the player believes to know about the other players.

An auction game with a dominant strategy solution is a second price auction [143] also known as a Vickrey auction or Vickrey mechanism. In a Vickrey auction players submit sealed bids to a trusted central party without any knowledge of the strategies of the remaining players. The highest bidder wins the (single indivisible) item and pays the value of the second highest bid for it. In this setting the dominant strategy for each player is the truth - the best action for a player is to submit a bid that corresponds to their true valuation of the auctioned item or resource. An asymmetric game (where players have different strategy sets) with private information where truthfully revealing their information is a weakly-dominant strategy for each player is called strategyproof or truthful. Strategyproof mechanisms include majority voting (majority rule - one of two alternatives receives more than half the votes), second-price auctions and all Vickrey-Clarke-Groves mechanisms (incentive-compatible generic mechanisms for achieving a socially-optimal solution in auction of multiple items).

Thus, systems stemming from mechanism design induce truth-telling from players even though they get to maintain their private information and ensure a desirable social welfare maximizing outcome that keeps the system in equilibrium. The field of economics is the origin of Nobel Prize winning work on resource allocation, auctions, equilibria, and mechanism design [18, 98, 58, 143, 33, 67, 91, 96]. Today, mechanism design has also found many applications in market theory, asset auctioning and allocation, supply chains, taxation, elections, government regulation, politics, logistics, transportation, network routing and resource allocation, among others. In transportation in particular, mechanism design has been applied to congestion theory [144], airport time slot auctions [122], urban road pricing [145], congestion pricing and tradable credit schemes [142, 152], tradable permits and ride sharing [55, 56, 147], commuter assignment [111], ramp control [13, 137], traffic flow control [53], parking slot assignment [156], dynamic road pricing [52], dynamic traffic assignment [21], collaborative logistics [77, 78], transportation preference elicitation [54], enterprise transportation outsourcing [24]. The next section presents an application of mechanism design to lane changing on a multi-lane road.

CHAPTER 6

GAME THEORETICAL MECHANISM FOR MULTI-OBJECTIVE ARGUMENTATION AMONG DRIVERLESS VEHICLES

For the purposes of this work, traffic on a shared road is viewed as a multi-player multi-round game of incomplete information where agents have no private information and no assumptions about others, similar to a Bayesian game [57] without a Common Prior Assumption [6]. The game could be considered as both cooperative and adversarial as vehicles need to agree to play by the rules and abide by the outcomes but players can also try to gain a competitive edge. Each smart vehicle in traffic is a player in the game and possesses certain private information and preferences related to objectives it maintains. An agent has little public information about other players on the road and potentially no information about their private preferences and goals. In order to participate and share resources, beyond a source and destination, a vehicle must select the manner in which it travels. In every round, in concert with its objective, a player makes strategy choices through which it bids for resources, possibly against others. A vehicle taking up a resource (e.g., a spatial position in a certain lane and driving at a certain speed) means another agent's use of resources may be impeded. Thus, conflict is part of the game and it needs to be resolved in a way that is agreeable and/or beneficial to players sharing the road. The aim of the game is to deliver the most combined utility for a group of smart vehicles while allowing them to take actions that will satisfy their individual goals. This is achieved through a truth-inducing lane and speed allocation mechanism for AV players on a multi-lane highway where agents bid for lane position and speed setting assignments and receive approval to implement actions that contribute to their individual objectives as well as to a socially beneficial outcome. The model was implemented and verified in NetLogo.

6.1 LANE AND SPEED ALLOCATION MECHANISM FOR DRIVERLESS VEHICLES ON A MULTI-LANE HIGHWAY

6.1.1 Preliminaries

For a mechanism to be employed in a multi-agent system, agents have to agree to abide by its rules, accept allocation decisions handed to them, and make payments to cover all applicable cost incurred. In order to present the general case, this work assumes that all AVs on the section of road governed by the mechanism are willing participants and there are no rogue agents acting independently and/or maliciously. Potentially, road access and usage can be granted under the condition to comply with the mechanism. Autonomous agents acting in good faith would have no reason to resist, since the mechanism guarantees maximal achievable social welfare at every round and cooperation will only improve overall performance and fair resource allocation.

AVs will need to be able to communicate with their environment to relay bids and data. Before any conflict resolution over resource allocation claims can take place, similar to task allocation algorithms, agents must communicate their situational awareness (surrounding vehicles speed and lane position) and agree with everyone else on the sensed data. However, this is data that can be readily sensed and should have little deviation among agents. An obvious workaround to requiring consensus on sensed data is for every vehicle to report its own lane position and speed to its immediate surroundings. Vehicular agents have no incentive to be dishonest while they are in traffic as the information they report is public and can be verified by other agents nearby and be determined to be untrue. Reporting false information can be further disincentivized by penalties that can take monetary or other forms. A significant amount of time and transferred data volume is saved when the requirement to converge on situational awareness data is taken out of the equation.

Data transfer volume is also influenced by the mode of information exchange within the system, the size of the system itself, and its grade of connectivity. The presented mechanism is a de facto auction, which can be generally implemented in a distributed manner, but in this VANET setting administration may be left to an appointed arbiter. The arbiter is the manager of the mechanism and sets objectives and policies that affect who is correct during conflicts among agents. The selected policies for the game are algorithms for conflict resolution and these algorithms ensure equilibrium outcomes. Having an arbiter has other positive side effects like maintaining the privacy of preferences (even though it does not provide a competitive edge in a strategyproof mechanism, privacy is preferred), limiting data transfer volume, and greatly reducing the required degree of connectivity for the local system. A centralized arbiter at the edge of the network can provide computational power, streamlined decision-making, and ensure the reliability of arbitration outcomes. In order to minimize the effect of network topologies with limited connectivity that cannot guarantee the effective propagation of bids to the auctioneer, auctions can be efficiently run within the set of direct neighbors [66, 133]. Indeed, as lane changes and speed adjustments in a traffic configuration only affect vehicles in the immediate vicinity, the mechanism manager will evaluate a relatively low number of bids before making a decision, so scalability is provided. Similar to auction algorithms [14], the mechanism can be implemented when in the absence of a dedicated arbiter (e.g., a cell tower base station with coverage over a portion of the road) one agent is designated or incentivized to accept the additional task of manager. Manager appointment is currently out of scope of this work but how ever mechanism management is assigned, it is kept local at the edge of the IoT. All agents will register with and abide by the decision of the authority responsible for the section of the road they are currently on. The NetLogo simulation discussed in section 6.4 is written for a single manager.

6.1.2 A Bird's Eye View

In the presented model AV agents maintaining private objectives travel on a multi-lane road. The road is partitioned in segments, which include a finite number of vehicles. Each segment is governed by the mechanism. The mechanism unfolds in rounds.

During every round, each driverless vehicle agent senses its surroundings and from its current state, determines the feasibility of potential actions pertaining to lane occupancy and speed setting. The vehicle can request to travel in any lane and at any speed but changes have to be solicited from and approved by the mechanism manager and in some cases will incur a small cost. The AV then consults its governing objective (vehicles may maintain an arbitrarily large number of objectives but for simplicity here they only have one main objective which is the basis for decision making). The governing objective computes the desirability of available actions and assigns them unique numeric values. The values effectively rank preferences but also represent the highest price the agent is willing to pay to implement a certain action. They are the player's strategy for the round. This private valuation is used to bid for lane and speed assignments. The bid accompanied by other necessary information is sent to the mechanism manager. The manager collects all bids and detects which, if any, potential actions an agent is bidding for are in conflict with the desired actions of others. The mechanism resolves any present conflict by granting lane occupancy and speed setting to each agent in a way that ensures the maximal welfare of the entire group. In some cases, the mechanism will collect payment. An agent will only incur cost if its approved actions present an externality on the system. Vehicles implement the awarded actions and after a short adjustment period, the cycle repeats.

The question emerges, what is the best strategy a driverless vehicle agent can play to make sure it is awarded one of its top action choices? Can an agent influence the cost of actions by strategically altering its bid? The mechanism presented in this work will show that the best strategy is always the truth.

6.1.3 The Environment

Consider a multi-lane expressway simulated in the simplest way. Lanes have characteristics pertaining to (i) *maximum speed* and (ii) *minimum speed* with the stipulation that speed limits increase in the higher lanes. Vehicles on the road maintain a governing objective. Relevant characteristics for the agents include their (i) *current speed*, (ii) *preferred speed*, and (iii) a level of *objective emphasis*. The preferred speed and objective emphasis levels determine the target speed and target lane for each player. On the road, every vehicle maintains a required safety distance (buffer zone) from the vehicle directly ahead of it.

6.1.4 The Density of Agents

The buffer zone requirement puts a natural upper bound on the number of vehicles that can travel in each lane. Depending on lane speed limits, the buffer zone size may fluctuate.

Lemma II.1. The maximum number of vehicles n of uniform length c allowed in a single lane segment of length L is $n = \left\lfloor \frac{L}{c+d} \right\rfloor$ where d is the required safety distance of the lane.

Proof. The space that an agent occupies in the lane is its own length plus the required safety distance. For n agents $nc + nd \leq L$ or $n \leq \frac{L}{c+d}$. Since n is a positive integer, $n = \left\lfloor \frac{L}{c+d} \right\rfloor$. □

Corollary II.1. Let a road segment of length L have m lanes with required safety distances d_i for each lane, $i \in (1, \ldots, m)$. For vehicles of varying length c_j with $j \in (1, \ldots, n)$, the maximum number of vehicles allowed on the road segment is $n = \sum_{i=1}^{m} \left\lfloor \frac{L-C_i}{d_i} \right\rfloor$ where C_i is the combined length of allowed vehicles in lane i.

Proof. For a single lane i, the combined length of allowed n_i vehicles is $C_i = c_1 + \cdots + c_{n_i}$. Then $\sum_{j=1}^{n_i}(c_j + d_i) \leq L$, or $n_i \leq \frac{L-C_i}{d_i}$. Therefore, $n_i = \left\lfloor \frac{L-C_i}{d_i} \right\rfloor$. For all lanes combined $n = \sum_{i=1}^{m} n_i = \sum_{i=1}^{m} \left\lfloor \frac{L-C_i}{d_i} \right\rfloor$. □

6.1.5 The Feasibility of Actions

Vehicles sense the environment and make decisions on the feasibility and desirability of potential actions pertaining to speed settings and lane positions. They do this with the help of their governing objective. This objective and the emphasis the vehicle places on it have a direct effect on the way an agent makes decisions and how it perceives the decisions of others. The perception of how much a potential action advances the agent's main goal will result in its quantitative ranking.

Agents in the system make decisions along two dimensions - lane and speed. Thus, consider possible atomic actions for all vehicles representing lane/speed action combinations: (1) *move up and decelerate*, (2) *move up and maintain speed*, (3) *move up and accelerate*, (4) *stay in the current lane and decelerate*, (5) *stay in the current lane and maintain speed*, (6) *stay in the current lane and accelerate*, (7) *move down and decelerate*, (8) *move down and maintain speed*, and (9) *move down and accelerate.* The conditional rule sets pertaining to these actions consider system priority, basic physical limitations, and environmental constraints. The mechanism for derivations of feasible actions was shown in Algorithm 1 in chapter 4. Recall that an agent can potentially change lanes if the desired lane is within the bounds of the environment and the spatial position the agent seeks to occupy in it is not obstructed, i.e., not currently occupied by another agent and there are no other agents within the buffer zone cone that represents safe space so that potential collisions with decelerating vehicles ahead and accelerating vehicles behind in the target lane are avoided. Acceleration and deceleration are contingent upon the availability of space ahead or behind coupled with the speed of blocking vehicles. The option to maintain the current lane and the current speed is presumed to be always available. Again, all speed decisions are preempted by a collision avoidance mechanism.

With the feasibility of separate lane and speed actions ascertained, the AV agent can determine the availability of atomic lane/speed action combinations as defined by the system. Once all feasible atomic actions are identified, they are ranked. The ranking is decided by rule sets for each objective taking into account the target lane and speed, the current speed, the current traffic lane and its speed restrictions, and the physical availability of space as determined by the neighboring agents. All players rank action combinations depending on where their current speed setting and lane position are in relation to their targets.

Next, consider the model's theoretic foundation.

6.1.6 Agents, Preferences, Strategies, and Outcomes

Assume that on a section of road with a designated manager there is a set (including the manager, if applicable) of moving self-interested driverless vehicle agents I adhering to safety distance requirements. Also assume that each agent $i \in I$ has a type $\theta_i \in \Theta$ drawn from a set of possible types Θ. A *type* is the private information and preferences of the agent within the context of its dominant objective. The type of an agent is static as long as its objective remains unchanged. If the vehicle is to change objectives, its preferences will also change. Some information in θ_i is constant - e.g., the make, model, physical dimensions and speed capabilities of the vehicle.

In general, and setting aside the possibility to form coalitions for the moment, it can be assumed that the player is selfish and makes decisions aiming to maximize its own utility. In every round, a player i uses its objective's rule sets to assign valuations to all currently feasible actions. These valuations over all currently available actions over ranked outcomes represent the highest price the agent is willing to pay to implement the corresponding atomic action. This makes up the main portion of a player's strategy, which is submitted to the mechanism manager in a bid to receive approval to implement what the agent intends to do next. For example, consider a smart vehicle i whose priority is to minimize its own travel time. If the vehicle aims to reach a target speed but it currently finds itself in a middle lane travelling at a speed close to the lane's limit, it has to consider feasible outcomes o_i from a set of possible outcomes $\mathcal{O}$. The first and most desirable outcome, o_1, would be to move to a lane with a higher speed limit where i can accelerate. Should that action not be available because the mechanism does not grant approval, the vehicle could arrive at outcome o_2, in which it accelerates in its current lane. If not allowed, the next best outcome, o_3, is to maintains speed and continue travelling in the current lane for the next cycle and then attempt to make changes that lead to a more desirable outcome in the following rounds. Other less beneficial but nevertheless potentially available options are to decelerate in the current lane (o_4) or to move to the lane down, where it might have to decelerate (o_5). Obviously, in accordance with the "get there faster" objective, the vehicle can rank the desirability of these available outcomes as $o_1 > o_2 > o_3 > o_4 > o_5$, and the utility derived from each outcome follows the same ordering $u_i(o_1, \theta_i) > u_i(o_2, \theta_i) > u_i(o_3, \theta_i) > u_i(o_4, \theta_i) > u_i(o_5, \theta_i)$. This order of utilities for the given outcomes allows the agent to assign a certain *valuation* $v_i = (v_1, ..., v_5)$ to each of the 5 atomic actions (lane and speed action combinations) it can take where the valuation of an action is a positive scalar within a mechanism defined range, e.g., $[0, 1]$. Since o_1 is the most desirable outcome, the action tied to it, move to a lane up and accelerate, will be valued highest, while moving to a lane down and decelerating will receive the lowest valuation. The valuation of available actions determines the *strategy* $s_i(\theta_i)$ of agent i, where θ_i contains v_i and other private relevant information. The set S_i of all strategies of an agent determined by its objective makes up the agent's *strategy profile.*

Sometimes, not all outcomes will be attainable because not all actions are available to an agent. Should agent i be travelling in the leftmost lane (for right-hand traffic), the option to move a lane up would be unavailable and the valuation assigned to it will be 0 signalling that the choice is not in play. Other natural and synthetic constraints like speed limit, vehicle capabilities, lane availability, lane position, and road restrictions, among others, may limit the set of strategies available to an agent. Agent i's *feasible strategy profile* FS_i is the set of all possible available strategies $fs_i(\theta_i)$. All strategies are aimed at maximizing the agent's own utility and do not take into account the combined utility of the group or any information about the types or actions of other players.

Definition II.6. Let $i \in I$ be an autonomous self-interested driverless vehicle agent with a type θ_i, where the type contains public and private information about the agent, including its valuation of n potential actions $v_i = (v_1, \ldots, v_n)$ derived from ranking projected outcomes of these actions within the context of the agent's current governing objective.

Agent i's *strategy profile* S_i is the set of all available *strategies* $s_i(\theta_i)$.

Agent i's *feasible strategy profile* FS_i is the set of all *feasible strategies* $fs_i(\theta_i)$ as determined by some set of scenario and environment constraints.

Each vehicle i in the designated area of a manager "plays" a feasible strategy $fs_i(\theta_i)$ in the form of a bid at some regular time interval (a cycle). A cycle represents a round of the game and is made up of a certain number of time slices. In each round, the manager uses part of the time slices to gather the bids, extract the configuration that maximizes the attainable social welfare in its specific area, and signal the approved actions to each player. The remaining time slices are used for implementation of actions, adjustment period, forward progression and valuation derivation of potential actions. The strategy profile of the round $fs = (fs_1(\theta_1), ..., fs_I(\theta_I)) = (fs_i(\theta_i), fs_{-i}(\theta_{-i}))$ is used to select an outcome $o \in \mathcal{O}$ delivering the highest achievable combined utility. The selection is made with the help of a *social choice function*.

