

Fachhochschule Dortmund
University of Applied Sciences and Arts
Embedded Systems Engineering

Modeling and Controlling Autonomous Transportation Systems in Smart Cities

Master Thesis

**Fachhochschule
Dortmund**

University of Applied Sciences and Arts

Author: BRODO, Luca
Matriculation Number: 7215156
Supervisor: Prof. Dr. rer. nat. Henkler Stefan
Date: 04.2024

Abstract

Declaration

I hereby confirm, that I have written the Master Thesis at hand independently – in case of a group work: my respectively designated part of the work -, that I have not used any sources or materials other than those stated, and that I have highlighted any citations properly.

Contents

1	Motivation	1
1.1	Thesis Contributions and Organization	2
2	Preliminaries	3
2.1	General Definitions	3
2.2	Model Predictive Control	6
2.3	Graph Transformation Theory	8
3	Literature Review	14
3.1	Fleet and Transportation Network	14
3.2	ATS Challanges	15
3.3	Applications of Graph Transformation Systems	18
4	Time-Invariant Model of ATS	20
4.1	Vehicle-centric Model	20
4.2	Analysing ATS challenges	27
4.3	The Complete ATS Management Problem Formulation	37
4.4	Use Case Analysis	39
4.5	Conclusion	47

5 Model Predictive Control of an ATS	49
5.1 Linear-Discrete Time Model	49
5.2 Problem Formulation	56
5.3 Reduced Connectivity Schema	63
5.4 Use Case Analysis	65
5.5 Conclusion	70
6 Summary and Outlook	71
A Additional Information on Rebalancing-Free Simulation	1
B Additional Information on the Simulation with Rebalancing	4
C Additional Information on the MPC for ATS	6
C.1 Propagation and Definition of the variables in \mathcal{X}	6
C.2 Further Proof of Stability	7
D Additional Information on the MPC Use Case	12
D.1 Statistics on Taxis in NYC	12
D.2 RCS Evolution	13

List of Figures

3.1	BPR Model and Approximation	17
4.1	Different Charging Profiles According To The Model	23
4.2	Example of a Sensible Request assignment	27
4.3	Simplified Example for the Rebalancing Strategy	35
4.4	Manhattan's Road Network Representation	40
4.5	Overview of System's Performance without Congestions	43
4.6	Overview of System's Performance for Routing-less Simulation 3.	44
4.7	Overview of System's Performance for Routing-less Simulation 4.	45
4.8	Road Usage and State of Charge of an exemplary Set of Vehicles for the Simulation 2 and 3 with Rebalancing.	48
5.1	Cruising speed in function of traffic density	51
5.2	RCS Dimensions for Different Application of Rule 2	67
5.3	Decrease in Number of Requests Over Time	68
5.4	Evolution of The Vehicle Usage Over Time	69
A.1	Overview of System's Performance with Congestion Limits and Less Charging Time	1
A.2	Overview of System's Performance with Congestion Limits and Less Charging Time	2

A.3	State of Charge of an exemplary Set of Vehicles for Simulation 5	2
A.4	State of Charge of an exemplary Set of Vehicles for Simulation 3 and 4	3
B.1	Road Usage and State of Charge of an exemplary Set of Vehicles for the Simulation 1 with Rebalancing	4
B.2	Road Usage and State of Charge of an exemplary Set of Vehicles for the Simulation 4 with Rebalancing	5
D.1	Depiction of The Evolution of the RCS	13

List of Tables

4.1	Parameters choice for the Chargin Stations	25
4.2	Details of the Rebalancing-less Simulation	42
4.3	Depot Distribution Used During the Simulation	42
4.4	Overview of the System's Performance for Routing-less Simulation.	42
4.5	Details of the Simulation with Vehicle Rebalancing	46
4.6	Performance of the Simulation with Vehicle Rebalancing	46
5.1	Various information about the MPC Simulations	66
5.2	Rule Application for RCS	67
C.1	Truth table for the propagation of g_{ji}^a	6
C.2	Truth table for the propagation of p_{ji}^a	7
C.3	Truth table for the constrains on w_{ji}^a and v_{ji}^a	7

Chapter 1

Motivation

Traffic management is a cornerstone of modern urban living, intricately woven into the fabric of our daily routines. As populations continue to surge and urbanization accelerates, the demands on transportation infrastructure intensify correspondingly. In the midst of these changes, the need for efficient traffic control and road network management becomes not just desirable, but necessary. At its core, efficient traffic control isn't merely about facilitating the movement of vehicles from point A to point B. On the contrary, it impacts much more aspects of societal well-being economic vitality and environmental sustainability. Consider, for instance, the consequences of traffic inefficiencies. Congestions not only waste valuable time but also incurs substantial costs to individuals and businesses alike. Productivity takes a hit as workers are stopped in gridlocked lanes, deliveries are delayed, and commerce grinds to a sluggish pace. Congestion in the United States exacts a hefty toll on the economy, amounting to approximately 121 billion dollars annually, equivalent to 1% of the nation's GDP ([49]). This figure encompasses not just the significant loss of 5.5 billion hours to traffic jams but also the burning of an extra 2.9 billion gallons of fuel. Furthermore, these calculations do not fully consider the potential costs stemming from negative side-effects such as vehicular emissions (including greenhouse gases and particulate matter) ([41]), travel-time uncertainty ([7]), and an elevated risk of accidents ([22]). The exponential growth in urbanization and vehicle ownership has intensified this problem, leading to increased travel times, fuel consumption, air pollution, and stress levels for commuters. Efficient traffic control mechanisms are essential to mitigate these negative consequences and create sustainable, livable urban environments. By optimizing the flow of vehicles, reducing congestion, and minimizing travel times, effective traffic management enhances economic productivity, improves air quality, and fosters healthier, more vibrant communities. Moreover, it promotes equitable access to transportation resources, facilitating social inclusion and enhancing overall quality of life for urban residents.

As we move towards the era of smart cities, the integration of Vehicle-to-Everything (V2X) communication systems and the emergence of AVs present

unprecedented opportunities to revolutionize traffic management. AVs, equipped with advanced sensors and algorithms, have the potential to further enhance traffic control by optimizing routes, adjusting speeds, and minimizing unnecessary stops. These vehicles can navigate through urban environments with precision, adapting to changing road conditions and traffic patterns in ways that human drivers cannot. We are, in other words, moving towards the era of Autonomous Transportation Systems (ATS), where AVs are set to revolutionize urban mobility and deliveries. ATSs must be built upon four main pillars. ATSs must (*i*) ensure optimal quality-of-service by incorporating metrics such as efficiency, reliability and safety; (*ii*) reduce and optimize road utilization while minimizing the number of necessary non-pedestrian zones and congestions, through strategic traffic management and routing algorithms; (*iii*) consider operational constraints, such as charging limitations and vehicle payloads; (*iv*) minimize the environmental impact. However, to fully realize the benefits of ATS, it is necessary to lay stable foundations to build upon. These include investing in robust infrastructure capable of supporting these technologies, establishing regulatory frameworks to ensure safety and interoperability, and, more importantly, establish novel models and techniques to manage and control fleet of autonomous vehicles. In light of these considerations, this thesis seeks to explore innovative approaches to traffic control that leverage advanced technologies and data analytics to optimize transportation systems and to ensure that the main pillars of ATSs are respected. Respecting these pillars serve as the theoretical framework for designing and implementing effective ATSs, contributing to the advancement of transportation systems with a focus on efficiency, sustainability, and overall societal well-being.

1.1 Thesis Contributions and Organization

The main objective of this thesis is to establish the necessary foundations towards a systematic approach to analyse, control and optimize ATSs systems with a particular focus on urban environment. The following is a summary of the organization and the contributions in this work.

Chapter 2: This chapter presents the theoretical background knowledge necessary, focusing on Model Predictive Control and Graph Grammar.

Chapter 4: An holistic time-invariant, vehicle centric model for ATSs is proposed. This is a general, yet efficient and adaptable, representation of an ATS. Furthermore, the Complete ATS Management problem is formulated, which, within the previously described model, aims at solving the main challenges in ATSs.

Chapter 5: This chapter develops a novel linear time-discrete model for ATS with the main objective of an optimized real-time application. In this regard, an MPC formulation is derived and proven to be stable in the Lyapunov sense. Furthermore, a possible optimization with GTS is proposed.

Chapter 6: Multiple directions for future researches are still to be explored. This chapter includes a discussion and future outlooks.

Chapter 2

Preliminaries

This chapter provides essential background information and definitions that form the foundation for the subsequent chapters in the document. Firstly, in Section 2.1, a plethora of useful general definitions are provided in order to facilitate the next sections and, as a result, the rest of the work as well. Section 2.2 introduces the concept of Model Predictive Control (MPC) as an advanced control strategy for dynamic systems, particularly focusing on the linear case. The linear MPC problem is formulated with a finite prediction horizon, system dynamics, and constraints on state and control variables. Additionally, the stability of MPC systems is briefly discussed, emphasizing the importance of demonstrating recursive feasibility and establishing stability in the sense of Lyapunov.

2.1 General Definitions

Definition 2.1: Positive Semidefinite A symmetric matrix A is said to be positive semidefinite if, for any non-zero column vector x , the following inequality holds:

$$x^T Ax \geq 0 \quad (2.1)$$

Here, x^T represents the transpose of the vector x , and $x^T Ax$ is the quadratic form associated with the matrix A . Another way to express positive semidefiniteness is in terms of eigenvalues. A symmetric matrix A is positive semidefinite if and only if all of its eigenvalues are non-negative.

Mathematically, $A \succeq 0$ indicates positive semidefiniteness.

Definition 2.2: Positive Definite A positive definite matrix is a more specific case of a positive semidefinite matrix. A symmetric matrix A is said to be positive definite if, for any non-zero column vector x , the following inequality holds:

$$x^T Ax > 0 \quad (2.2)$$

Here, x^T represents the transpose of the vector x , and $x^T Ax$ is the quadratic form associated with the matrix A . Alternatively, positive definiteness in terms of eigenvalues is expressed as follows: A symmetric matrix A is positive definite if and only if all of its eigenvalues are strictly positive. In mathematical notation, $A \succ 0$ indicates positive definiteness.