Definition II.7. A *social choice function* $f : FS \to \mathcal{O}$ selects an outcome $o \in \mathcal{O}$ from the set of feasible strategies $fs = (fs_1(\theta_1), \ldots, fs_I(\theta_I))$ submitted as bids by all I agents.

A social choice function essentially maps each feasible strategy profile to a single outcome. However, note that the social choice function assumes full information about agents and their preferences.

The utility an agent experiences from driving on the mechanism-controlled road is $u_i(o) = v_i(o) - p_i + c_i$ where $c_i > 0$ is some fixed utility attributed to forward progression and p_i is a mechanism specific payment that may have been incurred for implementing the approved action. The approved operation will always be one of the ranked actions submitted by the player.

Conflict could arise when a vehicle's potential lane/speed action would result in a lane position that coincides or interferes with the lane position another vehicle is vying for during the same cycle. Given the position an agent occupies at the beginning of the bidding time slice and the information it submits about intended actions, the projected position at the end of the action implementation time slice can be readily calculated.

Within this mechanism, for each round, every vehicle in the group ranks potential actions along two dimensions - lane and speed. A player is given a choice of n possible (but not always feasible) predetermined actions regarding speed - e.g., accelerate by 5 miles per hour, maintain speed, decelerate by 5 miles per hour, decelerate by 10 miles per hour, etc. Along the lane dimension, there are m theoretically available actions for lane changes - remain in the same lane, move to a lane up, move to a lane down, move up two lanes, etc. The agent determines the valuation of each possible lane/speed atomic action combination as seen through the prism of its main objective. Therefore, at time slice t the feasible strategy $fs_i^t(\theta_i)$ played by the agent contains information about its current position pos_i^t, velocity, physical dimensions, and a lane/speed valuation matrix v_i^t as described in Definition II.8.

Definition II.8. For an autonomous vehicular agent i with mechanism predetermined m possible lane actions and n possible speed actions, the $m \times n$ matrix

$v_i^t = \begin{pmatrix} (v_i^t)_{11} & \dots & (v_i^t)_{1n} \\ \vdots & \ddots & \vdots \\ (v_i^t)_{m1} & \dots & (v_i^t)_{mn} \end{pmatrix}$ is the *valuation* of all feasible lane/speed action combinations agent i includes in the bid submitted to the mechanism manager during time slice t. Additionally, $\forall (\mathrm{v}_i^t)_{xy} \in [0,1]$, $(v_i^t)_{xy} = 0$ iff either or both lane action x and speed action y are unfeasible, and for all feasible action combinations xy and wz $(v_i^t)_{xy} \neq (v_i^t)_{wz}$, where $x, w \in \{1, \dots, m\}$ and $y, z \in \{1, \dots, n\}$ and $x \neq w$, $y \neq z$.

In other words, the valuation matrix consists of positive scalars with distinctive values which signal the current strength of intent a vehicle has to implement the corresponding actions and the maximum price it is willing to pay to do so.

The numeric valuations are assigned according to the current state and objective preferences of the agent. Lane moves are prioritized over speed settings so an agent will always choose to make a change in the direction of the target lane over staying in the current lane over moving in the opposite direction. The speed preference is set according to its feasibility given the potential lane action, lane speed constraints, and target speed setting. Thus, feasible actions that bring the vehicle closer to its target lane or keep it in the target lane receive higher valuations. Multiple atomic actions that result in the same lane position are uniquely ranked according to their potential to reach or keep the target speed.

Note that even though Definition II.8 contains language pertaining to the driverless vehicle application domain, the valuation in its present form can be used as a bid for any two-dimensional divisible resource. As a matter of fact, more dimensions can be added by adding matrices, or the valuation can be scaled down to a vector.

6.1.7 Conflict and Conflict-Free Action Allocation

Using the timestamped bids submitted by all agents in its area, the manager extracts the relevant information and calculates a position matrix $pos_i^{t'} = \begin{pmatrix} (pos_i^{t'})_{11} & \dots & (pos_i^{t'})_{1n} \\ \vdots & \ddots & \vdots \\ (pos_i^{t'})_{m1} & \dots & (pos_i^{t'})_{mn} \end{pmatrix}$. Each element of $pos_i^{t'}$ represents the potential physical position (relative or absolute depending on the type of positioning employed) a vehicle would occupy at the end of the implementation time slice t' should the corresponding plausible lane/speed action combination from the valuation matrix be granted. Note that for each agent i there is a one-to-one correspondence between elements in v_i^t and $pos_i^{t'}$. Every element of $pos_i^{t'}$ is in the form (Lx, f, r), where x is the lane number, and f and r are the projected spatial positions of the front and rear bumpers of the vehicle respectively.

The mechanism determines assignment allocation for each vehicle along both dimensions - lane and speed. It calculates the set L of all possible assignments of lane actions for all vehicles. Since every vehicle can be assigned one of m lane actions, then $L = \{l_{11}, \ldots, l_{1m}; \ldots; l_{I1}, \ldots, l_{Im}\}$ where l_{ij} is an m-dimensional vector and $i \in I$. Also, $\forall k \in \{1, \ldots, m\}$ $(l_{ij})_k = 1$ iff $k = j$, $(l_{ij})_k = 0$ if $k \neq j$ and $\sum_{k=1}^{m} (l_{ij})_k = 1$. This means that each vector l_{ij} has a single non-zero element and that element is equal to 1 (or can be replaced with a multiplier the mechanism designer puts into place to scale payment as appropriate). Similarly, the set of all possible assignments of speed actions is $S = \{s_{11}, \ldots, s_{1n}; \ldots; s_{I1}, \ldots, s_{In}\}$ with the same characteristics - s_{ij} is an n-dimensional vector, and for $\forall k \in \{1, \ldots, n\}$ $(s_{ij})_k = 1$ iff $k = j$, $(s_{ij})_k = 0$ if $k \neq j$ and $\sum_{k=1}^{n} (s_{ij})_k = 1$. In other words, each player is assigned a single lane and a single speed action.

Next, while calculating the maximum social welfare of the group, the manager detects conflict for lane positions. Conflict occurs when potential actions, when undertaken by a pair of vehicles, would result in lane positions that are incompatible, i.e., they cannot occur at the same time because they would either occupy overlapping space or violate mechanism buffer zone constraints.

Definition II.9. An outcome $(l, s) = (l_1, s_1; \ldots; l_I, s_I) \in \mathcal{O}$ with $l_i \in L$ and $s_i \in S$ is a *lane and speed allocation* where $\forall i, j \in I$, $i \neq j$ and agent i is not in conflict with agent j, i.e. $|pos_i^{t'} s_i l_i \ominus pos_j^{t'} s_j l_j| >= d$ where $\ominus$ is an operation calculating the distance between two positions in the same lane and d is a mechanism determined required safety distance between vehicles in the same lane.

For a given lane and speed allocation (l, s), $pos_i^{t'}(l, s) - pos_j^{t'}(l, s) =$

$$= pos_i^{t'} s_i l_i \ominus pos_j^{t'} s_j l_j = (Lx, f_i, r_i) \ominus (Ly, f_j, r_j) = \begin{cases} \infty & \text{if } x \neq y, \\ \mathrm{f}_j - r_i & \text{if } x = y,\ f_j \geq f_i, \\ \mathrm{f}_i - r_j & \text{if } x = y,\ f_j < f_i \end{cases}$$

Therefore, if two or more vehicles are vying for coinciding or intersecting spacial position on the road, the manager must calculate the maximum attainable utility, appoint the winner of the debatable position and assign the losing agent(s) conflict-free actions of lesser valuation. The resulting speed and lane allocation contains one vector from L and one vector from S for each agent.

Note that the operation $\ominus$ is specific to the application domain. It has to be redefined as appropriate to use in other fields.

The optimal allocation outcome $(l, s) \in (L, S) \in \mathcal{O}$ extracted by the social choice function where all agents submit their true valuations is obtained by solving $(l, s) = argmax_{l \in L, s \in S} \sum\limits_{i=1}^{I} v_i^t(l, s) = argmax_{l \in L, s \in S} \sum\limits_{i=1}^{I} v_i^t s_i l_i$ where $\forall pos_i^{t'}(l, s)$ is conflict free within the allocation.

The maximum exists, because solving for *argmax* delivers a non-empty set since the status quo where all vehicles maintain speed and lane positions always exists.

Observation II.1. Since (l, s) is the optimal truth telling conflict-free allocation, $\nexists (l', s')$ that improves social welfare, i.e. $\sum_{i=1}^{I} v_i^t(l, s) \geq \sum_{i=1}^{I} v_i^t(l', s')$.

6.1.8 Prices and Payouts

For each agent the optimal obtainable allocation result $v_i^t(l, s) = v_i^t s_i l_i$ is the maximum valuation for a combination of lane/speed actions at bidding time slice t, the implementation of which results in a projected position that is either not disputed or is won by the agent. The utility the agent receives is $u_i^t(l,s) = v_i^t(l,s) - p_i^t + c_i$ where p_i^t is a potential cost incurred by agent i for taking the mechanism approved lane and speed actions at the end of the implementation time slice. The price p_i^t a player $i \in I$ pays for a bid at time slice t is determined by the difference in social welfare when the agent is not present in the group and when it is.

Definition II.10. The *mechanism specific price* $p_i^t = \sum\limits_{j=1, j\neq i}^{I} v_j^t(l,s)_{-i} - \sum\limits_{j=1, j\neq i}^{I} v_j^t(l,s) \geq 0$, where $(l,s)_{-i}$ is the optimal conflict-free allocation for the group without taking agent i's presence into consideration and (l,s) is the optimal conflict-free allocation for the whole group (but for the term the position assigned to agent i is de facto unavailable to all other agents).

In a matter of speaking, the price an agent pays is the cost of inflicting itself on the remaining group, the "damage" it imposes on its society. In mechanism design, this is known as the Clarke pivot rule [26]. The price is always non-negative because an agent's presence and reported preferences may disrupt the configuration of the remaining agents or may be completely independent of it. In other words, if a player changes the combined utility of the remaining group with its presence, it pays a price.

Definition II.11. The *utility* of an agent for a time slice t and allocation (l,s) is $u_i^t(l,s) = v_i^t(l,s) - p_i^t + c_i = v_i^t(l,s) - (\sum\limits_{j=1, j\neq i}^{I} v_j^t(l,s)_{-i} - \sum\limits_{j=1, j\neq i}^{I} v_j^t(l,s)) + c_i = \sum\limits_{i=1}^{I} v_i^t(l,s) - \sum\limits_{j=1, j\neq i}^{I} v_j^t(l,s)_{-i} + c_i$.

6.1.9 Lying as a Strategy

Autonomous vehicle agents were assumed to be rational and selfish, so theoretically there would be nothing stopping them from trying to gain an edge by falsifying the time slice bid they submit to the manager. Misrepresenting physical information is futile as it can be readily sensed and penalized but what about lying about valuation? Can an agent purposefully misreporting valuations fraudulently win the lane/speed action, diminish its incurred mechanism-specific price, or drive up its utility?

Proposition II.1. No agent can be better off by lying.

Proof. Should an agent i lie about its true valuation v_i^t in order to try to gain an edge and instead submit a valuation $(v_i^t)'$ while all other agents submit their true preferences, an alternative allocation (l', s') will emerge.
The new utility of agent i thus becomes $(u_i^t)'(l', s') = \sum_{i=1}^{I} v_i^t(l', s') - \sum_{j=1, j \neq i}^{I} v_j^t(l, s)_{-i} + c_i$. However, it was already shown in Observation II.1 that the optimal allocation where all agents divulge their true valuations maximizes social welfare. Thus, since $\sum_{i=1}^{I} v_i^t(l, s) \geq \sum_{i=1}^{I} v_i^t(l', s')$ and the second and third utility terms are common and constant for each agent i, it follows that no agent can be better off by lying about its true preferences. $\square$

Observation II.2. Truth telling in this case presents a dominant strategy since it maximizes the utility of an agent regardless of what strategies the other agents play. This is beneficial, given that players have no knowledge of the preferences of others. Since all agents are best served by telling the truth, the game has a dominant-strategy equilibrium, which is also a Nash equilibrium.

6.1.10 The Mechanism

A social choice function maps the true preference of a group of agents to an outcome and depends on agents telling the truth. The social choice function $f : FS \rightarrow \mathcal{O}$ (previously presented in Definition II.7) selects an outcome (l, s) given full information about agents' preferences $fs = (fs_1(\theta_1), \ldots, fs_I(\theta_I))$ where $fs \in FS = FS_1 \times \cdots \times FS_I$.

A mechanism utilizes an outcome function to produce a social utility maximizing allocation assignment that depends on the preferences of the agents. The preference information however, is private and a self-interested agent may misreport it in an effort to achieve a better outcome for itself. Thus, since truthfulness is not guaranteed, a mechanism is constructed to map the *reported* type of an agent to a desired outcome. In particular, a mechanism that returns the same outcome that would have been reached had the agents been telling the truth about their preferences is said to implement the social choice function.

Definition II.12. A mechanism $\mathcal{M}(S, g)$ *implements the social choice function* f if there is an equilibrium strategy profile s^*, s.t. $g(s^*) = f(fs)$, $\forall fs \in FS$.

In other words, each agent is afforded a set of strategies S_i from which it selects one and submits it to the manager during the time slice. The strategy may reflect the truth, but it also may not, since each agent is trying to maximize its own utility. The strategy set for all agents $s = (s_1(\theta_1), \ldots, s_I(\theta_I)) \in S = S_1 \times \cdots \times S_I$ is used by the mechanism outcome function $g : S \to \mathcal{O}$ to produce a result. If no agent has an incentive to change its chosen strategy, the strategy set is an equilibrium $s^* = (s_1^*(\theta_1), \ldots, s_I^*(\theta_I))$. If the outcome of that equilibrium produced by the outcome function is the same as the outcome produced by the social choice function, the mechanism $\mathcal{M}(S, g)$ implements the social choice function.

Definition II.13. A mechanism $\mathcal{M}(S, g)$ is a *direct revelation mechanism* if $S_i = FS_i$ for $\forall i$ and $g(fs) = f(fs)$ for $\forall fs \in FS$.

Definition II.13 posits that a direct revelation mechanism does not put a restriction on the strategy space of the players and achieves the same outcome regardless of the truthfulness of played strategies. Recall that the Revelation Principle of mechanism design postulates that if the full information social choice function can be implemented by a mechanism, the mechanism is direct and every agent has an incentive to reveal its true preferences [96, 91]. In that case, the social choice function is said to be *incentive-compatible*. If a mechanism is characterized as incentive-compatible, every participant achieves the best possible outcome just by following its own true preferences.

Definition II.14. A direct revelation mechanism $\mathcal{M}(S, g)$ implements an *incentive-compatible* social choice function if it has a dominant strategy equilibrium $s^* = (s_1^*(\theta_1), \ldots, s_I^*(\theta_I))$ s.t. $s_i^*(\theta_i) = fs_i(\theta_i)$ for $\forall i \in I$.

Definition II.15. A direct revelation mechanism $\mathcal{M}(S, g)$ where the truth is a weakly dominant strategy for each player regardless of what the remaining players do implements a *strategy-proof* social choice function.

Proposition II.2. Mechanism $\mathcal{M}(FS, g)$ where $g : FS \to (L, S)$ for $fs = (fs_1(\theta_1), \ldots, fs_I(\theta_I)) \in FS$ is incentive-compatible, strategy-proof and induces truth telling of agents.

Proof. Mechanism $\mathcal{M}(FS, g)$ does not restrict the strategy space of agents and delivers the same outcome as the social choice function so it is a direct revelation mechanism. It was already shown in Proposition II.1 that no agent can be better off by lying about its lane/speed plausible action valuation, so the mechanism induces truth-telling of agents. Per Observation II.2 truth telling was shown to be a weakly dominant strategy for each player regardless of what others do. The game thus has a dominant strategy equilibrium coinciding with the true valuations of all agents. Therefore $\mathcal{M}(FS, g)$ implements an incentive-compatible strategy-proof social choice function. □

Mechanism $\mathcal{M}(FS, g)$ is a variation of a Vickrey-Clarke-Groves mechanism [143, 26, 50]. It implements an efficient outcome in dominant strategies. The dominant strategy of each player is the truth and it is independent of the choice of strategy of the remaining players. The Clarke pivot rule [26] ensures that each player is charged its externality, namely the difference in social welfare when the player is absent and when the player is present. Since all valuations are non-negative, the players always get a net positive utility and the mechanism always charges a non-negative price. This makes the mechanism a win-win game - players get to undertake desirable actions and gain more than they pay and the mechanism receives a net positive payment while providing a service.

6.2 EXAMPLE SCENARIO

Consider the following simplified scenario as shown in Figure 6.1 where three vehicles travel in immediate vicinity of one another on a three lane highway under the authority of the same mechanism manager. Lanes have minimally overlapping minimum and maximum allowed speed ranges. Vehicle 1 drives in the fastest lane at the speed limit for the lane, vehicle 2 travels in the middle lane at its target speed, and vehicle 3 is in the slowest lane at a speed below its target. Each vehicle is 15 feet long. For safety reasons, the mechanism enforces a minimum distance of two car lengths (30 feet) between vehicles in the same lane. Additionally, the current governing objective of v_1 and v_3 is minimizing travel time while v_2's main objective is travelling in a minimally congested lane.

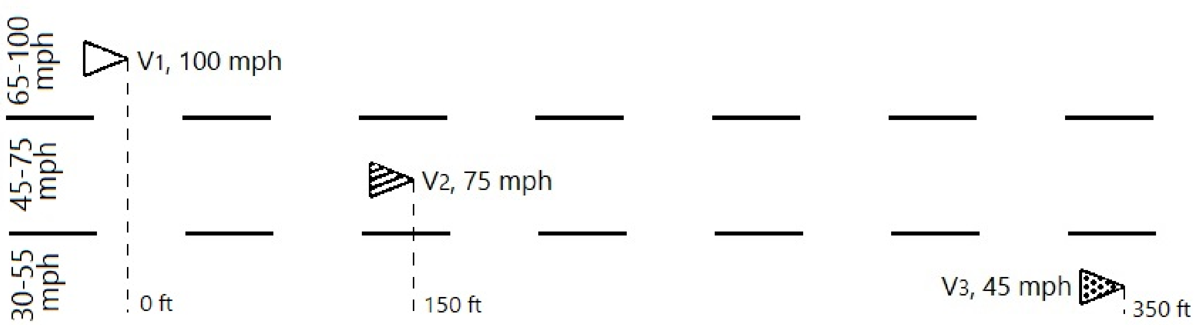

Figure 6.1: Mechanism Design for Lane and Speed Allocation Among Driverless Vehicles on a Highway Example

Let the mechanism afford all agents three potential actions for lane changes - move to a lane up, stay in the current lane, move to a lane down; and three potential actions for speed changes - decelerate by 5 miles per hour, maintain speed, accelerate by 5 miles per hour. For simplicity, for each cycle (round of the game) the mechanism accepts bids and evaluates them and the vehicles implement the granted actions in a single time slice. The time slice duration for this example is set at 5 seconds and the time required for data transmission is disregarded. In this scenario, the action valuation matrices the three vehicles formulate in accordance with their main objectives can look as follows (lane changes are recorded in the rows and speed changes in the columns):

$$v_1^t = \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ v_2^t = \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ \text{and } v_3^t = \begin{pmatrix} 0 & 0.8 & 1 \\ 0.1 & 0.4 & 0.5 \\ 0 & 0 & 0 \end{pmatrix}$$

Since vehicle 1 is travelling in the fastest lane, all options to move to a faster lane (row 1 in the v_1^t valuation matrix) are excluded by assigning them a valuation of 0. The exclusion can be made by the manager or the vehicle. The two best available choices for this vehicle according to its governing objective are to either maintain speed, represented by element $(v_1^t)_{22}$, or decelerate by 5 miles per hour $(v_1^t)_{21}$, both in the same lane. The vehicle is already travelling at the speed limit for the lane so the option to accelerate, $(v_1^t)_{23}$, is unavailable. Since the lane down has a maximum speed limit much lower than 5 miles per hour from the agent's current speed, the move to a that lane is currently unavailable until the vehicle decelerates sufficiently. Thus all options in row 3 of the valuation matrix of vehicle 1 are turned off by marking them with 0s. Vehicle 2's preferred course of action, $(v_2^t)_{22}$, is to remain in its current lane and maintain velocity, since its current speed of 75 mph is also its preferred speed. Decelerating in the same lane is the second choice because the new velocity is still close to the preferred speed and the lane is not as congested as the lanes on either side. Moving to the lane up is a less desirable option and delivers lower valuations as shown in the first row of the v_2^t valuation matrix. Moving to the lane down is currently unavailable to vehicle 2, since its speed is too high. Several 5 mph decelerations have to take place before the alternative becomes accessible. Similarly, vehicle 3's top choice is to move up a lane and increase speed, $(v_3^t)_{13}$. Lesser available options are ranked accordingly.

Without restriction of generality, assume that the manager is a cell tower with coverage over the road. The manager receives the timestamped bids, unpacks them and calculates the potential position matrices $pos_i^{t'}$ for every feasible action (with valuation greater than 0) of every agent. Distance calculations can be made with the help of positional road markings, GPS, road sensors, etc. The potential agent locations allow the manager to determine positional conflict between vehicles should a lane/speed action combination be allowed to be implemented. For simplicity, the minor distance loss in forward advancement incurred by an agent changing lanes is disregarded. For the example, distance calculation results are rounded up to the nearest whole number. The front of the last vehicle in the group (vehicle 1) can be considered to be relative position 0 as illustrated in Figure 6.1. The top valuation of the agent is to remain in its current lane and maintain its speed. At its present velocity of 100 mph v_1 will travel 733 feet within the 5 seconds of the time slice. As the length of the vehicle is 15 feet, v_1 would occupy the space between 733 and 718 feet ahead of its current location in lane 3, should it be allowed to implement its top choice actions. Similarly, all other non-zero valuations will receive a relative potential position, while any turned off lane/speed option is marked as -1. Thus the resulting positional matrices are:

$$pos_1^{t'} = \begin{pmatrix} -1 & -1 & -1 \\ L3,697,682 & L3,733,718 & -1 \\ -1 & -1 & -1 \end{pmatrix} \text{ corresponding to } v_1^t = \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$pos_2^{t'} = \begin{pmatrix} L3,663,648 & L3,700,685 & L3,737,722 \\ L2,663,648 & L2,700,685 & -1 \\ -1 & -1 & -1 \end{pmatrix} \text{ to } v_2^t = \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \text{ and}$$

$$pos_3^{t'} = \begin{pmatrix} -1 & L2,680,665 & L2,717,702 \\ L1,643,628 & L1,680,655 & L1,717,702 \\ -1 & -1 & -1 \end{pmatrix} \text{ to } v_3^t = \begin{pmatrix} 0 & 0.8 & 1 \\ 0.1 & 0.4 & 0.5 \\ 0 & 0 & 0 \end{pmatrix}.$$

The potential spatial positions with their valuations are shown in Figure 6.2.

The mechanism calculates the maximum attainable social welfare when all conflict is resolved. The highest possible collective utility of 3 is unachievable since the distance in

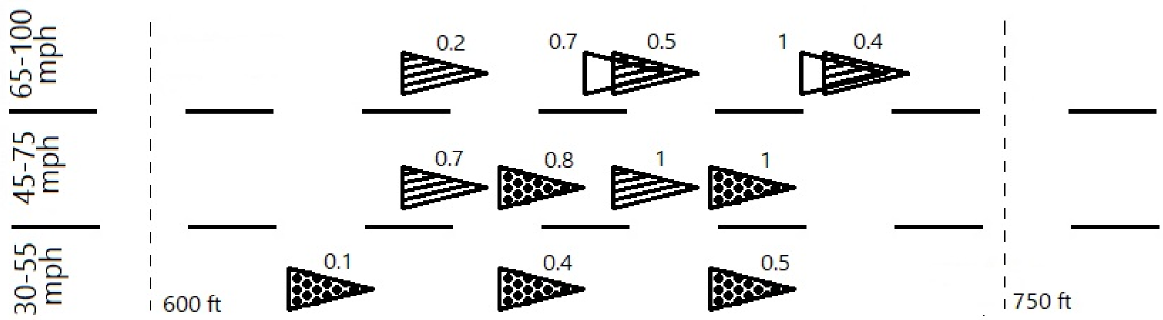

Figure 6.2: Mechanism Design for Lane and Speed Allocation Among Driverless Vehicles on a Highway Example - Potential Positions at the End of the Time Slice

top valued potential positions for v_2 and v_3 in lane 2 would violate the built-in mechanism safety distance of 30 feet. The most social welfare that can be extracted from this configuration is 2.7. Conflict is resolved and as a result, v_1 and v_3 are allowed to implement their top choice of actions and v_2 receives approval for its second choice. The optimal allocation becomes $(l, s) = (\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix})$.

Indeed, this is the best allocation. Of the 50 unique allocation candidates, there are 19 viable conflict-free configurations. Aside from the winning allocation, the remaining 18 result in combined group utility ranging between 1.3 and 2.5 as shown in Table 6.1. The dynamics of allocation construction are addressed in more detail in section 6.4.

Next, given the optimal allocation, the mechanism calculates if it is owed any payment for the round.

Price is determined according to the formula presented in Definition II.10. Vehicle 1 pays nothing since in this cycle it does not inflict itself on anyone - the best possible utility of the group for an alternative action allocation without it $(s, l)_{-1}$ is 1.7 and the utility of the allocation determined by the mechanism with the utility of vehicle 1 taken out is also 1.7. Similarly, $p_2^t = 2 - 2 = 0$. However, vehicle 3 moving up a lane and essentially forcing vehicle 2 to decelerate causes some externality on the system. The price vehicle 3 pays is

$$p_3^t = \sum_{j=1, j\neq 3}^{3} v_j^t(l,s)_{-3} - \sum_{j=1, j\neq 3}^{3} v_j^t(l,s) =$$

$$= \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} -$$

$$- \left[\begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right] = 2 - 1.7 = 0.3.$$

Table 6.1: Mechanism Design for Lane and Speed Allocation Among Driverless Vehicles on a Highway Example - Conflict-Free Allocation Candidates

Feasible Allocation	Actions and Valuation Vehicle 1	Vehicle 2	Vehicle 3	Social Utility
1	◯ ▷ 1	◯ ▷ 1	◯ ▷▷ 0.5	2.5
2	◯ ▷ 1	◯ ▷ 1	◯ ▷ 0.4	2.4
3	◯ ▷ 1	◯ ▷ 1	◯ ◁ 0.1	2.1
4	◯ ▷ 1	◯ ◁ 0.7	△ ▷▷ 1	2.7
5	◯ ▷ 1	◯ ◁ 0.7	◯ ▷▷ 0.5	2.2
6	◯ ▷ 1	◯ ◁ 0.7	◯ ▷ 0.4	2.1
7	◯ ▷ 1	◯ ◁ 0.7	◯ ◁ 0.1	1.8
8	◯ ▷ 1	△ ◁ 0.2	△ ▷▷ 1	2.2
9	◯ ▷ 1	△ ◁ 0.2	△ ▷ 0.8	2.0
10	◯ ▷ 1	△ ◁ 0.2	◯ ▷▷ 0.5	1.7
11	◯ ▷ 1	△ ◁ 0.2	◯ ▷ 0.4	1.6
12	◯ ▷ 1	△ ◁ 0.2	◯ ◁ 0.1	1.3
13	◯ ◁ 0.7	◯ ▷ 1	◯ ▷▷ 0.5	2.2
14	◯ ◁ 0.7	◯ ▷ 1	◯ ▷ 0.4	2.1
15	◯ ◁ 0.7	◯ ▷ 1	◯ ◁ 0.1	1.8
16	◯ ◁ 0.7	◯ ◁ 0.7	△ ▷▷ 1	2.4
17	◯ ◁ 0.7	◯ ◁ 0.7	◯ ▷▷ 0.5	1.9
18	◯ ◁ 0.7	◯ ◁ 0.7	◯ ▷ 0.4	1.8
19	◯ ◁ 0.7	◯ ◁ 0.7	◯ ◁ 0.1	1.5

△ move up, ◯ stay, ▽ move down, ◁ decelerate, ▷ maintain speed, ▷▷ accelerate

The manager relays the approved actions to all agents and collects any non-zero payment amounts. Actions are implemented and after a short adjustment period using up the remaining time slices, the process restarts for the next cycle.

For simplicity, the constant utility term c_i is disregarded. Therefore, reporting their true valuations resulted in individual utilities for the 3 players of $u_1^t = 1 - 0 = 1$, $u_2^t = 0.7 - 0 = 0.7$, and $u_3^t = 1 - 0.3 = 0.7$. Now consider an attempt by one of the agents to improve its utility by misreporting preference values. Vehicle 2's governing objective is travelling in less congested lanes, so maybe lying about its valuation for decreasing speed in the same lane will be one way to avoid the necessity to make room for vehicles changing into its lane directly ahead of it. Reasoning this way a rational agent could hide its true valuation for the action, 0.7, and report it as, for instance, 0.1. All else remaining equal, the maximum attainable social welfare of the system drops to 2.5 in which both vehicles 1 and 2 maintain their speed and lane, and vehicle 3 is forced to implement its third choice, $((v_3^t)_{23} = 0.5)$, namely to continue travelling in its current lane but accelerate by 5 miles per hour. Vehicle 3 incurs no cost for this action because the action allocation for the remaining two vehicles does not change when agent 3 is not present in the system. Vehicle 1 travels conflict-free during the time slice, so its cost is also zero. However, vehicle 2 pays a price for misrepresenting its valuation and its lack of flexibility. The cost incurred by it is $p_2^t = 2 - 1.5 = 0.5$ and its utility drops to 0.5 (the true valuation of 1 less the price to take the approved actions), so, lying not only does not improve outcomes, it may leave the vehicle worse off.

An attempt to achieve certain desired actions by lying about the availability of alternatives, e.g., agent 2 claiming that decelerating in its lane is unfeasible by assigning it valuation 0, can be easily detected and assigned a steep monetary penalty as the manager has situational awareness and knows what actions are available to all of its players.

Finally, a player might consider lying to try to decrease the cost it may potentially pay for the right to implement certain actions. In the example, consider agent 3 misrepresenting its valuation for move up/accelerate as 0.9 instead of 1. The decrease is small enough to still convey the strong desirability of the actions but could potentially decrease the price. However, the cost incurred by a player corresponds to the externality it causes on the system and is calculated disregarding the agent's own report of importance of actions. Under these conditions, the maximum achievable social welfare is 2.6 (1 from v_1, 0.7 from v_2, and 0.9 from v_3). Both agents 1 and 2 have zero payments, but agent 3 pays 0.3 as before. Undervaluing actions to decrease payment is thus futile and agents are again best served by telling the truth. This is not surprising, since, as shown in Proposition II.2 the mechanism is strategy-proof and incentivizes truth telling.

The illustrative example presented here was implemented and verified in a NetLogo simulation as seen in Figure 6.3.

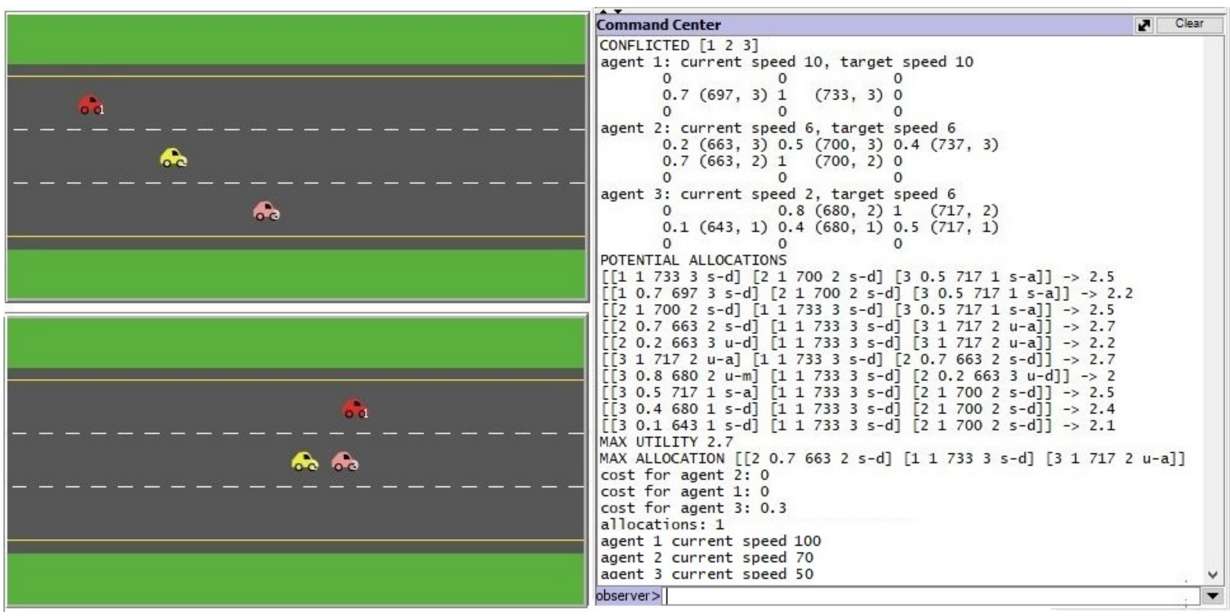

Figure 6.3: Mechanism Design Among Driverless Vehicles on a Highway Example - NetLogo Simulation Results

Note that in the presented example scenario even though social utility is maximized, v_3 will accelerate and pull in in front of v_2 which will decelerate per mechanism approved action. The actions are allowed by the mechanism because the constraint specifying the minimum safety distance between agents on the road is satisfied. However, this will still leave a 20 mph difference between the agents' velocities and vehicle 2 will ultimately be forced to decelerate more during the adjustment period between rounds to avoid a collision. A potential refinement of the mechanism will not only detect an agent inflicting an externality on the system but also the degree of its severity. The mechanism will either charge more for inconveniencing other players to a larger extent or adjust buffer zone requirements based on speeds.