Definition 2.3: Half-Space A half-space in n -dimensional space is defined by a linear inequality. The general form of a half-space is given by:

$$H = \{x \in \mathbb{R}^n \mid a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b\} \quad (2.3)$$

Alternatively, in vector form:

$$H = \{x \in \mathbb{R}^n \mid a^T x \leq b\} \quad (2.4)$$

Here, a_1, a_2, \dots, a_n are real constants, and b is a real constant. The vector (a_1, a_2, \dots, a_n) is the normal vector to the hyperplane defining the boundary of the half-space. The inequality $a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$ represents the side of the hyperplane where the half-space lies. Geometrically, a half-space is one of the two regions divided by a hyperplane. If the inequality is strict ($<$ instead of \leq), the half-space does not include points on the hyperplane itself. If the inequality is non-strict (\leq), the half-space includes points on the hyperplane. In two dimensions ($n = 2$), a half-space is a region divided by a straight line. In three dimensions ($n = 3$), it is a region divided by a plane, and so on. The intersection of multiple half-spaces forms a polyhedral region.

Definition 2.4: Polyhedral Region A polyhedral region is a geometric object in three-dimensional space that is defined as the intersection of a finite number of half-spaces. Formally, a polyhedral region P in three-dimensional space can be expressed as:

$$P = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{a}_i \cdot \mathbf{x} \leq b_i, \quad i = 1, 2, \dots, n\}$$

where \mathbf{x} is a three-dimensional vector representing a point in space, \mathbf{a}_i are vectors normal to the defining planes, and b_i are constants determining the position of these planes. The inequalities $\mathbf{a}_i \cdot \mathbf{x} \leq b_i$ specify that the point \mathbf{x} lies on or inside the half-space defined by the corresponding plane.

Definition 2.5: Recursive feasibility The MPC problem is deemed recursively feasible if, for all feasible initial states, assurance of feasibility is maintained at every state along the closed-loop trajectory.

Definition 2.6: Positively Invariant Set A set Ω is said to be a positively invariant set for a system $x(k+1) = f_\kappa(x(k))$, if

$$x(k) \in \Omega \implies x(k+1) \in \Omega \quad \forall k \in \mathbb{N}$$

Definition 2.7: Comparision Functions

For $\mathbb{R}_0^+[0, \infty)$, the following comparison functions exist

$$\mathcal{K} := \left\{ \alpha : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \middle| \alpha \text{ is continuous and strictly increasing with } \alpha(0) = 0 \right\} \quad (2.5)$$

$$\mathcal{K}_\infty := \left\{ \alpha : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \middle| \alpha \in \mathcal{K} \text{ and unbounded} \right\} \quad (2.6)$$

$$\mathcal{KL} := \left\{ \beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \middle| \begin{array}{l} \beta \text{ continuous} \\ \beta(\cdot, t) \in \mathcal{K} \quad \forall t \in \mathbb{R}_0^+ \\ \beta(r, \cdot) \text{ is strictly decreasing to 0 } \forall r \in \mathbb{R}_0^+ \end{array} \right\} \quad (2.7)$$

Definition 2.8: Lyapunov Function

Let $Y \subseteq X$ be a forward invariant set and $x^* \in X$. A function $V : Y \rightarrow \mathbb{R}^+$ is called a Lyapunov function for $x^+ = g(x)$ if the following two conditions hold for all $x \in Y$:

1. There exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ such that

$$\alpha_1(\|x - x^*\|) \leq V(x) \leq \alpha_2(\|x - x^*\|).$$

2. There exists $\alpha_V \in \mathcal{K}$ such that

$$V(x^+) \leq V(x) - \alpha_V(\|x - x^*\|).$$

Definition 2.9: Lyapunov Function.

Let \mathcal{X} be a positively invariant set for a system $x(k+1) = f_\kappa(x(k))$ containing a neighbour of the origin in its interior. A function $V : \mathcal{X} \rightarrow \mathbb{R}_+$ is called a Lyapunov function in \mathcal{X} if for all $x \in \mathcal{X}$:

$$\begin{aligned} V(x) &> 0 \quad \forall x \neq 0 \\ V(x) &= 0 \\ V(x(k+1)) - V(x(k)) &\leq -\alpha(x) \end{aligned}$$

2.2 Model Predictive Control

Model Predictive Control (MPC) is an advanced control strategy used for dynamic systems. In the case of a linear time-invariant system, the mathematical description of MPC involves optimizing a cost function over a finite prediction horizon while subject to system dynamics and constraints.

For the purpose of this work, we will only consider the linear case of the MPC. For more details, please refer to [44].

Consider a linear system described by the following state-space equations:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \quad (2.8)$$

where:

- x_k is the state vector at time k
- u_k is the control input (or control variable) at time k
- y_k is the output at time k
- A , B , and C are matrices that define the system dynamics.

The objective is to minimize a cost function J over a finite prediction horizon N . The cost function is typically defined as the sum of a quadratic performance index over the prediction horizon.

The choice of cost function is critical and highly depends on the final objective. For example, in case the objective is to track a reference signal, a typical cost function assumes the form described in (2.9).

$$J = \sum_{i=0}^{N-1} [(y_{k+i} - r_{k+i})^T Q (y_{k+i} - r_{k+i}) + u_{k+i}^T R u_{k+i}] \quad (2.9)$$

where

- r_{k+i} is the reference trajectory at time $k + i$
- $Q \succeq 0$ is the weighting matrix for the state error
- $R \succ 0$ is the weighting matrix for the control input

Alternatively, another typical cost function has the followin form,

$$J = x_N^T P x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \quad (2.10)$$

where

- x_i is the state vector at time i

- u_i is the control variable at time i
- $Q \succeq 0$ is the weighting matrix for the state error
- $P \succeq 0$ is the weighting matrix indicating the terminal cost
- $R \succ 0$ is the weighting matrix for the control input.

In (2.10), the term $x_N^T P x_N$ is used to mitigate the fact that the MPC is minimizing over a limited time-horizon by ensuring that the system converges to a desired state by the end of the prediction horizon. While the infinite horizon formulation, i.e. $N = \infty$ and $P = 0$, possesses nice properties, such as its robustness and the perfect tradeoff between short- and long-term benefits of actions, it can not be computed in practice. Intuitively, since MPC is dealing with optimization problems under constraints, an infinite horizon will bring to having an infinite amount of variables. This is therefore the motivation behind why the horizon is usually shortened to N steps and the terminal cost is introduced.

The Linear MPC problem, therefore, is formulated as follows.

$$\begin{aligned}
J^* = \min & \quad x_N^T P x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \\
\text{s.t. } & \quad x_{k+i+1} = A x_{k+i} + B u_{k+i} \quad \text{for } i = 0, 1, \dots, N-1 \\
& \quad x_{k+i} \in \mathcal{X}, u_{k+i} \in \mathcal{U} \quad \text{for } i = 0, 1, \dots, N-1 \\
& \quad x_{k+N} \in \mathcal{X}_f \\
& \quad x_k = x(k)
\end{aligned} \tag{2.11}$$

where J^* is the global minimum of J and $\mathcal{X}, \mathcal{X}_f, \mathcal{U}$ are polyhedral regions. These are used to indicate the constraints over the state and control variables, which have the following form:

$$\begin{aligned}
x_{\min} &\leq x_{k+i} \leq x_{\max} \quad \text{for } i = 0, 1, \dots, N \\
x_{\min|f} &\leq x_{k+i} \leq x_{\max|f} \quad \text{for } i = 0, 1, \dots, N \\
u_{\min} &\leq u_{k+i} \leq u_{\max} \quad \text{for } i = 0, 1, \dots, N-1
\end{aligned}$$

Furthermore, the optimization problem is subject to system dynamics:

$$x_{k+i+1} = A x_{k+i} + B u_{k+i} \quad \text{for } i = 0, 1, \dots, N-1$$

The solution to this optimization problem provides the optimal control sequence $u_k^*, u_{k+1}^*, \dots, u_{k+N-1}^*$. The first element of this sequence, u_k^* , is then applied to the system, and the optimization problem is solved again at the next time step.

2.2.1 Stability of MPC systems

While there exist multiple ways to prove stability of MPC systems, this section will only focus on the one which is mostly relevant for this work, namely proving stability by leveraging the characteristics of the terminal set. More specifically, by defining \mathcal{X}_f as convex set which includes the origin ($x(N) = 0$), one can prove stability by insuring certain properties to be satisfied. These properties assume the cost function to have the following form.

$$J(x) = \min_{x,u} V_f(X(N)) + \sum_{t=0}^{N-1} I(x(t), u(t)) \quad (2.12)$$

where $I(x(t), u(t))$ is the stage cost and $V_f(X(N))$ is the terminal cost. Accordingly, these are the properties to satisfy.

1. The stage cost must be strictly positive and zero at the origin.
2. The terminal set is invariant under the local control law $\kappa_f(x)$. Namely

$$x(t+1) = Ax + B\kappa_f(x) \in \mathcal{X}_f \quad \forall x \in \mathcal{X}_f \quad (2.13)$$

given that $X_f \subseteq X$ and $\kappa_f(x) \in \mathcal{U}$

3. Establish stability by illustrating that the terminal cost function serves as a Lyapunov function in \mathcal{X}_f .

$$V_f(x(t+1)) - V_f(x) \leq -I(x, \kappa_f) \quad \forall x \in \mathcal{X}_f \quad (2.14)$$

2.3 Graph Transformation Theory

Graph Transformation Theory (GTT), or graph rewriting, is a mathematical framework used to model and analyze the transformation of graphs. Graphs in this context represent structures composed of nodes and edges, and the transformations involve changing the structure of these graphs.

This section focuses on the fundamentals of this discipline and for more information, the reader is invited to consult the following resources: [21], [47], [27], [43] and [45]. The notions for this section have been collected while studying the aforementioned resources.

In order to introduce GTT, some concepts of category theory are necessary.