6.3 EMPIRICAL MODEL

A test bed was created to implement and verify the mechanism.

6.3.1 Environment and Agent Characteristics

As outlined before, the simulated expressway has lanes with minimum and maximum speed characteristics. The test bed adds another attribute, emission level. All three attributes were assigned simplified valid ranges with values between 1 and 10. For an example five-lane highway, as shown in Table 6.2, the top lane, Lane 5, is characterized by the maximum overall speed but also the maximum emission level. At the bottom, Lane 1 is the slowest but most emission-friendly lane.

Table 6.2: Empirical Model - Lane Characteristics of Simulated Five-Lane Highway

	Emission Levels (1-10)	Maximum Speed (1-10)	Minimum Speed (1-10)
Lane 5	10	10	9
Lane 4	8	8	7
Lane 3	6	6	5
Lane 2	4	4	3
Lane 1	2	2	1

The vehicle agents' current and preferred speed attributes are valued between 1 and 10, with 10 being the highest. The objective emphasis attribute takes values between 1 and the number of lanes being simulated, with the higher values commensurate with the importance the vehicle assigns to its governing objective.

As before, an assumption is made regarding prevailing laws governing the travel of vehicles from one lane to another. Without restriction of generality, it is assumed that once an agent starts to travel to a different lane the movement will be largely perpendicular to the vehicle's current position. Trajectory generation and tracking are not a subject of this algorithmic model.

6.3.2 Agent Objectives, Atomic Action Derivation, and Action Ranking

This case study defines three potential objectives: agents may choose to prioritize minimizing personal travel time (TT), reducing lane congestion level (CL), or reducing global emissions (EL). As before, by maintaining an objective vehicles on the road are working against some or all of the remaining objectives.

The system maintains 9 atomic actions. Only single lane changes are allowed and the acceleration/deceleration factor is 0.5 (on a speed scale from 1 to 10).

The mechanism for derivations of feasible actions is shown in Algorithm 5. It partially contains Algorithm 1 from Chapter 4 to help determine the physical achievability of lane and speed changes (lines 5.1-5.9)).

The ranking of feasible lane/speed actions is decided by rule sets for each objective taking into account the target lane and speed, the current speed, the current traffic lane and its speed restrictions, and the physical availability of space as determined by the neighboring agents. Travel Time (TT) and Emission Level (EL) vehicles strive to reach and maintain their target lane and speed as determined by the relationship between their preferred speed and objective emphasis. Congestion Level (CL) agents designate their preferred speed as target speed. They act to reach their targets and then consider and compare the relative congestion levels of the neighboring lanes. The least obstructed lane is selected as the new target lane and the level of objective emphasis is used to determine the new target speed (lines 5.44-5.55). All players will rank action combinations depending on where their current speed setting and lane position are in relation to their targets. Ranking particulars are omitted for brevity.

These are the abstract concepts relevant to the test bed.

Algorithm 5: Feasible Atomic Action Derivation

Variables: For $TargetLane$ $TargetSpeed$ falls within lane speed range.
If agent objective is Personal Travel Time,
$TargetSpeed = PreferredSpeed + ObjectiveEmphasis.$
If objective is Global Emission Level,
$TargetSpeed = PreferredSpeed - ObjectiveEmphasis.$
If objective is Congestion Level, $TargetSpeed = PreferredSpeed.$

```
if LaneUp in bounds and LaneUp not occupied then
    CanMoveUp = true
CanMaintainSpeed = true
if LaneDown in bounds and LaneDown not occupied then
    CanMoveDown = true
if SpaceBehind not occupied or (SpaceBehind occupied and CurrentSpeed of
  CarBehind < CurrentSpeed of myself) then
    CanDecelerate = true
if SpaceAhead not occupied or (SpaceAhead occupied and CurrentSpeed of
  CarAhead > CurrentSpeed of myself) then
    CanAccelerate = true
if TargeLane above CurrentLane and CanMoveUp then
    CanMoveUpAndMaintainSpeed = true;
    if CurrentSpeed < TargetSpeed then
        CanMoveUpAndAccelerate = true;
    else if CurrentSpeed > TargetSpeed then
        CanMoveUpAndDecelerate = true;
    else
        if CurrentSpeed < HighwayMaximumSpeed then
            CanMoveUpAndAccelerate = true;
        if CurrentSpeed > LaneUpMinimumSpeed then
            CanMoveUpAndDecelerate = true;
if TargeLane below CurrentLane and CanMoveDown then
    CanMoveDownAndMaintainSpeed = true;
    if CurrentSpeed < TargetSpeed then
        CanMoveDownAndAccelerate = true;
    else if CurrentSpeed > TargetSpeed then
        CanMoveDownAndDecelerate = true;
    else
        if CurrentSpeed > HighwayMinimumSpeed then
            CanMoveDownAndDecelerate = true;
        if CurrentSpeed < LaneDownMaximumSpeed then
            CanMoveUpAndAccelerate = true;
```

▷ Continues on next page..

Algorithm 5: Feasible Atomic Action Derivation

▷ Continued from previous page..

```
CanStayAndMaintainSpeed = true;
if (TargetLane = CurrentLane and AgentObjective != CongestionLevel) or
  TargetLane != CurrentLane then
    if CurrentSpeed < LaneMaximumSpeed and CanAccelerate then
        CanStayAndAccelerate = true;
    if CurrentSpeed > LaneMinimumSpeed and CanDecelerate then
        CanStayAndDecelerate = true;
if TargetLane = CurrentLane and AgentObjective = CongestionLevel then
    if CurrentSpeed < TargetSpeed and CanAccelerate then
        CanStayAndAccelerate = true;
    else if CurrentSpeed > TargetSpeed and CanDecelerate then
        CanStayAndDecelerate = true;
    else
        Calculate RelativeLaneCongestion within distance ±x;
        /* value of x determined by mechanism                         */
        if CanMoveUp then
            Calculate RelativeLaneUpCongestion within distance ±x;
        if CanMoveDown then
            Calculate RelativeLaneDownCongestion within distance ±x;
        MinCongestion = min( valid RelativeLaneCongestion,
          RelativeLaneUpCongestion, RelativeLaneDownCongestion);
        /* break ties in favor of the higher lane                     */
        TargetLane = the lane with MinCongestion;
        TargetSpeed = TargetSpeed−ObjectiveEmphasis∗(1−MinCongestion);
        if TargetSpeed < TargetLaneMinimumSpeed then
            TargetSpeed = TargetLaneMinimumSpeed;
        if TargetSpeed > TargetLaneMaximumSpeed then
            TargetSpeed = TargetLaneMaximumSpeed;
        if TargetLane = LaneUp then
            CanMoveUpAndAccelerate = true;
            CanMoveUpAndMaintainSpeed = true;
        else if TargetLane = LaneDown then
            CanMoveDownAndMaintainSpeed = true;
            CanMoveDownAndDecelerate = true;
        if CanAccelerate and CurrentSpeed != LaneMaximumSpeed then
            CanStayAndAccelerate = true;
        if CanDecelerate and CurrentSpeed != LaneMinimumSpeed then
            CanStayAndDecelerate = true;
```

6.4 SIMULATION

The empirical model was implemented and verified with a NetLogo simulation.

6.4.1 Settings

The simulation does not implement mechanism management assignment and considers the limited stretch of road as a single multi-agent group. When deployed on prohibitively long stretches of road, the implementation will require some simple additional code to partition traffic and assign vehicles to local subgroups based on their spatial position.

Again, for simplicity, agents have the same physical characteristics. Adding size and performance attributes is a minor extension. As before, to distinguish between vehicles' objectives, travel time (TT) prioritizing agents red, emission level (EL) agents are green, and congestion level (CL) agents are colored in yellow. The test bed allows for the number of different types of agents on the road, their maximum level of patience, and the number of lanes to be adjusted. In the current implementation the maximum number of lanes is set to 5, but that upper bound was chosen to mimic real world conditions and is not absolute. Small adjustments to scale up the legal value ranges of speed variables (1-10) and objective emphasis (1-5) will transition the application of the mechanism to an arbitrary (but realistic) number of lanes.

Driverless vehicles will have faster reaction times than human drivers so the required safety distance in the test bed is set at 1.5 car lengths. However, the mechanism can easily be assigned a different safety gap or be extended to dynamically calculate the required space depending on speed.

Before the simulation is run, all agents are spawned at random lane positions adhering to the buffer zone requirements with random current speeds within the legal range. They get a random assignment of preferred speed and objective emphasis. Depending on the type of objective an AV follows, it uses (a combination of) these indicators to determine lane and speed targets.

Each round (cycle) of the game lasts for 10 time slices (ticks). Agents derive and rank their feasible actions within a single tick. Conflict resolution, allocation construction, appointment of approved actions and their implementation, as well as payment collection all happen on the very next tick although these two time slices can actually be combined without issue. The remaining 8 time slices are used for further forward progression after adjustments and mechanism housekeeping. The total length of the round can be adjusted by adding additional time slices for forward progression or making decisions on their duration. Generally, the ideal duration of a full round is best decided in real world conditions. The aim is to space out time slices for action implementation in a way that does not hamper the ability of agents to undertake maneuvers necessary for progress to their destination but to not make the cycles too short as to not present a computational burden and not impede traffic by too frequent lane changes.

6.4.2 Happiness and Patience

The simulation includes an indicator for *happiness*. "Happy" agents have reached their target lane and are travelling in it at their target speed. What happens afterwards however, differs according to the governing objective. As before, happy TT and EL vehicles strive to maintain these perceived ideal settings, only occasionally making overtake maneuvers where necessary, or making mechanism sanctioned changes if there is conflict present and they did not win their highest valued actions. Also as before, for CL agents, a happy state indicates that it is time to look to the neighboring lanes and see if it is possible to make a switch to a lane less congested than their current one. Making a switch to a neighboring lane still affords the agent the ability to travel at a pace close to their original preferred speed. When finding a lane with a relatively lower congestion, a CL agent updates its lane and speed targets before moving to it. Once the congestion level prioritizing AV reaches a happy state again, the search for a less congested neighboring lane renews. In order for CL agents to not inadvertently deviate too far from their original preferred speed due to traffic density circumstances, in the test bed their target speed and target lane are reset back to the initial values every 10 cycles.

The simulation was written in a way that when assigning valuations to feasible actions, agents are led by the need to reach a happy state as quickly as possible. Because of road availability and mandatory safety distances lane changes in heavy traffic are somewhat less likely to be approved than simple speed changes in the same lane so, when ranking preferences, speed constraints local to the lane may be partially ignored in favor of moving towards the target lane faster. For example, if there is an opening in the lane above, a vehicle that finds itself in a lane somewhere below its target, will always choose its top rated feasible action combination as *move up* and the appropriate speed action given its current speed. A vehicle that should be moving down towards its target lane makes similar ranking decisions. Note that speed decisions are preempted by a collision avoidance mechanism which forces an agent to either decelerate or match the speed of a vehicle directly ahead should it find itself too close or driving fast enough to subsequently violate the predefined safety distance. Thus, agents strive to reach their target lane first and foremost, and make the accompanying speed decisions along the way.

The ability of agents to quickly reach a happy state and the capacity of TT and EL agents that have satisfied their objective targets to maintain said targets depends on overall traffic. Sometimes an agent has reached its goal lane and speed and maintains it until it is forced to decelerate when it catches up with a vehicle driving slower in the same lane. In order to avoid situations in which agents are stuck in this inferior condition for a long time, this simulation also includes the additional feature of *patience* for all agents. As before, an agent is given a random level of patience, which for the test bed is an integer between 10 and a user defined level and can be adjusted to reflect the sensitivity of agents to less than ideal conditions. For each time slice during which the vehicle is blocked by a slower driving vehicle directly ahead, it loses a unit of patience. In the time slice during which agents determine their feasible action valuations, if the patience level has dipped below zero the agent makes additional valuation decisions based on Algorithm 6.

Algorithm 6: Feasible Action Override for All Agents With $Patience \leq 0$

if $CanMoveUp$ **then**
    $ValuationOfMoveUpAndAccelerate = 1$;
    $XIfMoveUpAndAccelerate = x + (CurrentSpeed + 2)$;
    $YIfMoveUpAndAccelerate = y + 1$;
else if $CanMoveDown$ **then**
    $ValuationOfMoveDownAndAccelerate = 1$;
    $XIfMoveDownAndAccelerate = x + (CurrentSpeed + 3)$;
    $YIfMoveDownAndAccelerate = y - 1$;
reset $Patience$;

As shown, a player that has run out of patience will attempt an overtake maneuver. The test bed implements right-hand traffic and allows for passing of vehicles both on the left (first choice) and on the right (only if passing on the left is not available at the moment) to reflect traffic rules in the United States. The pass on the right option (overtaking on the inside) is easily turned off if local traffic laws prohibit it. In the simulation, the nine atomic actions presented in 6.1.5 can be assigned a unique value between 0.1 and 0.9. When patience has run out, should a move up or down be physically possible, the *move up and accelerate* or *move down and accelerate* action is assigned a valuation of 1, signalling its precedence over all other possibilities. These actions are also coupled with a higher than usual acceleration factor to achieve the goal of passing the blocking vehicle(s) quickly while disrupting traffic as little as possible. For simplicity, the simulation does not currently evaluate the probability of success of an overtake maneuver.

6.4.3 Bid Reduction to Conflict Subset

To alleviate the computational load on the allocation function by reducing the conflict set or in some (most) cases to entirely eliminate the need for the allocation function to fire, the model employs a bid reduction strategy. This strategy starts with the entire bid set and incrementally reduces it to the most minimal conflict set possible.

First, should an agent only have one available lane/speed action combination (stay and maintain speed), the mechanism automatically approves it and removes all conflicting bids for that projected position from contention. The algorithm then looks at the top valued choices for each agent. If the bid representing the top action for an agent is unconflicted, the owner of the bid is given approval to implement it and all its lesser valued bids are deleted. For example, for the random traffic configuration in Figure 6.4, the initial list of bids in the form [*agent-id valuation potential-x-coordinate potential-y-coordinate lane-speed-action*] submitted by the agents ordered along the x-coordinates looks as follows:

Lane 5: [7 0.8 7.8 4 u-m] [7 0.9 7.82 4 u-a]

Lane 4: [7 0.5 7.8 2 s-m] [7 0.6 7.82 2 s-a] [9 0.8 11.8 2 s-m] [9 0.9 11.82 2 s-a]

Lane 3: [10 0.4 9.38 0 s-d] [10 0.5 9.4 0 s-m] [8 0.5 11.5 0 s-m] [8 0.6 11.53 0 s-a]
[6 0.7 13.57 0 u-d] [6 0.9 13.6 0 u-m] [6 0.8 13.62 0 u-a]

Lane 2: [10 0.9 9.38 -2 d-d] [10 0.8 9.4 -2 d-m] [11 0.9 10.57 -2 u-d]
[11 0.8 10.6 -2 u-m] [6 0.5 13.57 -2 s-d] [6 0.6 13.6 -2 s-m]

Lane 1: [11 0.6 10.57 -4 s-d] [11 0.5 10.6 -4 s-m]

The algorithm reduces the bids as outlined above only leaving entries to reflect any present conflict. There are no agents with only one feasible action. Agent 7's top bid in lane 5 represents the *u-a* action (move up and accelerate). The potential position resulting from that action is spatially not in conflict with any other agents' intended position, so the vehicle receives approval to implement the action and agent 7's remaining bids (valued 0.8, 0.6, and 0.5) are removed. Similarly, agents 9 in lane 4 receives approval for its highest valued action *s-a* (stay and accelerate). Agents 6 and 8 are bidding for positions in the same lane but their highest valued actions will result in positions that satisfy the safety distance requirement of 1.5: vehicle 8's top valued (0.6) potential position's closest rival (agent 10 bid 0.5) is 2.13 units behind and ahead the distance to vehicle 6's closest bid (0.7) is 2.04. Similarly, agent 6's top bid (0.9) is unconflicted and the player is granted the *u-m* (move up and maintain speed) action, while player 8 is

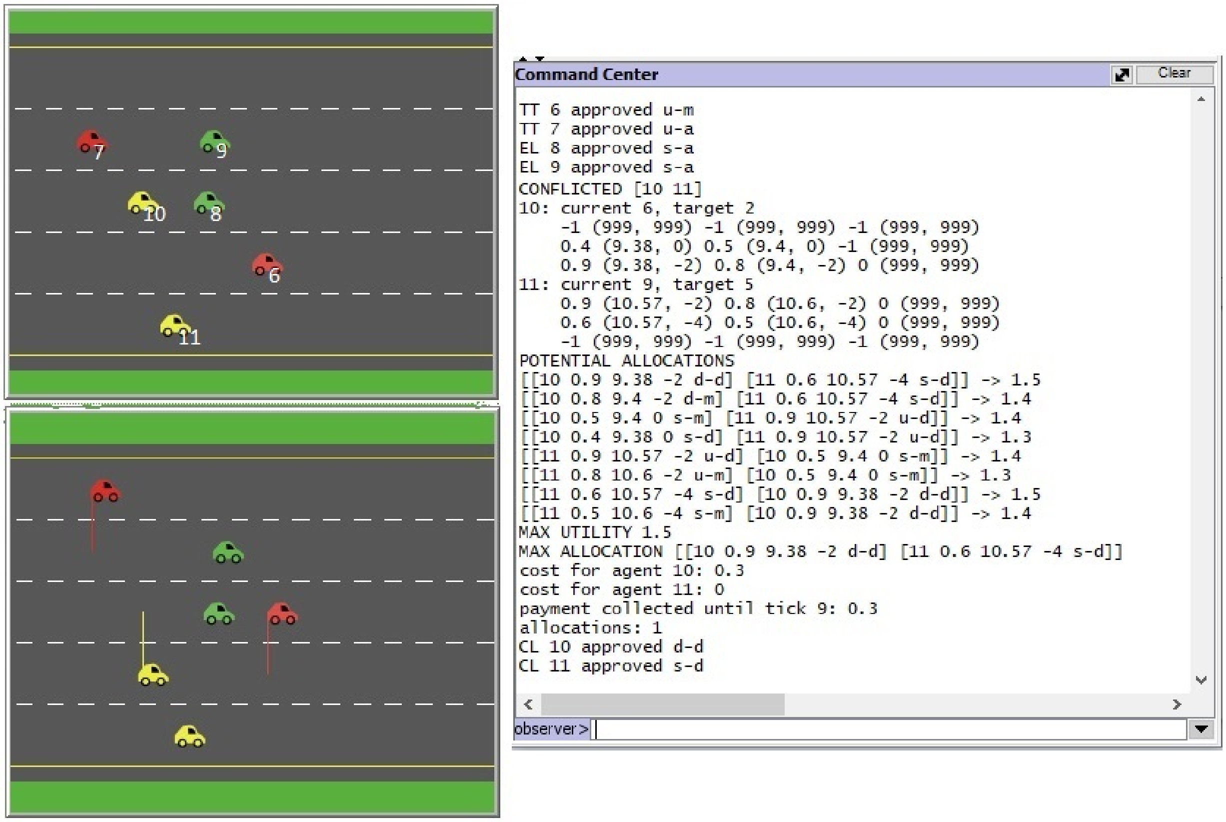

Figure 6.4: NetLogo Simulation of a 5 Lane Road

allowed to implement *s-a* and their remaining bids are taken out of contention. CL agent 10 is currently travelling in lane 3 with a target lane below, while CL agent 11 finds itself in lane 1 looking to move up. The top bids of agent 10 and 11 are found in lane 2, where, should the vehicles be allowed to implement their corresponding top actions, they would find themselves 1.19 distance away from each other. As the best choices are conflicted, the bid list is reduced only to the bids of the conflicted agents, becoming:

Lane 5: no bids

Lane 4: no bids

Lane 3: [10 0.4 9.38 0 s-d] [10 0.5 9.4 0 s-m]

Lane 2: [10 0.9 9.38 -2 d-d] [10 0.8 9.4 -2 d-m] [11 0.9 10.57 -2 u-d] [11 0.8 10.6 -2 u-m]

Lane 1: [11 0.6 10.57 -4 s-d] [11 0.5 10.6 -4 s-m]

From the reduced conflict set, the mechanism then constructs the best possible allocation in terms of collective utility as shown in Figure 6.4. Agent 10 is allowed its top choice (0.9) to move down and decelerate and agent 11 receives approval for its third best option valued at 0.6.

During a cycle, conflict may be entirely absent or be present among any number of agents. When the set of incompatible bids is not empty the system also calculates cost. If faced with conflict, to improve efficiency, the simulation only produces the top allocation candidates (as seen in the output in Figures 6.3 and 6.4). The underlying algorithm considers bids in the order from highest valuation to lowest. It attempts to extract the best allocation (if it exists) for each bid of every player. Once it finds an allocation for a bid, it disregards lesser potential configurations and moves on to the next lesser valued bid for the same agent. That is the reason why for the example scenario presented in section 6.2 from all feasible allocations shown in Table 6.1, the model is satisfied by only finding allocations 1, 2, 3, 4, 8, 9, and 13. By sorting bids and truncating the search space the model guarantees to find the social utility maximizing allocation in shorter time. The algorithm can be further optimized by the early detection of duplicates and by abandoning allocations when part way through constructing them it becomes mathematically impossible to deliver a better result than the current leading configuration. Further optimization is currently not the aim of this work.

After the allocation is constructed, what is left, is for the system to calculate cost, communicate approved actions to agents, collect payments owed and for vehicles to implement these actions.

6.4.4 Test Runs

The simulation was used for test runs of 1,000,000 time slices (100,000 full rounds) at different levels of traffic density. The number of vehicles was evenly split among agents adhering to the 3 different objectives. The legal range of valuations (and thus cost) of lane/speed actions was [0,1]. The characteristics and number of constructed allocations and the amount of payment collected for five of the test run are summarized in Table 6.3.

Table 6.3: Simulation 100,000 Full Cycle Test Runs, Conflict-Only Allocation Results

Traffic Density	Conflict Set Allocation Size				Total Allocations	Payment Collected
	2 agents	3 agents	4 agents	5+ agents		
15%	1105	10	0	0	1115	262.0
25%	1467	21	4	0	1492	342.0
50%	7392	264	277	19	7952	1842.5
75%	8629	452	450	32	9563	2341.6
85%	9175	522	523	66	10286	2347.50

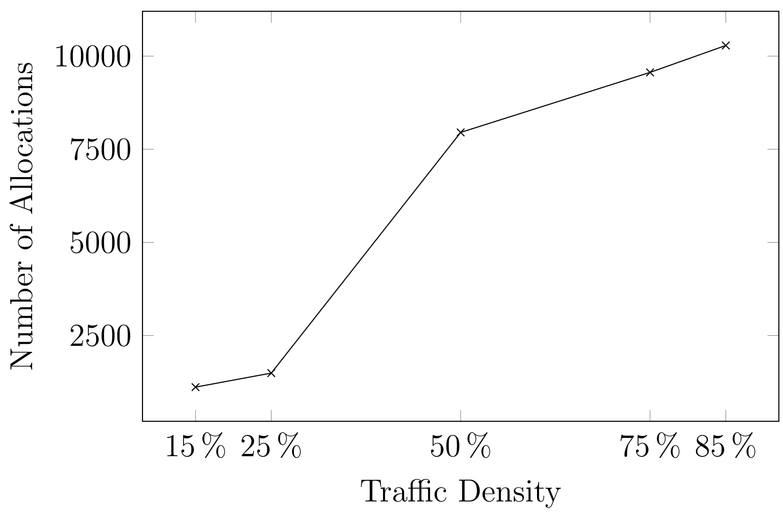

Figure 6.5: Simulation 100,000 Full Cycle Test Runs, Total Allocations per Traffic Density

Table 6.4 lists some of the simulation results for the happiness and speed attributes. Due to the different nature of a happy state for CL agents, they were partially separated out.

For the 100,000 cycle duration of the simulation, at 25% traffic density, on average 88.2% of TT and EL agents had simultaneously reached their ideal (happy) state. At that traffic density level, the happiness indicator for TT and EL agents reached as high as 100% and never fell below 50%. At 75% traffic density, at most 92.5% of TT and EL agents were happy at the same time. It is plain to see that as traffic volume increases, the ability of vehicles to maintain goal positions and velocities diminishes. At 25% traffic density all TT and EL agents were simultaneously happy 17.6% of the time as opposed to

Table 6.4: Simulation 100,000 Full Cycle Test Runs, Speed and Happiness Results

Performance Indicator	Traffic Density 25%	50%	75%
Average happiness of TT and EL agents	88.2%	70.6%	66.0%
Minimum happiness of TT and EL agents	50%	11.1%	5%
Maximum happiness of TT and EL agents Time slices at 100%	100% 175,678	100% 38	92.5% -
Average happiness of all agents	72.7%	56.9%	54.1%
Average difference of current to target speed for TT agents (1-10 scale) Max over/under difference of avg current speed to avg target speed for TT agents	-0.09 1.25/2.63	-0.49 0.61/3.57	-0.3 0.65/3.8
Average difference of current to target speed for EL agents (1-10 scale) Max over/under difference of avg current speed to avg target speed for EL agents	+0.02 2/1.33	+0.02 1.84/0.77	+0.1 3.7/0.5
Average difference of current to target speed for CL agents (1-10 scale) Max over/under difference of avg current speed to avg target speed for CL agents	+0.52 3.5/1.92	+0.5 2.65/1.42	+0.43 1.85/1.65

a mere 0.0038% when half the road was occupied by vehicles. Additionally, as outlined in Table 6.3, as traffic volume doubled from 25% to 50%, the level of conflict increased 5 times as reflected by the total number of conflict-resolving allocations triggered by agents' wishes. At higher traffic density rates however, as seen in Figure 6.5, as open lane positions become less available, the allocation growth rate becomes marginal. Even at 85% traffic density, conflict that could not be resolved by the preliminary bid reduction strategy was present in just 10,286 rounds out of 100,000 (10.3%) and in the vast majority of these cases the simulation successfully reduced the conflict set down to just two vehicles, which is trivial for the allocation function to resolve.

The diversity of speeds falling within the allowed range for a single lane results in some vehicles inadvertently blocking others behind them and slowing them down. Before they adjust their travel pace, AVs entering faster lanes at speeds temporarily at around or under the lane minimum speed can slow other lane occupants down as well. This resulted in Travel Time objective agents moving on average at a rate slightly under their goal. However, the average difference between current and target speeds for TT agents was still less than half a unit (on a 1-10 scale), so the mechanism did a good job of creating allocations benefiting the collective good that did not overly impede traffic. Emission Level agents travelled on average slightly above their target speed due to speeding up for overtake maneuvers they performed when blocked by slower vehicles for a period of time long enough to run their patience level down. By design, CL agents varied their targets but the test run results showed that on average they too were able to travel very close to their goals. Altogether, EL agents did best at maintaining their target speed followed by TT agents. As expected, the changing nature of CL vehicle goals resulted in the largest deviation from target speeds.

Even with restricted movement at 75% traffic density, the average happiness of agents was still high. Recall that a state of happiness was only indicated for vehicles currently travelling according to their *exact* wishes so agents who were only marginally removed from their happy state were not recorded. The mechanism as designed, achieved satisfactory outcomes for the whole group.

Tables 6.5 and 6.6 list happiness levels, target speed differences, and allocation characteristics for mixed traffic (all three objectives) and homogeneous traffic (TT or EL agents only). Even though speed and lane targets still vary from agent to agent, vehicles with the same objectives approach traffic in a similar way, so overall conflict level was lower. However, because of the tendency of TT agents to cluster in the higher lanes and at higher speeds it is more difficult to find gaps to perform overtake maneuvers, in homogeneous traffic these agents impeded each other's target speeds to a higher degree, leading to a lower average level of happiness. Conversely, EL agents in homogeneous traffic tend to keep to the lower lanes. At lower speeds however, opportunities for overtake maneuvers are easier to come by, which resulted in these agents being happier and travelling on average above their target speeds to a higher extent than in mixed traffic.

Table 6.5: Simulation 100,000 Full Cycle Test Runs, Results for Happiness and Speed for EL and TT Agents in Mixed and Homogeneous Traffic

	Traffic Density 50%			
	TT in mixed traffic	TT only traffic	EL in mixed traffic	EL only traffic
Minimum happiness	8.8%	0%	13.4%	0%
Average happiness	69.1%	67.2%	72.1%	79.0%
Maximum happiness	100%	87.5%	100%	92.5%
Average difference of current to target speed	-0.49	-0.61	+0.02	+0.17

Table 6.6: Simulation 100,000 Full Cycle Test Runs, Allocation Results in Mixed and Homogeneous Traffic

Traffic Density and Type	Conflict Set Allocation Size				Total Allocations	Payment Collected
	2 agents	3 agents	4 agents	5+ agents		
50% mixed traffic	7392	264	277	19	7952	1842.5
50% TT agents only	1645	52	12	2	1711	364.9
50% EL agents only	645	9	2	0	656	133.6

Figures 6.7, 6.8, and 6.9 show the test runs at the completion of the millionth time slice for traffic density rates of 25%, 50%, and 75% respectively. Vertical lines record just implemented lane changes, smiley face avatars denote vehicles in a happy state.

6.4.5 Time Complexity, Optimization, and Scalability

In the simulation's current version, bid creation takes place simultaneously for all n agents in runtime $O(1)$ for each agent. Each of the n agents has at most m bids, where m is a constant. For the 2-dimensional lane/speed case discussed here, $m = 9$.

Agents insert their bids into the proper spatial order in a list for each of the l lanes, where l is constant. This happens in $O(n)$ time with a conventional insertion strategy.

The current implementation of the mechanism's conflict evaluation and resolution functionality contains two steps:

1. *Bid reduction* - the bid set is reduced by approving only-option and uncontested top-option bids and removing any subpar bids for those agents

2. *Conflict-free allocation extraction* - the remaining conflicted bids are arranged in the conflict-free configuration delivering the highest combined utility

Lemma II.2. *The mechanism runs in $O(n^3)$ time (worst case).*

Proof. During step 1, bid reduction, the mechanism iterates through each of the l lanes' bid lists, looking for two kinds of bids. For only-option bids, the algorithm looks to the bid's left and right and deletes any other claims within the required safety distance. For unconflicted best-option bids, the algorithm needs to consult at most 3 bids ahead and 3 bids behind to find out if there's conflict with others, including the same agent's own bids. Subpar choices for all agents with approved top-option actions are consecutively removed with a conventional search strategy.

These operations are implemented with nested loops. In the worst case, there are $\sum_{i=1}^{l}(\underbrace{\sum_{j=1}^{x}(\sum_{k=j-1}^{1}+\sum_{j+1}^{x})}_{\text{only-option bid reduction}}+\underbrace{\sum_{j=1}^{x}}_{\text{top-option bid reduction}}+\underbrace{\sum_{j=1}^{x}}_{\text{remove subpar bids}})$ iterations, where x is the number of bids. The combined runtime is thus $O(n^2)$, since $l = const$ and at most $x = n * 3$.

Step 2, conflict-free allocation extraction, is only executed when after step 1 terminates not all agents have received approved actions. For the size of the conflict set c, the algorithm finds a conflict-free allocation in at most $\sum_{i=1}^{c}\sum_{j=1}^{9}\sum_{k=1}^{c}\sum_{o=1}^{9}\sum_{p=1}^{c-j-1}$ iterations, bringing the worst case runtime of the whole mechanism to $O(n^3)$. $\square$

While $O(n^3)$ runtime sounds like bad news, numerical analysis from test runs of the mechanism (see section 6.4.4) shows that the constraints of the environment and preliminary bid reduction strategies have an overwhelmingly positive effect on the computational burden placed on the allocation function. Enforcing lane speeds and an appropriate buffer zone between vehicles minimizes the number of initial bids deemed feasible and thus limits the size of the bid set. Bid ordering ensures that the only-option and top-option bid reduction operations usually terminate early. The test runs show what the bid reduction strategy delivers a conflict free set at least 90% of the time, so the computationally heavy allocation function does not have to be executed at all. And because of local partitioning, the size of the conflict set has a low upper bound and is minimal or trivial in the vast majority of cases ($n = 2$) as seen in Figure 6.6.