Definition 2.10: Category

A category \mathcal{C} consists of the following data:

1. Objects : A group of mathematical objects, denoted as $\text{Obj}(\mathcal{C})$
2. Morphisms (Arrows/Maps): For every pair of objects A and B , there is a set of morphisms denoted as $\text{Hom}_{\mathcal{C}}(A, B)$. If $f \in \text{Hom}_{\mathcal{C}}(A, B)$, we write $f : A \rightarrow B$. Morphisms must satisfy two properties:

- Associativity: For every triple of objects A, B, C and morphisms $f : A \rightarrow B$, $g : B \rightarrow C$, and $h : C \rightarrow D$, the composition $(h \circ g) \circ f$ is the same as $h \circ (g \circ f)$.
 - Identity : For every object A , there exists an identity morphism $1_A : A \rightarrow A$ such that $f \circ 1_A = f$ and $1_B \circ f = f$ for any morphism $f : A \rightarrow B$.
3. Composition: For each pair of morphisms $f : A \rightarrow B$ and $g : B \rightarrow C$, there exists a composite morphism $g \circ f : A \rightarrow C$ in $\text{Hom}_{\mathcal{C}}(A, C)$. Composition must be associative, i.e., $(h \circ g) \circ f = h \circ (g \circ f)$.

Example 2.1: Category of graphs A graph can be represented as the category having

1. Objects : tuple $G = (\mathcal{V}, \mathcal{E}, s, t)$ where
 - \mathcal{V} is the set of nodes
 - \mathcal{E} is the set of edges connecting the nodes
 - $s, t : \mathcal{E} \rightarrow \mathcal{V}$ are the source (target) of the edges, referred to as source (target) function
2. Morphisms: $f : G \rightarrow H$ must respect source and target functions, ie:

$$\begin{aligned}\forall e \in \mathcal{E}. f(s(e)) &= s(f(e)) \\ \forall e \in \mathcal{E}. f(t(e)) &= t(f(e))\end{aligned}$$

Example 2.2: Category of labelled graphs A labelled graph can be represented as the category having

1. Objects: tuple $G = (\mathcal{V}, \mathcal{E}, \mathcal{L}, s, t)$ where
 - \mathcal{V} is the set of nodes
 - \mathcal{E} is the set of edges connecting the nodes
 - \mathcal{L} is the set of labels associated with edges
 - $s, t : \mathcal{E} \rightarrow \mathcal{V}$ are the source and target functions, respectively
2. Morphisms: $f : G \rightarrow H$ must respect source and target functions, i.e.:

$$\begin{aligned}\forall e \in \mathcal{E}. f(s(e)) &= s(f(e)) \\ \forall e \in \mathcal{E}. f(t(e)) &= t(f(e))\end{aligned}$$

Definition 2.11: Product in a Category

Let \mathcal{C} be a category. Let X_1 and X_2 be objects of \mathcal{C} . A product of X_1 and X_2 is an object X , typically denoted $X_1 \times X_2$, equipped with a pair of morphisms $\pi_1 : X \rightarrow X_1$ and $\pi_2 : X \rightarrow X_2$ satisfying the following universal property:

For every object Y and every pair of morphisms $f_1 : Y \rightarrow X_1$ and $f_2 : Y \rightarrow X_2$, there exists a unique morphism $f : Y \rightarrow X_1 \times X_2$ such that the following diagram commutes:

$$\begin{array}{ccccc} & & Y & & \\ & \swarrow f_1 & \downarrow f & \searrow f_2 & \\ X_1 & \xleftarrow{\pi_1} & X_1 \times X_2 & \xrightarrow{\pi_2} & X_2 \end{array}$$

Definition 2.12: Coproduct in a Category

Let \mathcal{C} be a category. Consider objects X_1 and X_2 in \mathcal{C} . A coproduct of X_1 and X_2 is an object X , often denoted as $X_1 \sqcup X_2$, equipped with a pair of morphisms $i_1 : X_1 \rightarrow X_1 \sqcup X_2$ and $i_2 : X_2 \rightarrow X_1 \sqcup X_2$ satisfying the following universal property:

For every object Y and every pair of morphisms $f_1 : X_1 \rightarrow Y$ and $f_2 : X_2 \rightarrow Y$, there exists a unique morphism $f : X_1 \sqcup X_2 \rightarrow Y$ making the following diagram commute:

$$\begin{array}{ccccc} X_1 & \xrightarrow{i_1} & X_1 \sqcup X_2 & \xleftarrow{i_2} & X_2 \\ & \searrow f_1 & \downarrow f & \swarrow f_2 & \\ & & Y & & \end{array}$$

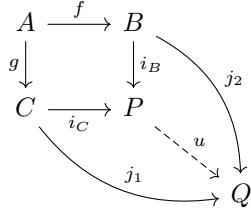
This means that any other morphism $X_1 \sqcup X_2 \rightarrow Y$ that respects i_1 and i_2 factors uniquely through the coproduct $X_1 \sqcup X_2$. In other words, the coproduct of X_1 and X_2 is an object $X_1 \sqcup X_2$ along with morphisms i_1 and i_2 such that, for any other object Y and morphisms f_1 and f_2 , there exists a unique morphism f making the diagram commute. The coproduct captures the idea of combining objects X_1 and X_2 in a way that respects morphisms to another object Y .

Definition 2.13: Pushout

Given a category with three objects A , B , and C , and two morphisms (arrows) $f : A \rightarrow B$ and $g : A \rightarrow C$, the pushout of f and g is an object P along with two morphisms $i_B : B \rightarrow P$ and $i_C : C \rightarrow P$ such that the following diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ g \downarrow & & \downarrow i_B \\ C & \xrightarrow{i_C} & P \end{array}$$

commutes and such that (P, i_1, i_2) is universal. That is, for any other such triple (Q, j_1, j_2) for which the following diagram commutes, there must exist a unique $u : P \rightarrow Q$ also making the diagram commute:



In simpler terms, the pushout is a construction in category theory that combines two morphisms (arrows) $f : A \rightarrow B$ and $g : A \rightarrow C$ in a way that respects the existing structure of the category.

Given two objects B and C with morphisms f and g from a common object A , the pushout P is another object, along with two morphisms $i_B : B \rightarrow P$ and $i_C : C \rightarrow P$. These morphisms create a commutative diagram, meaning that following any path from A to P results in the same composition, regardless of the route taken.

Moreover, the pushout has a universal property, stating that for any other object Q with morphisms j_1 and j_2 forming a similar commutative diagram, there exists a unique morphism $u : P \rightarrow Q$ making the entire diagram commute.

In simpler terms, the pushout captures a way of amalgamating information from B and C in a manner that is compatible with the given morphisms, and it does so in a universal way that any alternative attempt to amalgamate would factor uniquely through the pushout.

Definition 2.14: The Double Pushout Approach (DPO)

In the DPO, productions have the following form:

$$p : L \xleftarrow{l} K \xrightarrow{r} R$$

where:

- L (Left-hand side (LHS)) is a graph representing the initial pattern or structure to be recognized in the larger graph. This is the part of the graph that the production rule aims to match and replace.
- K (Interface graph) represents the region where the left-hand side L is embedded in the larger graph.
- R (Right-hand side (RHS)) is a graph representing the replacement or transformation that will be applied to the matched portion of the graph. After the application of the production rule, the matched portion of the graph (given by L and K) will be replaced by the structure defined in R .
- $l : K \rightarrow L$ is a morphism representing the embedding of the interface graph K into the left-hand side L .
- $r : K \rightarrow R$ is a morphism representing the embedding of the interface graph K into the right-hand side R .

The morphisms are usually injective, indicating that each element in the interface graph K has a unique counterpart in L and R , respectively. The overall interpretation of $p : L \leftarrow K \rightarrow R$ is that when the left-hand side L and its context K are found in a larger graph, they can be replaced by the right-hand side R according to the specified morphisms l and r .

This production is applied as follows.

$$\begin{array}{ccccc} L & \xleftarrow{l} & K & \xrightarrow{r} & R \\ \downarrow m & & \downarrow d & & \downarrow m' \\ G & \xleftarrow{l'} & D & \xrightarrow{r'} & H \end{array}$$

So, when this production rule is applied to G , the subgraph D matching the left-hand side L is replaced by the right-hand side R , resulting in the transformed graph H . The morphisms m , d , and m' ensure that the embedding and replacement are consistent with the larger graphs. The commutativity of the diagram ensures that the rewriting process is well-defined and respects the structure of the graphs involved.

1. Matching and Interface: The interface graph K is matched within the original graph G using the matching morphism $d : K \rightarrow D$. This identifies where the LHS L is located in G .
2. LHS Embedding: The LHS L is embedded into the original graph G using the embedding morphism $m : L \rightarrow G$. This creates a graph that includes the LHS structure.
3. Application of Production Rule: The LHS L is then replaced by the RHS R , respecting the embedding morphisms l and r . This transformation is performed within the identified context given by the interface graph.
4. Resulting Graph: The resulting graph H is obtained after the replacement. The context graph D is embedded into H using $r' : D \rightarrow H$, and the RHS R is included in H using $m' : R \rightarrow H$.

Example 2.3: Let's define a graph that represents a simple directed network:

$$G = \{\text{Nodes: } A, B, C, D; \text{ Edges: } (A \rightarrow B), (B \rightarrow C), (C \rightarrow D)\} \quad (2.15)$$

Let's define a DPO production rule to replace a chain of three nodes with a new node. The production rule is as follows:

$$p : L \leftarrow K \rightarrow R \quad (2.16)$$

Where:

- L (LHS): Three consecutive nodes forming a chain - $A \rightarrow B \rightarrow C$.

- K (Interface graph): The interface graph is the same chain of nodes - $A \rightarrow B \rightarrow C$.
- R (RHS): A single node - X .
- $l : K \rightarrow L$ is the identity morphism since L and K are the same chain.
- $r : K \rightarrow R$ maps the chain of nodes to the single replacement node.

The production rule essentially says that if a chain of three nodes is found in a graph, it can be replaced with a single node. The interface graph K ensures that the replacement occurs in the correct context.

$$\begin{aligned} \text{Initial Graph: } & G = \{A \rightarrow B \rightarrow C \rightarrow D\} \\ \text{Production Rule: } & p : \underbrace{A \rightarrow B \rightarrow C}_{L} \rightarrow \underbrace{A \rightarrow B \rightarrow C}_{K} \rightarrow \underbrace{X}_{R} \\ \text{Resulting Graph: } & H = \{X \rightarrow D\} \end{aligned}$$

After applying the production rule, the chain $A \rightarrow B \rightarrow C$ has been replaced by a single node X , resulting in the transformed graph H .

Definition 2.15: Dangling Condition

Let $G = (\mathcal{V}, \mathcal{E})$ be a graph with vertices \mathcal{V} and edges \mathcal{E} . The dangling condition imposes that, for each edge $e_i = (v_s, v_t) \in \mathcal{E}$, if e_i is not deleted, then both its source node v_s and target node v_t should be preserved. This condition is an essential condition in the DPO framework.

Definition 2.16: Identification Condition

Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ be a graph and $H = (\mathcal{V}_H, \mathcal{E}_H)$ be a host graph. Suppose there exists a morphism $f : \mathcal{V}_G \rightarrow \mathcal{V}_H$ and $g : \mathcal{E}_G \rightarrow \mathcal{E}_H$ such that two nodes or edges in G , denoted as $v_1, v_2 \in \mathcal{V}_G$ or $e_1, e_2 \in \mathcal{E}_G$, are matched into a single node or edge in H (i.e., $f(v_1) = f(v_2)$ or $g(e_1) = g(e_2)$ via a non-injective morphism), the identification condition imposes that those should be preserved. This condition is an essential condition in the DPO framework.

Definition 2.17: Negative Application Condition

Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ be a graph and $H = (\mathcal{V}_H, \mathcal{E}_H)$ be a host graph. Suppose there exists a morphism $f : \mathcal{V}_G \rightarrow \mathcal{V}_H$ and $g : \mathcal{E}_G \rightarrow \mathcal{E}_H$ such that two nodes or edges in G , denoted as $v_1, v_2 \in \mathcal{V}_G$ or $e_1, e_2 \in \mathcal{E}_G$, are matched into a single node or edge in H (i.e., $f(v_1) = f(v_2)$ or $g(e_1) = g(e_2)$ via a non-injective morphism), the negative application condition (NAC) specifies conditions under which a transformation rule should not be applied. If the graph G contains a subgraph that satisfies the NAC condition, the transformation rule should not be applied to prevent the introduction of invalid configurations into the host graph H .

Chapter 3

Literature Review

This chapter will focus on an exploration of various pieces of literature and related works that have been thoroughly reviewed, aligning with the overarching purpose of this study.

This chapter has been divided into multiple subsections, which follow the structure of this work. Section 3.1 explores various mathematical models and challenges inherent in designing efficient and adaptable transportation systems. Subsequently, it delves into the intricacies of modeling battery systems. Section 3.2 deals with various challenges that are common within ATS, such as dispatching, routing, and rebalancing. Finally, Section 3.3 discusses the applications of Graph Transformation Systems (GTS) in engineering design exploration and traffic management, highlighting their versatility in addressing complex system dynamics.

3.1 Fleet and Transportation Network

Most of the literature reviewed during the development of this study focused on (Autonomous) Mobility-on-Demand (AMoD), therefore only considering human mobility. In this case, *Frazzoli et al.* in [62] identify three main mathematical models used to represent transportation networks: continuum, queuing theory and graph-theoretic models. This theses focuses on the latter.

In graph-theoretic models, the transportation network is modeled as a directed graph $G(V, R)$, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. Usually, each node $v \in V$ represents a location, or a station. In [5], in a more abstract way, nodes are used to represent areas of interest. Each edge $\langle v_1, v_2 \rangle \in E$ represents a connection, which could be a road or a collection of those, from v_1 to v_2 . Furthermore, each link is usually associated with a metric, such as distance $d : E \rightarrow R_{\geq 0}$. Many studies adopt a dynamic fluid approach, representing AVs and customers not as individual entities but rather as flows moving between nodes. In this framework, models that do not vary with time often simplify to static network flow problems. Alternatively, graph-theoretic

models can be integrated with vehicle-centric models, where both AVs and customers are modeled individually. This chapter delves into the exploration of various sources and solutions that have been examined in the development of the battery model utilized in this study. As the first work considered, *Montoya et al.* in [34] employed linear piece-wise approximations to approximate the non-linear charging function, reporting an error of approximately 1%. Additionally, they introduced the concept of different types of charging stations. *Froger et al.* in [13] and *Kancharla et al.* in [26] also employed piece-wise linear functions for their respective charging regimes. Conversely, *Nie et al.* in [39] did not consider the possibilities for vehicle recharging in their model.

In terms of modeling the battery charging profile, diverse approaches have been proposed. *Han et al.* in [20] developed a model based on the internal resistance and voltage of the battery; however, this model was disregarded in favor of others that better captured the battery charging profile based on more relevant factors. *Yu et al.* in [60] proposed a sophisticated model for charging constraints in general. Although their approach differs from the one presented in this work, as it incorporates charging during routing, this aspect was excluded in favor of an alternative approach. *Lee et al.* in [30] considered a simplified battery model that provides states of charge (SoC) as a function of the charging current. The author asserted the general applicability of their approach to every charging profile, provided $\text{SoC}(t)$ is a concave and non-decreasing function with $\text{SoC}(0) = 0$, and an inverse function $\text{SoC}^{-1}(a)$ exists for $0 < a < Q$.

3.2 ATS Challenges

AV Dispatching

For the vehicle dispatching problem, multiple solutions have been already proposed. For example, *Vasco et al.* in [54] model it as a linear minimum cost multicommodity flow problem; *Holzapfel et al.* in [23] model it as a MIP problem. Others, on the other hand, make use of other heuristics. For example, *Levin et al.* in [31] and *Mora et al.* in [2] use nearest neighbors. Recently, to improve real-time capabilities, some approaches have been proposed which make use of deep learning, like for e.g. the one proposed by *Yu et al.* in [61].

AV Routing

On a basic level, the routing problem can be reconducted to the Vehicle Routing Problem (VRP) (a thorough analysis and explanation can be found in [53]). Usually, the VRP is solved and analyzed as a static problem, which implies that requests are known in advance. As a result, origins and destination of each trip is also known a priori. As pointed out by *Frazzoli et al.* in [62], in ATSs requests are dynamic, meaning they are not known in advance. More specifically, the

task of managing an ATS can be viewed as a specific case of the dynamic one-to-one pickup and delivery problem. As pointed out by *Zhang* in [64], to provide a more detailed characterization of these systems, several additional attributes and constraints must be considered:

- Immediate Service Requests: Requests are made for immediate service rather than being scheduled in advance with specified time windows.
- Stochastic Customer Arrivals and Travel Times: The timing of customer requests and the duration of travel are subject to stochastic variability.
- Multi-Occupancy Vehicles: Contrary to [64], vehicles must be considered to have a capacity, rather than being single-occupancy. This is motivated by the fact that recently developed AV are equipped with more than one seat (see for e.g. [15])

Wollenstein-Betech et al. in [58] propose a solution to the routing-rebalancing problem in mixed traffic situations. They propose an interesting two-graph solutions, i.e. one graph for the ATS and one for the pedestrians which are interconnected. Furthermore, they also consider private and public vehicles. *Pavone et al.* in [42] also propose a solution for the rebalancing problem. Although they mainly consider MoD systems, their approach can be easily applied to ATSs. Their solution, moreover, focuses only on the rebalancing issue, which is studied using a steady-state fluid model, as pointed out in [58]. Furthermore, this work does not take into consideration congestions in their model.

AV Rebalancing

In the context of network flow models, rebalancing is usually formulated as a (integer) optimization problems, such as in the works of *Zgraggen et al.* in [63], *Carron et al.* in [8] and of *Smith et al.* in [51]. Crucially, in real-time testing, the rebalancing problem is solved leveraging the model-predictive control framework, such as in the works mentioned above.

Wollenstein-Betech et al. in [59] investigate effective pricing and rebalancing strategies for AMoD systems from a macroscopic planning standpoint. It aims to optimize profits while ensuring system load balance. Using a dynamic fluid model, the study demonstrates equilibrium attainment via pricing policies and develops an optimization framework for determining optimal pricing and rebalancing approaches. They model customers and vehicles located in regions using a queuing approach, i.e. two queues per region representing vehicles and clients. *Salazar et al.* in [48] study the rebalancing problem while considering two important aspects, namely mixed traffic and congestions. The first is treated by modeling the road network and the walking network as two separate entities, which combined make up a supernetwork defining the entire transportation network. The latter are treated according to the Bureau of Public Roads (BPR) model ([40]), more specifically by approximating their model linearly. The BPR

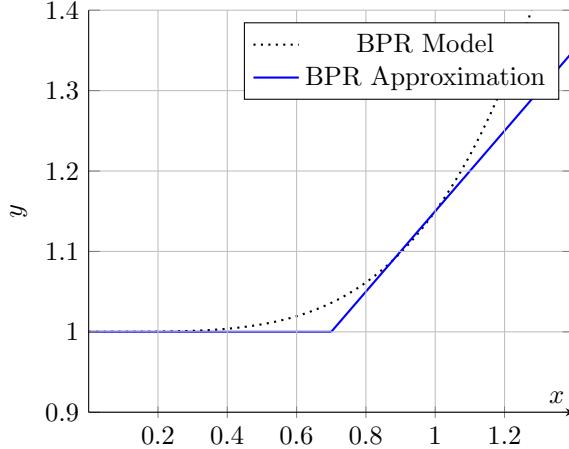


Figure 3.1: BPR Model and its Approximation as a piecewise linear function. In this example, $x_{th} = 0.7, x_{max} = 1.4, a = 1, b = 0.5$

model of congestions can be expressed by (3.1) and is visualized in Fig 3.1 as a black dotted line.

$$f_{BPR}(x) = 1 + 0.15x^4 \quad (3.1)$$

Since this model would lead to an unconvex, i.e. intractable, problem, the authors propose a piece wise approximation of it, which take the form expressed in (3.2)

$$y = \begin{cases} a & \text{if } x \in [0, x_{th}] \\ a + b \cdot (x - x_{th}) & \text{if } x \in [x_{th}, x_{max}] \end{cases} \quad (3.2)$$

where a and b are used to model the height of the horizontal line and the slope of the second line, x_{th} is the threshold of the piecewise approximation and x_{max} defines the approximation window. This approximation can be seen in Fig 3.1 as a blue line.

(3.2) is then used to calculate the traverse time for each edge. While this is a more representative congestion model than the one proposed in this paper in Section 4.1, one might argue that the two serve two different purposes. While the one proposed in [48] serves to express the travel time in terms of the amount of vehicles currently traveling on the node, the one proposed in this work is used to simply limit the amount of vehicles traveling on the link to avoid situations of stop-and-go traffic. Furthermore, it is worth mentioning that this model can be easily integrated into the model by modifying the definition of the travel time function.