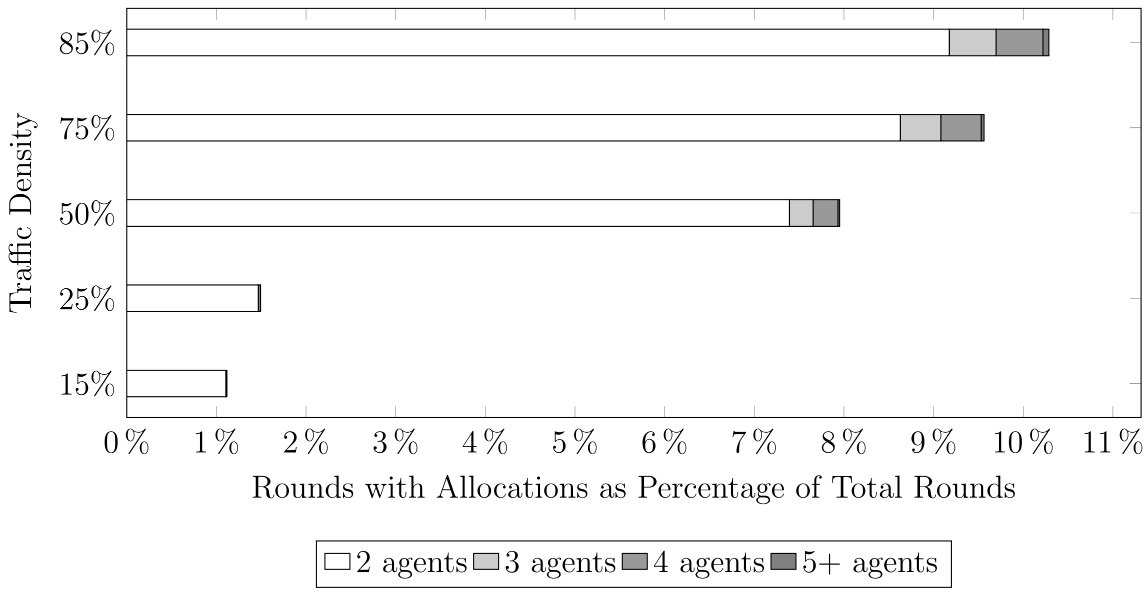

Figure 6.6: Simulation 100,000 Full Cycle Test Runs, Size of Allocations per Traffic Density

There are several potential avenues for runtime reduction and optimization. Lane changes only affect agents close by and conflict only arises among immediate neighbors so partitioning mechanisms can be employed in several ways. When the mechanism governs a long stretch of road, the conflict set can be split into longitudinally disjoint local clusters, further reducing its size. Also, the conflict set can be vertically partitioned into smaller sets when conflicted positions are in disjoint sets of lanes involving separate groups of agents. The local conflict subsets can be processed simultaneously to save time. As a matter of fact, after receiving the agents' bids, the manager can run a partitioning algorithm to subdivide its area of influence into smaller sets without overlapping bids to process separately.

The allocation function can be implemented to recognize duplicate allocations and ignore them, as well as to abandon suboptimal allocations early. Depending on traffic density, the mechanism manager can set an upper bound on the number of agents making up the conflict set. Should there be too much conflict, the manager can skip the allocation altogether, instructing each agent to stay in the same lane and maintain the current speed. Several seconds of forward progression will inevitably result in a different configuration at the beginning of the next round. This work focuses on the mechanism as a concept and does not further investigate its optimization.

As outlined, many factors contribute to conflict reduction on a scale sufficient enough to only necessitate firing the computationally expensive allocation function in a small number of rounds. Vehicles regulate their own speed so if an agent has to slow down because of a lane change made some distance ahead, it can do that without having to bid for it. Speed and safety distance constraints further cut down the set of feasible actions. In lighter traffic, there are fewer total bids and fewer conflicts. Denser traffic results in fewer feasible action options. The preliminary bid reduction strategy cuts down the size of the conflict set, in most cases eliminating it entirely. Should the allocation function have to be executed, it is only run on the reduced conflict set(s). The option to skip allocation when too much conflict is present and the substantially diminished size of the local conflict subset greatly reduce the calculation burden and alleviate scalability concerns connected to the Clarke pivot rule.

6.4.6 Additional Features and Future Extensions

The simulation offers features like focusing on a single vehicle, showing identifying characteristics of agents, briefly showing lane change traces. It also shows informational plots for allocation sizes, average target speed vs. actual speed for each type of agent, driver patience, and happiness levels.

Additional features specific to the application domain can be added to the

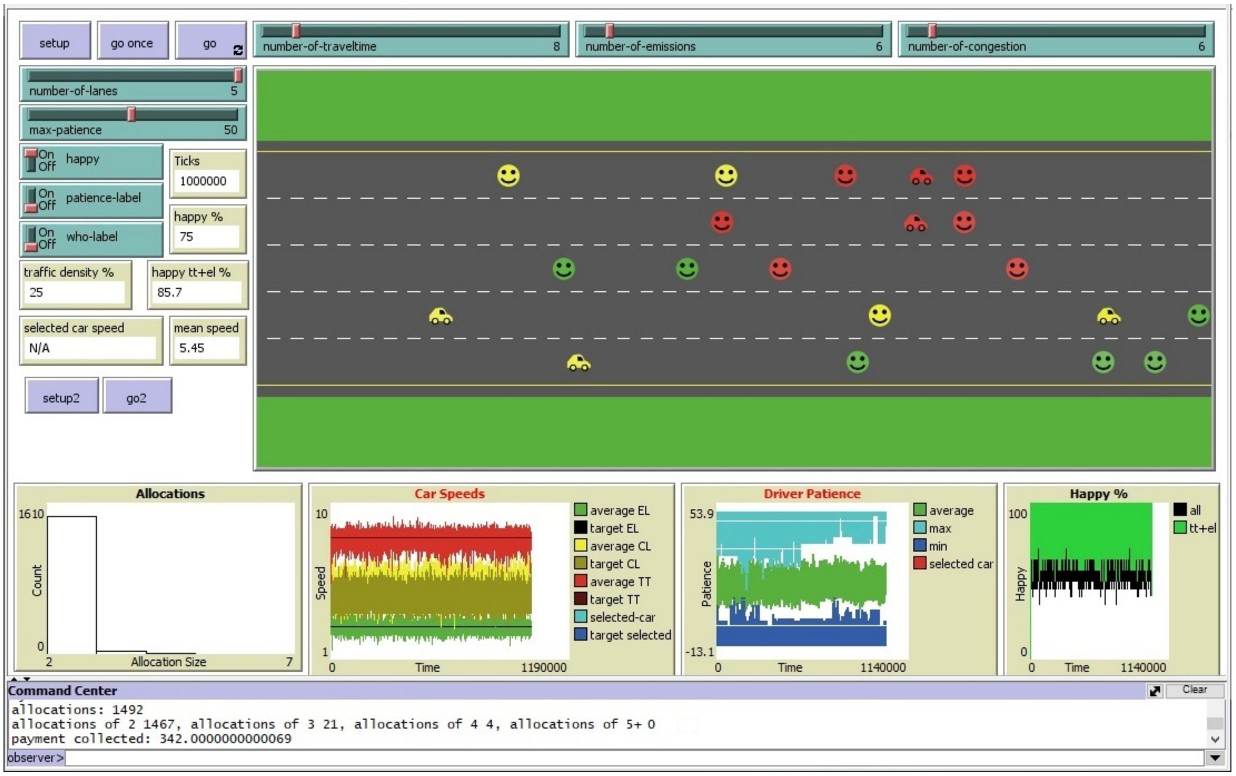

Figure 6.7: NetLogo Simulation Completed 1,000,000 Time Slice Test Run for Traffic Density of 25%

simulation while still keeping the integrity of the underlying model. For added behavioral flexibility, agents could be allowed to switch between objectives at will. Auxiliary objectives can be modelled and included to represent AV intentions - e.g maximizing fuel efficiency, following a particular lead vehicle, etc. Multiple ranked active objectives could fire their rule sets in turn to arrive at the final valuation. For progress improvement, agents could sense the feasibility of overtake maneuvers before undertaking them. The current happiness metric can be extended by degrees of satisfaction. The patience metric and the circumstances that cause it to decrease could be refined. The payment scale can be changed and monetary or other incentives can be added to the resolution of utility ties. The allocation algorithm can be augmented with degrees of externality to reflect the severity of imposition an approved action for an agent would inflict on others. The resulting scale can be used to make more sophisticated allocation decisions or can result

Figure 6.8: NetLogo Simulation Completed 1,000,000 Time Slice Test Run for Traffic Density of 50%

in the utilization of a multiplier for mechanism payment according to the circumstances. Of course, a weight multiplier can be introduced or the valid value range of the valuation matrix can be scaled to reflect the desired monetization rate as determined by the mechanism designer.

Additionally, the simulation should be equipped with attributes allowing for the diversity of agents in terms of physical features. Those features in combination with current speeds should be taken into account when calculating an acceptable buffer zone during travel. Agents could be allowed to maintain shorter distances when driving in slower lanes or in collaborative situations like e.g., platooning. In a platoon, agents could be allowed to bid together. Next, inclusions can be made to add obstacles that can be treated as unavailable road positions. The model can be extended to account for extraneous vehicles on the road. Such vehicles can be malicious, for which a security

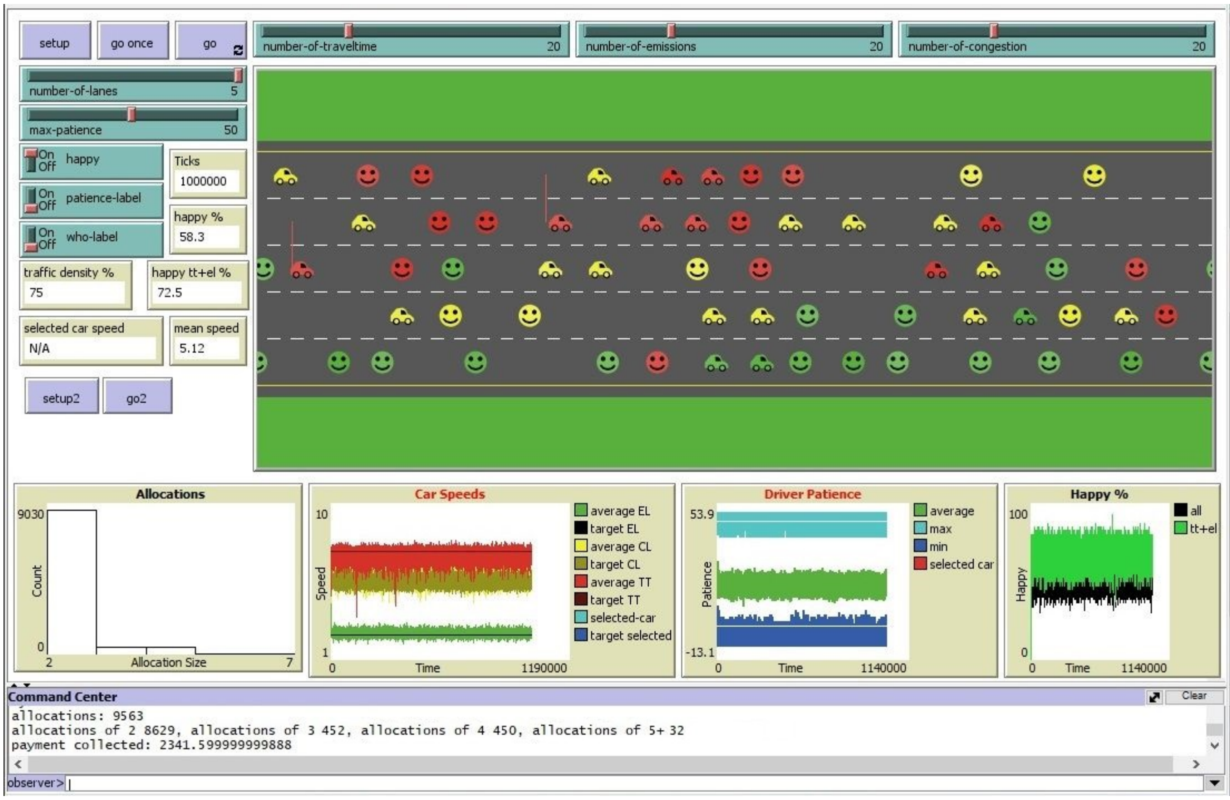

Figure 6.9: NetLogo Simulation Completed 1,000,000 Time Slice Test Run for Traffic Density of 75%

feature should be added, or can be specialized, e.g., public health or safety vehicles for which all AVs have to make room. They can also be non-autonomous vehicles. Something as simple as a turn signal from a non-autonomous vehicle can signify that a lane position is to become unavailable to other agents. However, negotiating such conditions would require the mechanism design model to be preempted by a different set of rules and is thus out of the current scope of this work. Finally, if separate mechanisms to enable the utilization of highway on and off ramps are added to the model, the mechanism can be fine tuned to a potential real world deployment.

6.5 CONCLUSION

Part II of this dissertation presented a game-theoretic multi-objective model for the allocation of lane positions and speed settings among autonomous vehicles on a multi-lane highway. Rooted in mechanism design, the approach delivers a social utility maximizing strategyproof mechanism for negotiation and conflict resolution. The localized nature of lane-changes ensures the scalability of the model. The system also benefits from performance advantages granted to it by its deployment at the edge of the Internet of Things.

The mechanism can be applied to AV systems when the greater good is more important than the individual welfare of an agent (e.g., to not put a strain on the infrastructure, to monetize, when there are a lot of public vehicles, etc.). In reality, the mechanism can be applied to IoT systems of any kind when the allocation of two or more resources is required. The valuation matrix can be easily extended with more dimensions and the distance calculation operation can be changed to reflect a decision pertinent to the application domain. The valuation matrix can be scaled down to a valuation vector when a single resource dimension is called for.

A prototyped NetLogo implementation was presented equipped with a limited set of shared objectives governing patterns of driving. The empirical model includes rule sets for determining action feasibility and for the ranking of eventualities. It demonstrates the ability of the system to create allocations that resolve conflict in favor of the collective good. Features of the system are adjustable to fit the deployment scenario.

Even though parts of the presented model are specific to the application domain, by making the context-appropriate changes, the approach can be implemented and utilized in any field with similar demands on its resources. The proposed mechanism heralds a step toward autonomous negotiation among machines and it equips interconnected smart cyber-physical systems with efficient and meaningful ways to privately communicate preferences to an arbitrating body. The model requires minimal resources and gives agents the ability to take part in a negotiation process free from manipulation and resulting in a socially satisfactory outcome.

CHAPTER 7

APPLICATIONS

The application domain for the models for automated negotiation presented in this work is that of driverless vehicles. However, both approaches can be adapted and utilized in just about any context that requires unsupervised parlaying among smart devices, even when those devices are heterogeneous. The argumentation prototype outlined in chapter 4 can be used in application scenarios exhibiting the need for democratic resolution of conflict in a distributed system. Through an alteration of the domain-specific operation, the mechanism design model delineated in chapter 6 can be applied in any context that requires a collection of smart devices to perform negotiation with other agents so that the collective good is maximized and the privacy of information is maintained.

Both systems can find limitless industrial applications. The choice between models is based on the scenario specifics - privacy of information, type of resource or task to be shared, type of beneficial outcome, level of system centralization, etc. Application opportunities can be found within smart cities - in transportation, aside from negotiated lane changes, argumentation and mechanism design can be used in vehicle platooning, parking spot assignment and in directing traffic at intersections. Additional deployment opportunities exist in autonomous multi-agent systems in mobile arenas like security, search and rescue, or environmental remediation where each approach can enable robot swarms to efficiently self-organize and divide the area of operations. Within the context of smart energy and smart living, either negotiation system can be a useful tool in energy conservation and reduction of waste - e.g., smart power supply controllers negotiating power consumption and down time with connected devices. In medicine and education, these algorithms can govern the allocation of available operating rooms and surgical teams or classrooms and teachers. Applications can also be found in fields currently employing notoriously unfair competition practices like entertainment ticket sales.

The presented models for unsupervised machine-to-machine argumentation and resource allocation among autonomous “things” can be applied anywhere within the IoT where conflict may arise when devices communicate, dialogue, debate, bargain, or negotiate with one another. The outcome of these processes will be the democratic or greater-good sharing of a single divisible resource, multiple or multi-dimensional resources, or tasks.

CHAPTER 8

CONCLUSION

The rapid deployment of heterogeneous smart devices within the burgeoning Internet of Things brings about the need for reliable algorithms for unsupervised negotiation among these agents. Regardless of the application domain, the distribution of tasks or resources among machines often requires human involvement, which may affect efficiency and fairness. Constructing dependable algorithmic models for fast and scalable delivery of democratic or socially beneficial outcomes through automated machine-to-machine negotiation requiring minimal resources is especially important in high stakes dynamic environments like real-time vehicle traffic. With a considerable push towards full autonomy of vehicles, the need to efficiently and reliably codify interactions among driverless transportation agents while simultaneously maintaining the safety of human occupants and material goods looms large. In order to reach the point at which driverless vehicle deployment is safe enough to be widely accepted and supported by the public, sophisticated unsupervised communication and negotiation mechanisms have to be put in place giving autonomous traffic the ability to self-regulate.

This dissertation presented two automated multi-objective machine-to-machine negotiation models to govern real-time continuous autonomous vehicle traffic on multi-lane roads where agents made decisions about the manner in which they progressed to their destination as it related to lane and speed selection. The first model, outlined in Part I and published in [89], used abstract argumentation augmented with social voting to deliver the democratic approval of lane change actions performed by autonomous vehicle agents on a highway. The second, proposed in Part II and published in [88], was rooted in game theory and represented a strategyproof monetizable mechanism designed to provide a social utility maximizing allocation of lane position and speed setting for agents in highway traffic. Two simulations presented a de facto proof of concept for the application of abstract argumentation and mechanism design concepts to lane changing on multi-lane roads. The models can be fine tuned, also in real time, to the appropriate traffic conditions by changing legal speed ranges for lanes, making lanes unavailable, selecting appropriate required safety distances, etc. It should be noted that highway ramps for entry and exit will have to be regulated differently as they do not represent continuous traffic flow that has time to recover from a denial to implement a selected action.