In the abovementioned study, the energy consumption of the AVs is also modeled differently compared to this work. Within their network graph G , the energy consumption of each AV with efficiency η traveling through arc (i, j)

with distance d_{ij} is expressed as:

$$e_{ij} = \left(\frac{\rho_a}{2} \cdot A_f \cdot c_d \cdot v_{ij} + c_r \cdot m_v \cdot g \right) \cdot \frac{d_{ij}}{\eta_{AV}} \quad \forall (i, j) \in G \quad (3.3)$$

where the aerodynamic drag is determined by the air density ρ_a , the frontal area A_f , the drag coefficient c_d and the moving velocity v_{ij} . The friction of the wheels on the road is determined by the rolling friction coefficient c_r , the mass of the vehicle m_v , and the gravitational acceleration g .

This is a more vehicle-based approach compared to the one adopted in this work, which only considers the battery dischargment, as it takes into consideration various other aspects such as drag and rolling friction. For the purpose of this work, however, it has been chosen to neglect those factors, which could be later introduced into the battery-dischargment model, to favour a more simplistic approach. Furthermore, given that in the aforementioned paper is assumed to have constant velocity $v_{ij} = \frac{d_{ij}}{t_{ij}}$, one can trivially adopt a similar solution for the model in this paper by making the same assumption.

Wallar et al. in [57] tackle the rebalancing problem independently from any other ATS challenge, while also considering ride-sharing possibilities. The main idea of the proposed approach is to assign free vehicles to regions, which are computed offline, according to the estimations of traveling request per region. In this case, the requests are estimated using the particle filter. Furthermore, the division in region for the graph is formulated as an integer linear programming problem using a rechebability matrix R , which indicates whether a station j is recheable from i within a maximum time t_{max} ($R_{ij} = 1$ else $R_{ij} = 0$). Cleverly, the authors also took into consideration the fact that, although a vehicle is free, it requires some time to reach the assigned rebalancing region. In other words, if a vehicle requires eight minutes to reach a certain region, considering a time horizon of ten minutes, it will be available only for two minutes, i.e. 20% of the time. This is used to limit the oversaturation of vehicles in the respective rebalancing regions.

3.3 Applications of Graph Transformation Systems

Due to the flexible nature of graph and their intuitive understanding, Graph transformation systems (GTS) are widely use for design exploration. *Voss et al.* in [56] propose a study whre they demonstrate the flexibility of this method in various engineering disciplines. For example, *Zhao et al.* in [65] utilize this approach to explore the design space of a terrain-optimized robot. In the context of mechatronic systems, GTS can be used for the development of components, such as the gear box design proposed in *Fürst et al.* in [14], and of entire systems, as demonstrated by *Gross et al.* in their series of three papers for the development of a satellite ([16], [17], [18]).

In the context of road and traffic management, *Beck et al.* in [4] propose a method to optimize the performance of vehicle routing over a graph netwrk based on GTS. While graph transformations are not used to find a direct solution to the VRP, those are used to transform the graph in such a way that the routing algorithms' performance is optimized.

Raadsen et al. in [43]introduce a category theoretical approach in formalizing transport network transformations by adopting and adapting pattern graph transformation techniques. This study introduces two novel spatial aggregation methods applicable to a class of traffic assignment models, employing a category theoretical approach. The formalization technique, originally developed for quantum physical processes, offers an intuitive graphical representation with a rigorous mathematical foundation. The method shares similarities with regular expressions and functional programming, providing insights for constructing solvers or algorithms. The proposed aggregation methods are compatible with traffic assignment procedures, decomposing the network into a constant free-flowing part and a smaller demand-varying delay part, aiming to reduce computational costs without sacrificing accuracy, as demonstrated in a large-scale case study.

Furthermore, another GTS application in the context of traffic management can be found in *Lara et al.* in [10]. This study introduces a domain-specific visual language, named Traffic, designed for the domain of traffic networks. The syntax of the language is formally defined using meta-modelling. For semantics, the study employs two approaches: the first involves graph transformation for operational semantics, while the second incorporates timing information and establishes a denotational semantics using Timed Transition Petri Nets (TTPN). The transformation from the Traffic formalism to TTPN is defined through graph transformation. Both semantic approaches are implemented and analyzed using the AToM tool, showcasing their applicability through examples.

Chapter 4

Time-Invariant Model of ATS

This chapter focuses on tackling the main problems for an ATS. First, a time-invariant vehicle-centric model for the transportation network is developed in Section 4.1. The model presented in this document expands upon the one formulated in [5], using graph theory and a vehicle-centric approach to provide flexibility in handling different vehicle capacities and types. This model will be the basis upon which sub-problems of an ATS will be formulated and analyzed in Section 4.2. With those insights, the Complete ATS Managing problem will be formulated in Section 4.3 and its performance evaluated in Section 4.4.

4.1 Vehicle-centric Model

The model provided in this section will expand the model formulated in [5]. It is worth noting that, although the model presented in [5] is specific for the scenario described in the same work, it can be expanded to cover more general use cases. Therefore, the model of the transportation network will be developed using graph theory. While much of the existing literature relies on fluid-dynamic models ([62]), wherein customers and AVs are depicted as continuous flows between nodes, this study will employ a combination with a vehicle-centric approach. In the latter, customers and AVs will be individually modeled. Contrary to the solutions proposed by most of the literature, this work combines mobility of people with goods transportation, therefore modeling vehicles individually will provide the required flexibility to handle different capacities and the increased complexity given by the plethora and variety of vehicles, the set of which will be indicated as \mathcal{A} .

Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ being the directed graph representing corresponding transportation network of a city, where \mathcal{V} is the set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Any vertex, or node, $v \in \mathcal{V}$ represents a location. For the context of this paper, the two terms will be used interchangeably, as they ultimately

indicate the same thing. Following the work in [5], each node consists of an area of interests. An edge $\langle v_i, v_j \rangle \in \mathcal{E}$ represents a connection, which can consist of a road or a combination of those, linking v_i to v_j . The level of abstraction is decided by the engineer and can be used to regulate the granularity of the model. Locations v_i and v_j can indicate low level key points, such as cross roads or trafficked traffic lights, as well as higher level areas such as residential areas or leisure centers. Similarly, an edge can contain multiple crossroads or joints or simply indicating a path from v_i to v_j , regardless of additional details. A correct granularity is not trivial to decide a priori and a general approach is likewise hard to delineate. Determining the level of abstraction remains, therefore, an engineer's prerogative and highly depends on the system in question. It is also worth noting that the, for conveniency and to better reflect the situation in real world applications, the graph is a uni-direct graph. To model a two-way road, one should simply make use of two edges from locations v_i and v_j .

As also described in [5], each edge is associated with multiple metrics and information.

Firstly, at each edge one must attribute a travel time T . T is a function $T : \mathcal{E} \times \mathcal{A} \rightarrow \mathbb{R}_{>0}$, which at each given a vehicle and edge maps a float value $T_{i,j}^a \in \mathbb{R}_{\geq 0}$ indicating the time required for the vehicle of type a to travel the path from v_i to v_j . In this work, the unit of $T_{i,j}^a$ is not of particular interest, as it depends on the application.

Secondly, each edge is also associated with a distance $d : \mathcal{E} \rightarrow \mathbb{R}_{\geq 0}$, which maps an edge to a float value $d_{i,j}$ indicating the distance from v_i to v_j .

Another factor which is considered in this work, as stated above, is pollution. This metric is associated to each edge and is a function $f : \mathcal{E} \times \mathcal{A} \rightarrow \mathbb{R}_{\geq 0}$. To simplify the discussion for this work, f is assumed to produce a certain value referred to as pollution index. In other words, it is not supposed to represent a measurable element, such as CO₂ emissions, but rather a value which can represent multiple quantitative factors and is higher if the combination of path and vehicle type is highly polluting. While it can be argued that the pollution can also be simply a function of d and T , this abstraction does not account for other elements, such as road type, vehicle fuel and road slope. The index f , therefore, is a mathematical abstraction which allows a more flexible model for the system. To use this abstraction, however, each edge must include additional information, including the one mentioned before.

Similarly to what observed by *Zhang et al.* in [46], we can introduce the concept of congestions in the model by constraining the routing by the capacity of each road. In other words, we can associate each edge with a capacity $c : \mathcal{E} \rightarrow \mathbb{N}_{>0}$ indicating the limit in terms of car occupancy above which the traffic in that edge slows down eventually reaching a congestion. As pointed out in [46], this simplified model is adequate in this context as well as the aim of this work focusing solely on the control of vehicles to prevent congestion rather than analyzing their behavior in congested network. While it is understood that congestions behave differently in real world scenarios and other, more sophisticated approaches exist ([32], [55]), for the sake of simplicity, we will assume $T_{ija} = \infty$ if the number of vehicles in that edge from $\langle v_i, v_j \rangle$ to be larger than the capacity c_{ij} .

In addition, each edge $\langle v_i, v_j \rangle$ will also include information regarding traffic limitations, i.e. whether a vehicle type is allowed or not to travel that link. Intuitively, limitations will be represented as a function $s : \mathcal{E} \times \mathcal{A} \rightarrow \{0, 1\}$, where 1 implies the vehicle can traverse that edge. Following the example in [5], limitations can be because of various factors, like for e.g. weight or height. For this reason, it is convenient to abstract away such details and just indicate whether a link is traversable or not. Notably, one could use this representation to transform a bidirect graph G' to be equivalent to G by letting $s(e, a) = 0$ for any one-way road $e = \langle v_i, v_j \rangle$ for all $a \in \mathcal{A}$. For convenience, this function will be incorporated in the definition of the capacity, updating it to become (4.1).

$$c(e = \langle v_i, v_j \rangle) = \begin{cases} 0 & \text{if } s(e) = 0 \\ c(e) & \text{otherwise} \end{cases} \quad (4.1)$$

Following the discussion above, this is equivalent to setting $T_{ija} = \infty$ for all $a \in \mathcal{A}$ according to this model specifications. In this way, we can treat the road as being unaccessible without increasing the number of conditions and decrease the readability of the model.

Each vehicle $a \in \mathcal{A}$ is modeled as a tuple $\langle \underline{s}_a, \bar{t}_a, B_a(t), \mathcal{R}_a, \mathcal{T}_a, P_a, G_a, C_a, F_a \rangle$ where $\underline{s}_a \in \mathcal{V}$ and $\bar{t}_a \in \mathcal{V}$ are the starting and terminal node respectively; $B_a(t) \in \mathbb{R}_{\geq 0}$ is the state of charge at time t ; \mathcal{R}_a is the set of requests assigned to vehicle a ; $G_a \in \mathbb{R}_{\geq 0}$ and $P_a \in \mathbb{R}_{\geq 0}$ are the goods and people capacity and $C_a \in \mathbb{R}_{>0}$ and $F_a \in \mathbb{R}_{>0}$ indicate operational cost and pollution factor. Furthermore, each set of assigned requests \mathcal{R}_a is a subset of the set of all requests in the systems, i.e. $\mathcal{R}_a \subseteq \mathcal{R}$. For the purpose of this model, we will assume each request assignment to be unique, i.e. $\mathcal{R}_a \neq \mathcal{R}_b$ and $\mathcal{R}_a \subseteq \mathcal{R} \setminus \mathcal{R}_b$, for all $(a, b) \in \mathcal{A}$ with $a \neq b$.

Each vehicle's battery is modeled as a tuple $\mathcal{T}_a = \langle Q_a, I_a^b, R_a^-, R_a^+, \theta_a \rangle$, where $Q_a \in \mathbb{R}_{>0}$ will be used to indicate battery capacity; $R_a^+ \in \mathbb{R}_{>0}$ and $R_a^- \in \mathbb{R}_{>0}$ are the charging rate and discharging rate, respectively; and $\theta_a \in [0, 1]$ is used to model the battery breakpoint. These will be modeled according to the two operating mode, i.e. charging and discharging. While it is understood that these two operations are highly influenced by multiple factors, it is sensible to make assumptions in order to simplify the model. As it is also widely spread in industry, one can neglect the influence of external factors such as weather condition or intrinsic characteristics of the battery, such as temperature or age. The charging profile will be modelled taking inspiration from the model proposed by Lee *et al.* in [30]. As mentioned above, the vehicles have a charging rate, a battery capacity and a breakpoint, namely R_a^+ , Q_a and θ_a respectively. The state of charge at time $t \in \mathbb{R}_{\geq 0}$ of a certain vehicle will be obtained according to the model described in (4.2), which is derived from the CC-CV (Constant current - Constant Voltage) scheme. (A more complete explanation can be found in [33]).

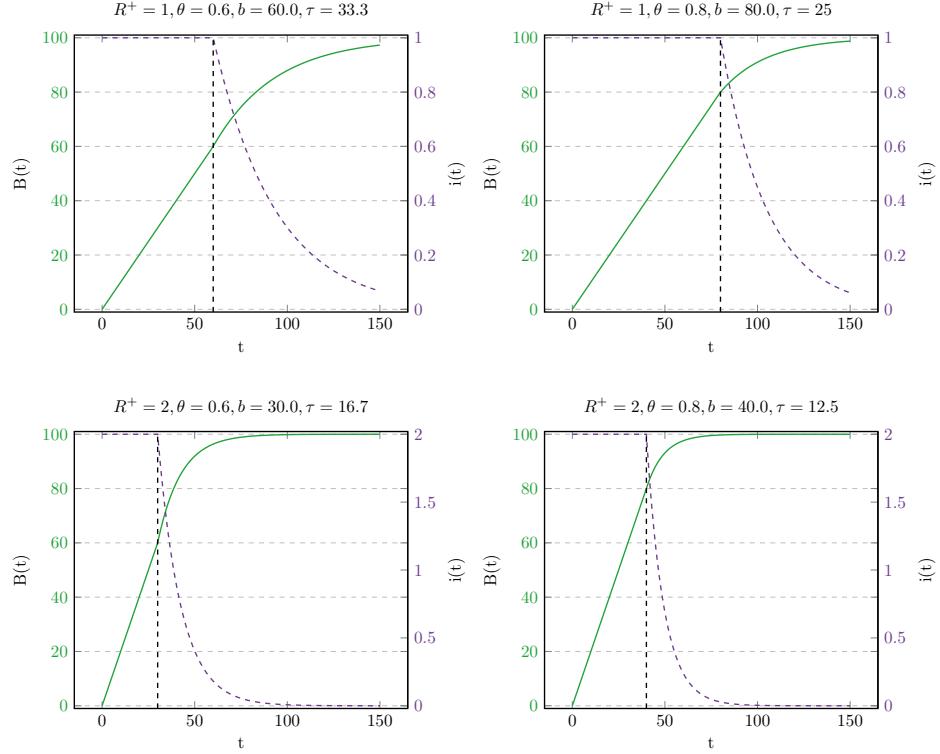


Figure 4.1: Different charging profile obtained according to the model developed in (4.2). B is the State of Charge in Percentage, while $i(t)$ in function of the time unit t . The battery have $Q = 100$

$$B_a(t+1) = \begin{cases} B(t)_a + R_a^+ & \text{if } t \leq b_a \\ Q_a - \frac{Q_a i(t)(1 - \theta_a)}{R_a^+} & \end{cases} \quad (4.2)$$

with $b_a = \frac{\theta_a Q_a}{R_a^+}$, $i(t) = \begin{cases} R_a^+ & \text{if } t \leq b_a \\ R_a^+ e^{-(t-b_a)/\tau} & \end{cases}$ and $\tau = \frac{C_n}{R_a^+ \theta_a} \in \mathbb{R}_{\geq 0}$ representing how quickly the current decreases. The value of C_n depends on the charging device considered.

An example of the charging profiles from this model can be seen in Fig 4.1. Since $i(t)$ is assumed to be constant at time $t \leq b_a$, i.e. $i(t) = R_a^+$, the first

condition is derived as follows.

$$\begin{aligned}
B_a(t + \alpha) &= B_a(t) + \int_t^{t+\alpha} R_a^+ dt \\
B_a(t + \alpha) &= B_a(t) + R_a^+ \int_t^{t+\alpha} dt \\
B_a(t + \alpha) &= B_a(t) + R_a^+ \Big|_t^{t+\alpha} \\
B_a(t + \alpha) &= B_a(t) + R_a^+(t + \alpha - t) \\
B_a(t + \alpha) &= B_a(t) + \alpha R_a^+
\end{aligned}$$

with $\alpha = 1$.

The discharging, on the other hand, will be modeled as it being directly proportional to the travel time T , as described in (4.3).

$$B_a(t + T_{u,v}^a) = B(t)_a - R_a^- T_{u,v}^a \quad (4.3)$$

The equation describes the relationship between the initial and final battery levels on edge $\langle v_u, v_v \rangle$ during the transit. Following (4.3), one can assign to each edge a rate of battery discharge by defining it as the difference between the state of charge at time t and at time $t + T_{u,v}^a$.

$$e_{u,v}^a = B(t)_a - B_a(t + T_{u,v}^a) = R_a^- T_{u,v}^a \quad (4.4)$$

(4.4) allows to describe the discharging rate as a function of only the edge and the vehicle.

In terms of operational costs, multiple factors should be considered and the analysis must be extended from the one developed in [5], where the operational cost was only in function of the vehicle type. Similarly to the aforementioned work, this model assumes a certain operational cost depending on the vehicle type, however the discussion is also extended in terms of vehicle charging cost. In other words, we can decouple it from the general concept of the operational cost per vehicle and considering in function of the charging or discharging profiles described above. Furthermore, it also makes sense to assign the cost also depending on the distance traveled, i.e. the value d associated to the edges.

Compared to [5], in this work, the model also extends the information associated to the nodes. Nodes will be of two categories, namely normal nodes and charging (or depot) stations, whose sets are denoted as \mathcal{V}_n and \mathcal{V}_c , respectively. We can, therefore, conclude that $\mathcal{V} = \mathcal{V}_n \cup \mathcal{V}_c$. The information associated to the nodes depend from the category each node belongs to. Nodes belonging to \mathcal{V}_n do not possess any specific information, as they are not of high importance for this work. Charging nodes \mathcal{V}_c , on the other hand, possess two main characteristics, namely their capacity and their charging ability. Intuitively, the term capacity is used to refer to the number of available parking spots, identified as z_v .

Furthermore, three types of charging ability will be considered, i.e. fast, normal and slow charging, which will be based on the model in (4.2). The choice of parameters is explained in Table 4.1.

Type	C_n	τ	θ_a	R_a^+
Slow	20	33.3	1	0.6
Medium	15	18.17	1	0.8
Fast	10	6.25	2	0.8

Table 4.1: Choice of parameters for the charging stations according to the type. For the charging profile, please refer to Fig 4.1. The first column is the constant related to the charging type, the second is the $\tau = \frac{C_n}{R_a^+ \theta_a} \in \mathbb{R}_{\geq 0}$ calculated with the values on the third and fourth column.

It is a sensible choice to limit the domain of the starting and terminal node of each vehicle to the sets of charging and depot nodes, i.e. $s_a \in \mathcal{V}_c \cup \mathcal{V}_d$ and $\bar{t}_a \in \mathcal{V}_c \cup \mathcal{V}_d$. In other words, this choice implies that each vehicle's destination will either be the depot at the end of the shift, for example, or a charging node, where its battery can be charged to accommodate future requests.

Requests are modeled as tuples $\langle \underline{s}', \bar{t}', G', P', \lambda, a', b' \rangle$, where $\underline{s}' \in \mathcal{V}_n$, $\bar{t}' \in \mathcal{V}_n$ represent the pickup and delivery point respectively; $G' \in \mathbb{R}_{\geq 0}$ ($P' \in \mathbb{R}_{\geq 0}$) refers to the amount of goods (people) required to transport, and $\lambda \in \mathbb{R}_{>0}$ is the rate of requests, in customers per unit time, which, therefore, makes the requests stationary and deterministic. Additionally, requests must be delivered within a time window within $[a', b']$.