Participating in either arbitration system requires no advanced computational power. Even simple devices can be negotiation-enabled. As long as they can be furnished with an expert system capable of forming arguments based on sensed data, or are able to infer the ranking of their preferences from sensed data and their governing objective, agents can take part in the process. Should agents are not able to or cannot be sufficiently equipped to perform conflict resolution tasks or absorb the additional energy drain resulting from computation, these capabilities can be delegated within the edge of the local IoT.

Both presented prototypes successfully regulate traffic on a multi-lane road with very minimal resources. They conduct operations at the edge of the Internet of Things, taking advantage of the localized nature of effects of traffic changes to deliver the safe and efficient distribution of road resources. Both models allow for agents to maintain multiple private objectives and report preferences for their potentially ensuing actions. These prospective subsequent actions are valued within the decision making framework provided by the objective currently governing the vehicle's behavior. Decisions made by both algorithms rest on sensed data and inputs from agents reflecting their preferences. Both can be implemented as fully distributed or partially centralized. Involving an arbiter for decision making on the system level however, can keep inputs private on the individual agent level and can improve trust and performance. Both presented mechanisms are fully unsupervised as they require no human interaction whatsoever.

The two presented systems can also be combined into one. As a matter of fact, mechanism design can be successfully used for abstract M2M argumentation. Argumentation framework extensions provide the rules for the acceptability of arguments.

From a game theoretical perspective agents taking part in the argumentation process are adversarial players so their choice of strategy influences outcomes as well. Since agents' strategies, actions, and choice of arguments are dictated by selfish objectives, their perspectives on the acceptability of arguments will naturally differ too. Thus, what arguments are acceptable should be decided by employing the mechanism design approach [118].

Without further modifications, the two mechanisms can be deployed together and the governing mechanism can be changed on demand. The active model will dictate the type and amount of information it requires from agents but the results will remain satisfactory. This model switching can be utilized, for instance, when a highway changes from a free road to a stretch where tolls are to be collected for infrastructure upkeep or other purposes. In this case the social voting argumentation model will be preempted by the mechanism design model for the purposes of monetization. Other cases when one might want to choose the second model over the first due to the sheer amount of available action choices is when there is a large amount of extraneous vehicles, when the road owner wants greater control over traffic flow, or when there is construction, road damage or an accident and the mechanism needs greater authority over the availability of lanes and the ability to control speeds. Of course in all these cases, also when utilizing the computationally cheaper social voting argumentation mechanism, the system can announce e.g., temporary speed restrictions for lanes and vehicles will have to adhere to them and self-regulate within the framework of these new conditions. And certainly, since all autonomous vehicle agents' actions are preempted when the necessity to avoid a collision arises, agents will not blindly implement approved actions when the road situation does not allow it, which adds an additional layer of safety. Since both systems function in rounds, actions due to extraordinary circumstances are quickly recovered from in the following rounds. Thus, for traffic, both mechanisms can be utilized depending on the preferences of the body governing over the road - private owner, local government, etc.

Looking further into the future, both presented models can be applied to vehicle traffic that is not restricted to the ground. The Aeronautics Research Mission Directorate of NASA is leading the way to a new era in air transportation. The Urban Air Mobility program [97] is looking to extend mobility within metropolitan areas to a synergistic air transportation system for both passengers and cargo. Mechanisms will have to be put in place to arbitrate between traffic agents looking to occupy air space. Theoretically, there is no restriction in the proposed models on the number and spacial location of a lane position. If the autonomous vehicle has the means to attain the position, it can bid for it. So allocation of resources - lane and/or speed can be adapted to reflect lane positions in the 3-dimensional space, e.g., bid to move to a lane left, right, up, or down from the current position.

Of course, applications are not limited to autonomous vehicles. The two presented models can be applied to just about any application domain that requires automated negotiation for tasks or resources among machine agents. Both the social voting augmented abstract argumentation framework for the democratic resolution of conflict and the game-theoretical strategyproof mechanism delivering a social utility maximizing outcome represent a significant step towards smart agent autonomy in unsupervised negotiations.

REFERENCES

[1] U.S. Energy Information Administration. Monthly energy review, August 2016. https://www.eia.gov/totalenergy/data/monthly/archive/00351608.pdf, last accessed: January 19, 2021.

[2] Yasir Ali, Zuduo Zheng, Md. Mazharul Haque, and Meng Wang. A game theory-based approach for modelling mandatory lane-changing behaviour in a connected environment. *Transportation Research Part C: Emerging Technologies*, 106:220–242, 2019.

[3] Leila Amgoud and Jonathan Ben-Naim. Ranking-based semantics for argumentation frameworks. In Weiru Liu, V. S. Subrahmanian, and Jef Wijsen, editors, *Scalable Uncertainty Management*, pages 134–147, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[4] Leila Amgoud and Jonathan Ben-Naim. Argumentation-based ranking logics. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 1511–1519, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.

[5] Leila Amgoud, Simon Parsons, and Nicolas Maudet. Arguments, dialogue, and negotiation. In *Proceedings of the 14th European Conference on Artificial Intelligence*, ECAI'00, pages 338–342, Amsterdam, The Netherlands, The Netherlands, 2000. IOS Press.

[6] Dimitrios Antos and Avi Pfeffer. Representing bayesian games without a common prior. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, page 1457–1458, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[7] Kevin Ashton. That 'internet of things' thing. *RFID Journal*, June 2009. https://www.rfidjournal.com/articles/view?4986 (accessed February 2019).

[8] Isaac Asimov. *I, Robot (The Isaac Asimov Collection)*. Doubleday, 1988.

[9] Maksat Atagoziyev, Klaus W. Schmidt, and Ece G. Schmidt. Lane change scheduling for autonomous vehicles**this work was supported by the scientific and technological research council of turkey (tubitak) [award 115e372]. *IFAC-PapersOnLine*, 49(3):61–66, 2016.

[10] Haijian Bai, Jianfeng Shen, Liyang Wei, and Zhongxiang Feng. Accelerated lane-changing trajectory planning of automated vehicles with vehicle-to-vehicle collaboration. *Journal of Advanced Transportation*, 2017, 2017.

[11] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365—-410, 2011.

[12] Trevor J.M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10):619–641, 2007. Argumentation in Artificial Intelligence.

[13] Lofti Benmohamed and Semyon M. Meerkov. Feedback control of highway congestion by a fair on-ramp metering. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 3, pages 2437–2442, Dec 1994.

[14] Dimitri Bertsekas. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20(4):133–149, August 1990.

[15] David Bevly, Xiaolong Cao, Mikhail Gordon, Guchan Ozbilgin, David Kari, Brently Nelson, Jonathan Woodruff, Matthew Barth, Chase Murray, Arda Kurt, Keith Redmill, and Umit Ozguner. Lane change and merge maneuvers for connected and automated vehicles: A survey. *IEEE Transactions on Intelligent Vehicles*, 1(1):105–120, 2016.

[16] Lawrence Birnbaum. Argument molecules: A functional representation of argument structure. In *Proceedings of the Second National Conference on Artificial Intelligence*. AAAI Press, 1982.

[17] Lawrence Birnbaum, Margot Flowers, and Rod McGuire. Towards an ai model of argumentation. In *Proceedings of the First AAAI Conference on Artificial Intelligence*, AAAI'80, pages 313–315. AAAI Press, 1980.

[18] John C. Harsanyi. *Rational Behavior and Bargaining Equilibrium in Games and Social Situations*. 01 1977.

[19] Martin Caminada and Gabriella Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, Jan 2011.

[20] Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2):109, Nov 2009.

[21] Guido Cantelmo, Francesco Viti, Ernesto Cipriani, and Marialisa Nigro. A utility-based dynamic demand estimation model that explicitly accounts for activity scheduling and duration. *Transportation Research Procedia*, 23:440–459, 2017. Papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July, 2017.

[22] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Graduality in argumentation. *J. Artif. Intell. Res. (JAIR)*, 23:245–297, 01 2005.

[23] CBInsights. 40+ corporations working on autonomous vehicles, December 2020. https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/, last accessed: January 19, 2021.

[24] Chen Cheng. Mechanism design for enterprise transportation outsourcing based on combinatorial auction. In *2014 11th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–5, June 2014.

[25] Aaditya Prakash Chouhan, Gourinath Banda, and Kanishkar Jothibasu. A cooperative algorithm for lane sorting of autonomous vehicles. *IEEE Access*, 8:88759–88768, 2020.

[26] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, Sep 1971.

[27] Laurène Claussmann, Marc Revilloud, Dominique Gruyer, and Sébastien Glaser. A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):1826–1848, 2020.

[28] Jared Cohon and David Marks. A review and evaluation of multiobjective programming techniques. 11, 04 1975.

[29] European Commission. Europe on the move. Sustainable mobility for Europe: safe, connected, and clean, 2018. https://eur-lex.europa.eu/resource.html?uri=cellar%3A0e8b694e-59b5-11e8-ab41-01aa75ed71a1.0003.02/DOC_1&format=PDF, last accessed: July 9, 2019.

[30] Sylvie Coste-Marquis, Sébastien Konieczny, Pierre Marquis, and Mohand Akli Ouali. Weighted attacks in argumentation frameworks. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, pages 593–597. AAAI Press, 2012.

[31] Thanh-Son Dao, Christopher Michael Clark, and Jan Paul Huissoon. Optimized lane assignment using inter-vehicle communication. In *2007 IEEE Intelligent Vehicles Symposium*, pages 1217–1222, 2007.

[32] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 71–78, New York, NY, USA, 2006. ACM.

[33] Peter A. Diamond and James A. Mirrlees. Optimal taxation and public production i: Production efficiency. *The American Economic Review*, 61(1):8–27, 1971.

[34] Bruno Dias, Alexandre Santos, António Duarte Costa, Bruno Ribeiro, Fabio Gonçalves, Joaquim Henriques Macedo, Maria João Nicolau, Oscar Gama, and Susana Sousa. Agnostic and modular architecture for the development of

cooperative its applications. *Journal of Communications Software and Systems*, 14(3):218–227, 2018. cited By 0.

[35] Ernst D. Dickmanns, Reinhold Behringer, D. Dickmanns, T. Hildebrandt, Markus Maurer, F. Thomanek, and J. Schiehlen. The seeing passenger car 'vamors-p'. In *Proceedings of the Intelligent Vehicles '94 Symposium*, pages 68–73, 1994.

[36] Yannis Dimopoulos, Antonis C. Kakas, and Pavlos Moraitis. Argumentation based modelling of embedded agent dialogues. In Simon Parsons, Nicolas Maudet, Pavlos Moraitis, and Iyad Rahwan, editors, *Argumentation in Multi-Agent Systems*, pages 169–181, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[37] Yang Ding, Weichao Zhuang, Yahui Qian, and Hong Zhong. Trajectory planning for automated lane-change on a curved road for collision avoidance. In *WCX SAE World Congress Experience*. SAE International, Apr 2019.

[38] Shilp Dixit, Saber Fallah, Umberto Montanaro, Mehrdad Dianati, Alan Stevens, Francis Mccullough, and Alexandros Mouzakitis. Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects. *Annual Reviews in Control*, 45:76–86, 2018.

[39] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77:321–358, 1995.

[40] Paul E. Dunne. Prevarication in dispute protocols. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, ICAIL '03, pages 12–21, New York, NY, USA, 2003. ACM.

[41] Paul E. Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2):457–486, 2011.

[42] EU Economic Commission for Europe. Convention on road traffic. *Vienna Convention*, 1968.

[43] Sinan Eğilmez, João Martins, and João Leite. *Theory and Applications of Formal Argumentation*, chapter Extending Social Abstract Argumentation with Votes on Attacks, pages 16–31. Springer Berlin Heidelberg, 2014.

[44] Daniel J. Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[45] Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings. An agenda-based framework for multi-issue negotiation. *Artificial Intelligence*, 152(1):1–45, 2004.

[46] Giancarlo Fortino, Antonio Guerrieri, Wilma Russo, and Claudio Savaglio. Integration of agent-based and cloud computing for the smart objects-oriented iot. In *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 493–498, May 2014.

[47] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

[48] Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. Edge-centric computing: Vision and challenges. *SIGCOMM Comput. Commun. Rev.*, 45(5):37–42, September 2015.

[49] Noah J. Goodall. Can you program ethics into a self-driving car? *IEEE Spectrum*, 53(6):28–58, June 2016.

[50] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[51] Lie Guo, Ping-Shu Ge, Ming Yue, and Yi-Bing Zhao. Lane changing trajectory planning and tracking controller design for intelligent vehicle running on curved road. *Mathematical Problems in Engineering*, 2014, 2014.

[52] Linghui Han, David Z.W. Wang, and Chengjuan Zhu. The discrete-time second-best dynamic road pricing scheme. *Transportation Research Procedia*, 23:322–340, 2017. Papers Selected for the 22nd International Symposium on

Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July, 2017.

[53] Yu Han, Andreas Hegyi, Yufei Yuan, and Serge Hoogendoorn. Validation of an extended discrete first-order model with variable speed limits. *Transportation Research Part C: Emerging Technologies*, 83:1–17, 2017.

[54] Yusuke Hara. Behavioral mechanism design for transportation services: Laboratory experiments and preference elicitation cost. *Transportation Research Part B Methodological*, 115:231–245, 09 2018.

[55] Yusuke Hara and Eiji Hato. A car sharing auction with temporal-spatial od connection conditions. *Transportation Research Procedia*, 23:22–40, 2017. Papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July, 2017.

[56] Yusuke Hara and Eiji Hato. Analysis of dynamic decision-making in a bicycle-sharing auction using a dynamic discrete choice model. *Transportation*, 46(1):147–173, Feb 2019.

[57] John C. Harsanyi. Games with incomplete information played by "bayesian" players, i-iii. part ii. bayesian equilibrium points. *Management Science*, 14(5):320–334, 1968.

[58] John C. Harsanyi. A new theory of equilibrium selection for games with complete information. *Games and Economic Behavior*, 8(1):91–122, 1995.

[59] Jason Hartline. Mechanism design and approximation, 2017. http://jasonhartline.com/MDnA/, last accessed: October 16, 2018.

[60] Henry Hexmoor. On strategic coordination. *International Journal of Computational Intelligence: Theory and Practice*, 6:41–46, 01 2011.

[61] Henry Hexmoor, Raj Prothvi, and Santosh Srigiri. Speed adjustment for highway lane change. *International Journal of Information Systems and Computer Sciences*, 7(3):16–18, 2018.

[62] Henry Hexmoor and Kailash Yelasani. Economized sensor data processing with

vehicle platooning. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 12(6):428–433, 2018.

[63] Martin Hilscher, Sven Linker, Ernst-Rüdiger Olderog, and Anders P. Ravn. An abstract model for proving safety of multi-lane traffic manoeuvres. In Shengchao Qin and Zongyan Qiu, editors, *Formal Methods and Software Engineering*, pages 404–419, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[64] Man Lung Ho, Ping T. Chan, and A. B. Rad. Lane change algorithm for autonomous vehicles via virtual curvature method. *Journal of Advanced Transportation*, 43(1):47–70, 2009.

[65] Christine M. Hoehner, Carolyn E. Barlow, Peg Allen, and Mario Schootman. Commuting distance, cardiorespiratory fitness, and metabolic risk. *American journal of preventive medicine*, 42(6):571–578, 2012.

[66] Matthew Hoeing, Prithviraj Dasgupta, Plamen Petrov, and Stephen O'Hara. Auction-based multi-robot task allocation in comstar. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07, pages 1–8, New York, NY, USA, 2007. ACM.

[67] Leonid Hurwicz. The design of mechanisms for resource allocation. *The American Economic Review*, 63(2):1–30, 1973.

[68] Hubert Igliński and Maciej Babiak. Analysis of the potential of autonomous vehicles in reducing the emissions of greenhouse gases in road transport. *Procedia Engineering*, 192:353 – 358, 2017. 12th international scientific conference of young scientists on sustainable, modern and safe transport.

[69] Peter Jackson. *Introduction to Expert Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 1998.

[70] Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. *J. Log. Comput.*, 9:215–261, 04 1999.

[71] Ang Ji and David Levinson. A review of game theory models of lane changing.

Transportmetrica A: Transport Science, 16(3):1628–1647, 2020.

[72] Ajinkya A. Joshi, Diane L. Peters, and Jennifer M. Bastiaan. Autonomous lane change control using proportional-integral-derivative controller and bicycle model. In *WCX SAE World Congress Experience*. SAE International, Apr 2020.

[73] Antonis Kakas and Pavlos Moraitis. Adaptive agent negotiation via argumentation. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 384–391, New York, NY, USA, 2006. ACM.

[74] Maryam Kamali, Louise A. Dennis, Owen McAree, Michael Fisher, and Sandor M. Veres. Formal verification of autonomous vehicle platooning. *Science of Computer Programming*, 148:88 – 106, 2017. Special issue on Automated Verification of Critical Systems (AVoCS 2015).

[75] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015.

[76] John L. Pollock. Defeasible reasoning and degrees of justification. 1:7–22, 03 2010.