4.1.1 Model Evaluation

Some comments are in order. The model in Section 4.1 is time-invariant. According to the definition presented by *Frazzoli et al.* in [62], time invariance in the context of transportation modeling refers to the assumption that the number of requests remain constant over time allowing for the simplification of temporal dynamics and treating specific time intervals as homogeneous units. This modeling concept is employed when the rate of change in transportation demands is deemed slow compared to the average travel time of individual trips, often observed in stable urban environments ([36]). While this model is indeed applied in a relatively dense urban environment, some integrations are required in order to adapt it to time-varying scenarios. Moreover, customer requests are also assumed to be known. This requirement can be fulfilled in practice with requests made in advance or some techniques to estimate requests throughout the day. It is important to note that request estimation might lead to suboptimal performance.

Secondly, as already mentioned in the previous section, the model used to de-

scribe congestions is indeed rather simple and more complex formulation might better capture the phenomena. However, the semplification is considered powerful enough for the purpose of this work and, as pointed out by *Zhang et al.* in [46] as well, more sophisticated models can be used offline using simulation techniques to derive the capacity metric used in this model.

Some comments are in order also regarding vehicles. Firstly, the vehicles are autonomous and fault proof. It is outside of the scope of this work to deal malfunctioning vehicles or exceptional situations outside of the normale functioning regime of the system. Additionally, the vehicles are assumed to be fully electric. This assumption is widely used in literature and it is motivated, among other aspects, also by the recent trends in industry to transition towards electrical mobility. Furthermore, the model provided for the battery charging and discharging profiles is rather simplistics and more sophisticated approaches exist in literature (see Chapter 3 for more details). However, these models should be seen an addition to the corrent approach and, albeit with some potential modifications, it is plausible they can be integrated in the system model. While in [5] vehicles have been designed to be capable of transporting a potentially large number of people, some considerations should be made in this regard. While it is sensitive to consider vehicles capable of transporting up to 50 people in terms of environmental, economical and overall transportation efficiency and while it is also expected that the model designed would work with those vehicles as well, in terms of practical use, it might be more appropriate to make use of those vehicles in different ways. Notably, it should be studied whether it is efficient to tailor the routes of those vehicles according to passenger needs. Motivated also by the fact that passengers might have common stops, it might be more sensible to treat those vehicles as buses are treated nowadays, i.e. with a pre-determined route among stops which are placed according to the most common stops, such as hospitals or train stations. Similarly, if large vehicles to transport goods are to be considered, such as large trucks, those are usually not used for home delivery, but rather used to transport goods among specialized centers. It must be noted that both situations do not invalidate this work, or any previous work. It is clear that the system described in this work in combination to the truck for goods transportation. Regarding people mobility, while buses are indeed an already established and efficient transportation mean, this systems aims at filling up the situations where a bus system is indeed lacking, such as transportation of people with special needs or, in general, more tailored to the specific needs of potential customers.

In this model, some assumptions on the nodes have been made. Firstly, there is no distinction between depot and charging nodes. The difference between charging and depot stations lays on the fact that charging nodes are not meant to host vehicles for long periods of time, contrary to depots. The difference can be envisioned as if charging stations were gas stations and depots were bus depots. The model can be trivially extended according to this distinction. Additionally, it is assumed that the charging stations are all of the same types. It might be argued that some stations might have different types of chargers. This characteristics can be reflected by the model simply by “splitting” them.

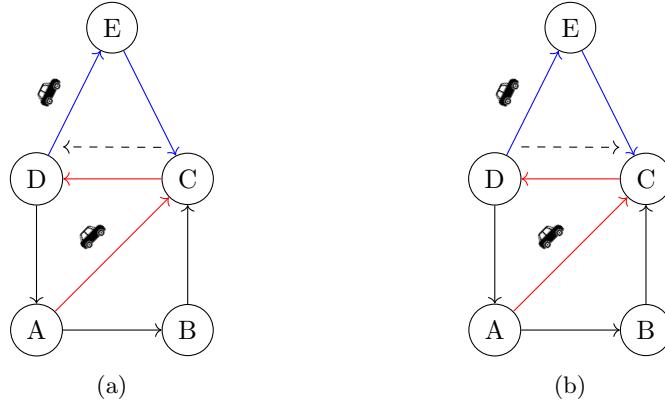


Figure 4.2: Fig 4.2a and Fig 4.2b show a simplified example of a sensible request assignment. In red and blue are the paths the two AVs can traverse, while the dashed blue two different requests (In Fig 4.2a the customer asks to go from C to D, while in Fig 4.2b the customer asks to go from D to C). In the case of Fig 4.2a, it is more sensible to assign the request to the red AV, while in the case of Fig 4.2b, the blue AV is a better choice.

A station having, for instance, two types of chargers can be represented by two equivalent nodes in the graph having different category. Moreover, the variety of chargers in Table 4.1 can also be easily extended.

4.2 Analysing ATS challenges

Before defining the problems for the different challenges faced by the ATS considered in this work, it is essential to lay down the motivations behind certain assumptions made during the conceptual phase. Following the previously described model, it is assumed that each vehicle starts and ends at a charging or depot node, s_a and t_a , which do not have to be necessarily the same. This choice has been made to reduce the number of vehicles parked on the street, since only limited locations would have parked vehicles. Furthermore, if a vehicle leaves from a depot at the beginning of the shift, it is assumed to have a fully charged battery, i.e. $B_a(0) = 100$. Additionally, since normal nodes do not possess charging capabilities, vehicles can be charged only after, or before, serving a request.

4.2.1 Dispatching

Informally, the dispatching problem can be defined as the task of assigning requests to the most suitable vehicle. There exist already multiple solutions proposed for the problem and the reader is encouraged to refer to Chapter 3 for a more thorough analysis.

Dispatching is critical for the overall system performance and must be done in a way that can further facilitate the next steps. To improve performance, dispatching can not be decoupled and solved as a stand-alone problem. For example, Fig 4.2 shows a simplified situation where a sensible dispatching, which depends on a posterior step, will improve system performance. Nevertheless, during the dispatching problem, some additional elements must be considered as well.

Since the model proposed in Section 4.1 is a vehicle centric model, we can model the dispatching problem with the help of a binary variable x_{ar} defined in (4.2.1).

$$x_{a,r} = \begin{cases} 1 & \text{if } r \text{ is assigned to } a \in \mathcal{A} \\ 0 & \end{cases} \quad \forall r \in \mathcal{R}$$

According to the model, each vehicle has a capacity which must not be exceeded. Such constraint can be expressed as follows for both people and goods.

$$\sum_{r \in \mathcal{R}} P'_r \cdot x_{a,r} \leq P_a \quad \forall a \in \mathcal{A} \quad (4.5)$$

$$\sum_{r \in \mathcal{R}} G'_r \cdot x_{a,r} \leq G_a \quad \forall a \in \mathcal{A} \quad (4.6)$$

Furthermore, if a vehicle is already on the move, the request can be picked up only if the vehicle's charge is enough to satisfy such request as well. In this case, such requirement can be expressed as follows.

$$\sum_{r \in \mathcal{R}} e(\underline{s}'_r, \bar{t}'_r) x_{a,r} \leq B_a \quad \forall a \in \mathcal{A} \quad (4.7)$$

where $e : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}_{>0}$ is a function expressing the required energy to go from \underline{s}'_r to \bar{t}'_r .

It must also be ensured that each request has been served at most λ_r times , i.e. (4.8).

$$\sum_{a \in \mathcal{A}} x_{a,r} \leq \lambda_r \quad \forall r \in \mathcal{R} \quad (4.8)$$

Furthermore, each request must assigned at most to one vehicle. This is expressed by (4.9).

$$\sum_{r \in \mathcal{R}} x_{a,r} \leq 1 \quad \forall a \in \mathcal{A} \quad (4.9)$$

The above mentioned equations are based upon the work developed by *Hyland et al.* in [24].

This way allows to define the following cost function which expresses the number

of served requests.¹

$$\mathcal{J}_s = \sum_{a \in \mathcal{A}} \sum_{r \in \mathcal{R}} -x_{a,r} \quad (4.10)$$

The main strength of this approach is that can be integrated naturally in the formulation for the other steps, like for e.g. the one in Section 4.2.2.

Alternatively, the dispatching problem can be solved independently without being integrated in other steps. For example, in Chapter 3, some works are mentioned that make use of heuristics such as nearest neighbours. On the one hand, these approaches are known to obtain sub-optimal solutions for the problem; on the other hand, they provide flexibility and might result in less computational complexity.

4.2.2 Routing

After being assigned to incoming requests, vehicles must be routed in such a way that can reach all customers and therefore satisfy all the requests. In other words, the routing problem consists of determining paths, i.e. a series of edges, each vehicle must travel in the graph to fulfill the requests while respecting all the requirements and minimizing some metrics, i.e. a cost function. Since this problem can be reconduced to the vehicle routing problem, in particular dynamic pickup and delivery problems (reviewd by *Toth et al.* in [52] and *Laporte et al.* in [29]), the formulation used in this work will be based on this family of problems.

Within our model, binary flow variables will be used to identify whether a vehicle should traverse a link. Formally, this can be expressed as

$$V_{u,v}^a = \begin{cases} 1 & \text{if } a \text{ traverses } (u, v) \in \mathcal{E} \\ 0 & \end{cases} \quad \forall a \in \mathcal{A}, \forall u, v \in \mathcal{V}$$

Such a variable is handy to construct cost functions, like for e.g. the one proposed in (4.11) and (4.12)

$$\mathcal{J}_T = \sum_{a \in \mathcal{A}} \sum_{(u,v) \in \mathcal{E}} T_{u,v}^a V_{u,v}^a \quad (4.11)$$

$$\mathcal{J}_d = \sum_{a \in \mathcal{A}} \sum_{(u,v) \in \mathcal{E}} d_{u,v} V_{u,v}^a \quad (4.12)$$

Similar cost functions are frequently used in literature ([1]), as they are general, in a sense that many other metrics could be reconducted to them. For example, residual charging or operational costs are directly influenced by the two.

¹To align with subsequent formulations more seamlessly, the objective is to minimize the negative sum rather than maximizing it.

Furthermore, the formulation of the cost function derived from these metrics is rather trivial and intuitive, while at the same time producing desirable results in practice.