[77] Mariam Lafkihi, Shenle Pan, and Eric Ballot. Rule-based incentive mechanism design for a decentralised collaborative transport network. *International Journal of Production Research*, 0(0):1–17, 2019.

[78] Minghui Lai and Xiaoqiang Cai. *Algorithmic Mechanism Design for Collaboration in Large-Scale Transportation Networks*, pages 257–282. Springer International Publishing, 2019.

[79] Jorge Laval and Carlos Daganzo. Lane-changing in traffic stream. *Transportation Research Part B: Methodological*, 40B:251–264, 03 2006.

[80] João Leite and João Martins. Social abstract argumentation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume*

Volume Three, IJCAI'11, pages 2287–2292. AAAI Press, 2011.

[81] Hongluo Li and Yutao Luo. Study on steering angle input during the automated lane change of electric vehicle. In *Intelligent and Connected Vehicles Symposium*. SAE International, Sep 2017.

[82] Tingting Li, Jianping Wu, Ching-Yao Chan, Mingyu Liu, Chunli Zhu, Weixin Lu, and Kezhen Hu. A cooperative lane change model for connected and automated vehicles. *IEEE Access*, 8:54940–54951, 2020.

[83] Dianchao Lin, Li Li, and Saif Eddin Jabari. Pay to change lanes: A cooperative lane-changing strategy for connected/automated driving. *Transportation Research Part C: Emerging Technologies*, 105:550–564, 2019.

[84] Jennie Lioris, Ramtin Pedarsani, Fatma Yildiz Tascikaraoglu, and Pravin Varaiya. Platoons of connected vehicles can double throughput in urban roads. *Transportation Research Part C: Emerging Technologies*, 77:292 – 305, 2017.

[85] Marco Lippi and Paolo Torroni. Argument mining: A machine learning perspective. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation*, pages 163–176, Cham, 2015. Springer International Publishing.

[86] Todd Litman. Autonomous vehicle implementation predictions. implications for transport planning, June 2020. https://vtpi.org/avip.pdf, last accessed: January 19, 2021.

[87] Ellie Lovellette and Henry Hexmoor. Voronoi diagrams for automated argumentations among internet of things. *Multiagent and Grid Systems*, 12(4):303–318, 2016.

[88] Ellie Lovellette and Henry Hexmoor. Lane and speed allocation mechanism for autonomous vehicle agents on a multi-lane highway. *Internet of Things*, page 100356, 2021.

[89] Ellie Lovellette, Henry Hexmoor, and Kane Rodriguez. Automated argumentation

for collaboration among cyber-physical system actors at the edge of the internet of things. *Internet of Things*, 5:84–96, 2019.

[90] Yugong Luo, Gang Yang, Mingchang Xu, Zhaobo Qin, and Keqiang Li. Cooperative lane-change maneuver for multiple automated vehicles on a highway. *Automotive Innovation*, 2(3):157–168, 2019.

[91] Eric Maskin. Nash equilibrium and mechanism design. *Games and Economic Behavior*, 71:9–11, 2011.

[92] Paul-Amaury Matt and Francesca Toni. A game-theoretic measure of argument strength for abstract argumentation. In Steffen Hölldobler, Carsten Lutz, and Heinrich Wansing, editors, *Logics in Artificial Intelligence*, pages 285–297, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[93] Friedemann Mattern and Christian Floerkemeier. *From the Internet of Computers to the Internet of Things*, pages 242–259. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[94] H.-G. Meissner and E. D. Dickmanns. Control of an unstable plant by computer vision. In Thomas S. Huang, editor, *Image Sequence Processing and Dynamic Scene Analysis*, pages 532–548, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg.

[95] Martin Moẑina, Jure Ẑabkar, and Ivan Bratko. Argument based machine learning. *Artificial Intelligence*, 171(10):922–937, 2007. Argumentation in Artificial Intelligence.

[96] Roger B. Myerson. Perspectives on mechanism design in economic theory. *The American Economic Review*, 98(3):586–603, 2008.

[97] NASA. Urban air mobility, 2019. https://www.nasa.gov/uam, last accessed: November 25, 2019.

[98] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

[99] United Nations. Sustainable development goal 3, target 3.6, 2016.

https://unstats.un.org/sdgs/metadata/files/Metadata-03-06-01.pdf, last accessed: July 9, 2019.

[100] Maximilian Naumann, Hendrik Königshof, and Christoph Stiller. Provably safe and smooth lane changes in mixed traffic. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1832–1837, 2019.

[101] Jie Ni, Jingwen Han, and Fei Dong. Multivehicle cooperative lane change control strategy for intelligent connected vehicle. *Journal of Advanced Transportation*, 2020, 2020.

[102] Jianqiang Nie, Jian Zhang, Wanting Ding, Xia Wan, Xiaoxuan Chen, and Bin Ran. Decentralized cooperative lane-changing decision-making for connected autonomous vehicles*. *IEEE Access*, 4:9413–9420, 2016.

[103] Julia Nilsson, Mattias Brannstrom, Erik Coelingh, and Jonas Fredriksson. Lane change maneuvers for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1087–1096, 2017.

[104] Noam Nisan. *Introduction to Mechanism Design (for Computer Scientists)*, pages 209—-242. Cambridge University Press, 2007.

[105] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[106] National Conference of State Legislatures. Autonomous vehicles state bill tracking database, 2021. http://www.ncsl.org/research/transportation/autonomous-vehicles-legislative-database.aspx, last accessed: January 19, 2021.

[107] U.S. Department of Transportation. Critical reasons for crashes investigated in the national motor vehicle crash causation survey, 2015. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115, last accessed: July 9, 2019.

[108] U.S. Department of Transportation. 2016 fatal motor vehicle crashes: Overview, 2016. https://crashstats.nhtsa.dot.gov/Api/Public/Publication/812456, last

accessed: July 9, 2019.

[109] World Health Organization. Global status report on road safety, 2018. https://apps.who.int/iris/bitstream/handle/10665/277370/WHO-NMH-NVI-18.20-eng.pdf, last accessed: July 9, 2019.

[110] World Health Organization. Improving global road safety, 2020. https://www.un.org/pga/74/wp-content/uploads/sites/99/2020/08/Draft-Resolution-Road-Safety.pdf, last accessed: January 19, 2021.

[111] Minoru Osawa, Haoran Fu, and Takashi Akamatsu. First-best dynamic assignment of commuters with endogenous heterogeneities in a corridor network. *Transportation Research Part B: Methodological*, 117:811–831, 2018. TRB:ISTTT-22.

[112] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[113] Simon Parsons. Cognitive carpentry: A blueprint for how to build a person by john l. pollock, mit press, 1995, isbn 0-262-16152-4, pp 377, £29.50. *The Knowledge Engineering Review*, 14(1):91—-95, 1999.

[114] Christian Pek, Peter Zahn, and Matthias Althoff. Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1477–1483, June 2017.

[115] Donald Pennington. *The Social Psychology of Behaviour in Small Groups*. Routledge, 2002.

[116] Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1-2):25–75, 1997.

[117] Bangjun Qiao and Xiaodong Wu. Lane change control of autonomous vehicle with real-time rerouting function. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1317–1322, 2019.

[118] Iyad Rahwan and Kate Larson. Mechanism design for abstract argumentation. pages 1031–1038, 01 2008.

[119] Iyad Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, Peter Mcburney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *Knowl. Eng. Rev.*, 18(4):343–375, December 2003.

[120] Iyad Rahwan and Fernando Tohmé. Collective argument evaluation as judgement aggregation. pages 417–424, 01 2010.

[121] Anand S. Rao and Michael P. Georgeff. An abstract architecture for rational agents. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, KR'92, pages 439–449, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[122] Stephen J. Rassenti, Vernon L. Smith, and Robert L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics*, 13(2):402–417, 1982.

[123] Ju Ren, Yi Pan, Andrzej Goscinski, and Raheem A. Beyah. Edge computing for the internet of things. *IEEE Network*, 32(1):6–7, Jan 2018.

[124] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. Survey on scenario-based safety assessment of automated vehicles. *IEEE Access*, 8:87456–87477, 2020.

[125] Nicolás D. Rotstein, Martín O. Moguillansky, Alejandro J. García, and Guillermo R. Simari. A dynamic argumentation framework. In *Proceedings of the 2010 Conference on Computational Models of Argument: Proceedings of COMMA 2010*, pages 427–438, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[126] Tuomas W. Sandholm. Multiagent systems. chapter Distributed Rational Decision Making, pages 201–258. MIT Press, Cambridge, MA, USA, 1999.

[127] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, Jan 2017.

[128] Klaus W. Schmidt. Cooperative adaptive cruise control for vehicle following during lane changes**this work was supported by the scientific and technological research council of turkey (tubitak) [award 115e372]. *IFAC-PapersOnLine*, 50(1):12582–12587, 2017. 20th IFAC World Congress.

[129] Stefano Severi, Francesco Sottile, Giuseppe Abreu, Claudio Pastrone, Maurizio Spirito, and Friedbert Berens. M2m technologies: Enablers for a pervasive internet of things. In *2014 European Conference on Networks and Communications (EuCNC)*, pages 1–5, June 2014.

[130] Guillermo Simari and Iyad Rahwan. *Argumentation in Artificial Intelligence*. 01 2009.

[131] Viktor Skrickij, Eldar Šabanovič, and Vidas Žuraulis. Autonomous road vehicles: recent issues and expectations. *IET Intelligent Transport Systems*, 14(6):471–479, 2020.

[132] A Smiley and K.A. Brookhuis. Alcohol, drugs and safety. *In Road Users and Traffic Safety*, pages 83–105, 01 1987.

[133] P. B. Sujit and Randy Beard. Distributed sequential auctions for multiple uav task allocation. In *2007 American Control Conference*, pages 3955–3960, July 2007.

[134] Katia P. Sycara. Argumentation: Planning other agents' plans. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'89, pages 517–523, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[135] Alireza Talebpour, Hani S. Mahmassani, and Samer H. Hamdar. Modeling lane-changing behavior in a connected environment: A game theory approach. *Transportation Research Part C: Emerging Technologies*, 59:216–232, 2015. Special Issue on International Symposium on Transportation and Traffic Theory.

[136] Pete Thomas, Andrew Morris, Rachel Talbot, and Helen Fagerlind. Identifying the causes of road crashes in europe. *Annals of advances in automotive medicine / Annual Scientific Conference ... Association for the Advancement of Automotive*

Medicine. Association for the Advancement of Automotive Medicine. Scientific Conference, 57:13–22, 09 2013.

[137] Qiong Tian, Hai-Jun Huang, Hai Yang, and Ziyou Gao. Efficiency and equity of ramp control and capacity allocation mechanisms in a freeway corridor. *Transportation Research Part C: Emerging Technologies*, 20(1):126–143, 2012. Special issue on Optimization in Public Transport+ISTT2011.

[138] Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, 2 edition, 2003.

[139] Sadayuki Tsugawa, Teruo Yatabe, Takeshi Hirose, and Shuntetsu Matsumoto. An automobile with artificial intelligence. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'79, page 893–895, San Francisco, CA, USA, 1979. Morgan Kaufmann Publishers Inc.

[140] Albert W. Tucker. The mathematics of tucker: A sampler. *The Two-Year College Mathematics Journal*, 14(3):228–232, 1983.

[141] UN United Nations. Convention on road traffic. *Geneva Convention*, 1949.

[142] Thomas Vanoutrive. Don't think of them as roads. think of them as road transport markets: Congestion pricing as a neoliberal political project. *Progress in Planning*, 117:1–21, 2017.

[143] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[144] William S. Vickrey. Congestion theory and transport investment. *The American Economic Review*, 59(2):251–260, 1969.

[145] José M. Viegas. Making urban road pricing acceptable and effective: searching for quality and equity in urban mobility. *Transport Policy*, 8(4):289–294, 2001.

[146] Sebastian vom Dorff, Bert Böddeker, Maximilian Kneissl, and Martin Fränzle. A fail-safe architecture for automated driving. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 828–833, 2020.

[147] Kentaro Wada and Takashi Akamatsu. A hybrid implementation mechanism of tradable network permits system which obviates path enumeration: An auction mechanism with day-to-day capacity control. *Transportation Research Part E: Logistics and Transportation Review*, 60:94–112, 2013.

[148] Chang Wang, Qinyu Sun, Zhen Li, and Hongjia Zhang. Human-like lane change decision model for autonomous vehicles that considers the risk perception of drivers in mixed traffic. *Sensors*, 20:2259, 04 2020.

[149] Di Wang, Manjiang Hu, Yunpeng Wang, Jianqiang Wang, Hongmao Qin, and Yougang Bian. Model predictive control–based cooperative lane change strategy for improving traffic flow. *Advances in Mechanical Engineering*, 8(2):1687814016632992, 2016.

[150] Meng Wang, Serge P. Hoogendoorn, Winnie Daamen, Bart van Arem, and Riender Happee. Game theoretic approach for predictive lane-changing and car-following control. *Transportation Research Part C: Emerging Technologies*, 58:73–92, 2015.

[151] Mingyu Wang, Zijian Wang, Shreyasha Paudel, and Mac Schwager. Safe distributed lane change maneuvers for multiple autonomous vehicles using buffered input cells. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4678–4684, 2018.

[152] Di Wu, Yafeng Yin, Siriphong Lawphongpanich, and Hai Yang. Design of more equitable congestion pricing and tradable credit schemes for multimodal transportation networks. *Transportation Research Part B: Methodological*, 46(9):1273–1287, 2012.

[153] Han Yu, Zhiqi Shen, and Cyril Leung. From internet of things to internet of agents. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 1054–1057, Aug 2013.

[154] Hongtao Yu, H. Eric Tseng, and Reza Langari. A human-like game theory-based

controller for automatic lane changing. *Transportation Research Part C: Emerging Technologies*, 88:140–158, 2018.

[155] Wenwen Zhang, Subhrajit Guhathakurta, Jinqi Fang, and Ge Zhang. Exploring the impact of shared autonomous vehicles on urban parking demand: An agent-based simulation approach. *Sustainable Cities and Society*, 19:34 – 45, 2015.

[156] Bo Zou, Nabin Kafle, Ouri Wolfson, and Jie (Jane) Lin. A mechanism design based approach to solving parking slot assignment in the information era. *Transportation Research Part B: Methodological*, 81:631–653, 2015. Optimization of Urban Transportation Service Networks.

VITA

Graduate School
Southern Illinois University

Ellie B. Lovellette

elovellette@gmail.com

Technische Universität Darmstadt, Germany
Bachelor of Science, Mathematics and Computer Science, July 2007

Southern Illinois University Edwardsville
Master of Science, Computer Science, May 2011

Dissertation Paper Title:
Automated Algorithmic Machine-to-Machine Negotiation for Lane Changes Performed by Driverless Vehicles at the Edge of the Internet of Things

Major Professor: Dr. H. Hexmoor

Publications:

Ellie Lovellette and Henry Hexmoor. 2021. Lane and Speed Allocation Mechanism for Autonomous Vehicle Agents on a Multi-Lane Highway. Internet of Things, page 100356.

Dennis J Bouvier, Ellie Lovellette, and John Matta. 2021. Overnight Feedback Reduces Late Submissions on Programming Projects in CS1. In Australasian Computing Education Conference (ACE '21), Virtual, SA, Australia. ACM, New York, NY, USA, 5 pages.

Dennis Bouvier, Ellie Lovellette, John Matta, Jing Bai, Jacqueline Chetty, Stan Kurkovsky, and Jia Wan. 2019. Factors Affecting the Adoption of Peer Instruction in Computing Courses. In Proceedings of the Working Group Reports on Global Computing Education (CompEd-WGR '19). Association for Computing Machinery, New York, NY, USA, 1–25.

Ellie Lovellette, Henry Hexmoor, and Kane Rodrieguez. 2019. Automated Argumentation for Collaboration Among Cyber-Physical Actors at the Edge of Internet of Things. Internet of Things, Volume 5, pp. 84-96, Elsevier.

Ellie Lovellette and Henry Hexmoor. 2017. Voronoi Diagrams for Automated Argumentations Among Internet of Things. Multiagent and Grid Systems. Volume 12, pp. 303-318.

Ellie Lovellette, John Matta, Dennis Bouvier, and Roger Frye. 2017. Just the Numbers: An Investigation of Contextualization of Problems for Novice Programmers. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). ACM, New York, NY, USA, pp. 393-398.

Dennis Bouvier, Ellie Lovellette, John Matta, Bedour Alshaigy, Brett Becker, Michelle Craig, Jana Jacková, Robert McCartney, Kate Sanders, and Mark Zarb. 2016. Novice Programmers and the Problem Description Effect. In Proceedings of the 2016 ITiCSE Working Group Reports (ITiCSE '16). ACM, New York, NY, USA, pp. 103-118.

Ellie Lovellette and Henry Hexmoor. 2016 Abstract Argumentation Using Voronoi Diagrams. International Conference on Collaboration Technologies and Systems (CTS), Orlando, FL, pp. 472-477.