For example, in the effort to minimize the environmental impact of the ATS, the pollution index discussed in previous sections can be utilized to formulate the cost function outlined in (4.13).

$$\mathcal{J}_f = \sum_{a \in \mathcal{A}} \sum_{(u,v) \in \mathcal{E}} f_{u,v}^a V_{u,v}^a \quad (4.13)$$

Up to this point, the metrics under consideration have been geared towards minimization. Put differently, the goal is to reduce travel, encompassing both distance and time, to enhance system performance. Likewise, minimizing environmental impact is crucial in this scenario. However, there are instances where maintaining certain metrics at higher levels is preferable. For instance, closely tied to operational costs and environmental impact, it is advantageous to keep the state of charge at its maximum. Hence, it is imperative to maximize the cost function in Equation 4.14.

$$\mathcal{J}_B = \sum_{a \in \mathcal{A}} B_a \quad (4.14)$$

Finally, while it should also be explored whether the combination of those could improve the overall performance of the system. For this purpose, the cost functions can be combined using weights as follows.

$$\mathcal{J}_{tot} = \lambda_T \mathcal{J}_T + \lambda_d \mathcal{J}_d + \lambda_f \mathcal{J}_f + \lambda_B \mathcal{J}_B \quad (4.15)$$

where $\lambda_i \in \mathbb{R}$ with $i \in \{T, d, f, B\}$ are the weights.

It is not uncommon to also find in literature cost functions which are developed in terms of operational costs, as it provides a general idea to reason about this problem and allows for a systematic evaluation of different dispatching strategies. Moreover, the inclusion of operational costs in the literature emphasizes the real-world impact of dispatching decisions, aligning theoretical models with practical considerations. This connection to tangible costs not only enhances the applicability of proposed solutions but also contributes to the development of more realistic and effective dispatching strategies. Considering that, within our model in Section 4.1, each vehicle is modeled to have an operational cost, one can also derive a cost function similarly to what already done in [5].

Before formulating the routing problems formally, a small consideration must be made. For convenience, the set of nodes reachable from node u by traversing a single edge is defined as \mathcal{N}_u^+ , i.e. the ingoing neighbours of u , and \mathcal{N}_u^- as the set of outgoing neighbours of u .

Additionally, in order to ease the notation, the set containing all the initial stations of each request $r \in R_a$ will be indicated as \underline{S}'_a . Likewise, \bar{T}'_a will be the set of terminal stations of each request $r \in R_a$.

Unconstrained Version

On a basic level, i.e. without considering any type of operational constraints, one can formulate the routing problem as follows.

Algorithm 4.1: (*URP*) Given a transportation network \mathcal{G} and a set of vehicles \mathcal{A} , defined within the description in section Section 4.1, solve:

$$\min \text{ (4.11), (4.12), (4.13) or (4.15)}$$

s.t.

$$\sum_{u \in \mathcal{V}} V_{u,v}^a - \sum_{w \in \mathcal{V}} V_{v,w}^a = 0 \quad \forall a \in \mathcal{A}, v \in \mathcal{V} \setminus \{\underline{s}_a, \bar{t}_a\} \quad (4.16)$$

$$\sum_{u \in \mathcal{N}_{\underline{s}_a}^+} V_{\underline{s}_a,u}^a = 1 \quad \forall a \in \mathcal{A} \quad (4.17)$$

$$\sum_{u \in \mathcal{N}_{\bar{t}_a}^-} V_{u,\bar{t}_a}^a = 1 \quad \forall a \in \mathcal{A} \quad (4.18)$$

$$\sum_{u \in \mathcal{N}_{t_r'}^-} V_{u,t_r'}^a = 1 \quad \forall r \in \bar{R}_a, \forall a \in \mathcal{A} \quad (4.19)$$

$$\sum_{u \in \mathcal{N}_{s_r'}^+} V_{s_r',u}^a = 1 \quad \forall r \in \bar{R}_a, \forall a \in \mathcal{A} \quad (4.20)$$

$$p_u^a - p_v^a + P_a \cdot V_{u,v} \leq P_a - \sum_{r \in \mathcal{R}} P'_r \quad \forall u, v \in \mathcal{E}, v \neq \underline{s}_a, u, v \notin \underline{S}'_a \cup \bar{T}'_a \quad (4.21)$$

$$g_u^a - g_v^a + G_a \cdot V_{u,v} \leq G_a - \sum_{r \in \mathcal{R}} G'_r \quad \forall u, v \in \mathcal{E}, v \neq \underline{s}_a, u, v \notin \underline{S}'_a \cup \bar{T}'_a \quad (4.22)$$

$$g_{t_r}^a \geq g_{s_r}^a \quad \forall a \in \mathcal{A}, \forall r \in \bar{R}_a \quad (4.23)$$

$$p_{t_r}^a \geq p_{s_r}^a \quad \forall a \in \mathcal{A}, \forall r \in \bar{R}_a \quad (4.24)$$

$$g_u^a \leq P_a; \quad p_u^a \leq P_a, \quad \forall a \in \mathcal{A} \quad (4.25)$$

(4.16) insures that for all edges which do not lead to a source or destination, if a reaches v from a road, an incoming flow will lead to an outgoing one, i.e. it is the flow conservation constraint. In other words, they guarantee connections between roads. (4.17) - (4.18) insure that each universal source and destination is reached ones. Similarly, (4.19) - (4.20) achieves the same result, but for each requests. (4.21)-(4.25) assure that no sub-tour are presentes. This is inspired from the Miller-Tucker-Zemli formulation and utilizes two continuous

decision variables $p^a, g^a \in \mathbb{R}_{\geq 0}$ representing the cumulative load of the vehicle a . Specifically, (4.23)-(4.24) are included to insure that a starting node of a request must be visited before the terminal location.

In order to find the minimum number of vehicles required, among others, one can use the method described *Brodo* in [5].

Constrained Version

While it could bring some interesting insights, the URP does not contain the necessary information to provide efficient routing for the scenario considered in this work. The goal is to identify the best possible path that serves all the requests and satisfies all the constrain requirements. For this formulation, another variable will be needed to deal with time-related requirements. The decision variable $s_u^a \in \mathbb{R}_{\geq 0}$ indicates the service time of vehicle a at node u . This concept is based on the work in [25]

The *Congestion-Free Routing Problem (CRR)* is formally defined as follows.

Algorithm 4.2: (*CRR*) Given a transportation network \mathcal{G} and a set of vehicles \mathcal{A} , defined within the description in section Section 4.1, solve:

$$\min(4.11), (4.12), (4.13) \text{ or } (4.15)$$

s.t.

$$(4.16) - (4.20)$$

$$\sum_{a \in \mathcal{A}} V_{u,v}^a \leq c_{u,v} \quad \forall (u, v) \in \mathcal{E} \quad (4.26)$$

$$s_u^a + T_{u,v}^a - M * (1 - V_{u,v}^a) \leq s_v^a \quad \forall (u, v) \in \mathcal{E}, \forall a \in \mathcal{A} \quad (4.27)$$

$$a'_v \leq s_v^a \leq b'_v \quad \forall v \in \mathcal{V}, \forall a \in \mathcal{A} \quad (4.28)$$

$$\sum_{(u,v) \in \mathcal{E}} e_{u,v}^a \cdot V_{u,v}^a \leq B_a(0) \quad \forall a \in \mathcal{A} \quad (4.29)$$

$$M = \max_{(u,v) \in \mathcal{E}} \{b_u + T_{u,v} - a_u\}$$

(4.26) insures the number of vehicles in the link $\langle u, v \rangle$ do not exceed the capacity of that link. (4.27) establishes the relationship between the service time of each node, implying that the service time of a predecessor must be lower than the successor. (4.28) establish the time window constraints, indicating that it must be within the interval of the request. For nodes which are not associated with a termination node of a request, one will simply set $a'_v = 0$ and $b'_v = \infty$, also insuring that $s_v^a \geq 0$. (4.29) assures that the vehicle charge is enough to cover all path. $B_a(0)$ can be assumed to be 100, i.e. that the batteries are full at the beginning of service. In this formulation, the sub-tour elimination constraint is not required at it is already imposed by (4.27).

Combination with Dispatching

In order to combine dispatching and routing into the same formulation, one must adapt some of the conditions specified in the formulations above. More specifically, each equation related to the requests must be changed to accommodate the fact that requests have not been previously assigned.

Accordingly, the *Routing and Dispatching Problem (RDR)* can be formulated as follows.

Algorithm 4.3: (*RDR*) Given a transportation network \mathcal{G} , a set of vehicles \mathcal{A} and a set of requests \mathcal{R} , defined within the description in section Section 4.1, solve:

$$\min(4.10), (4.11), (4.12), (4.13) \text{ or } (4.15)$$

s.t.

$$(4.5), (4.6)$$

$$(4.8), (4.9)$$

$$(4.16)$$

$$(4.26), (4.28), (4.29)$$

$$\sum_{u \in \mathcal{N}_{\underline{s}_a}^+} V_{\underline{s}_a, u}^a = 1 \quad r \in \mathcal{R}, \forall a \in \mathcal{A} \quad (4.30)$$

$$\sum_{u \in \mathcal{N}_{\bar{t}_a}^-} V_{u, \bar{t}_a}^a = 1 \quad r \in \mathcal{R}, \forall a \in \mathcal{A} \quad (4.31)$$

$$\sum_{u \in \mathcal{N}_{\bar{t}_r'}^-} V_{u, \bar{t}_r'}^a \geq x_{a,r} \quad \forall r \in \mathcal{R}, \forall a \in \mathcal{A} \quad (4.32)$$

$$\sum_{u \in \mathcal{N}_{\underline{s}_r'}^+} V_{\underline{s}_r', u}^a \geq x_{a,r} \quad \forall r \in \mathcal{R}, \forall a \in \mathcal{A} \quad (4.33)$$

$$s_u^a + T_{u,v}^a - M * (1 - V_{u,v}^a) \leq s_v^a \quad \forall (u, v) \in \mathcal{E}, \forall a \in \mathcal{A} \quad (4.34)$$

$$u, v \notin \{\underline{s}_a', \bar{t}_a' | \forall r \in \mathcal{R}\}$$

$$s_{\bar{t}_a'}^a - s_{\underline{s}_a'}^a \geq x_{a,r} \quad \forall r \in \mathcal{R}, \forall a \in \mathcal{A} \quad (4.35)$$

Constraints (4.30) and (4.31) impose that a vehicle must leave the starting depot and reach its final depot, respectively. (4.32) guarantees that, when a request r is allocated to vehicle a , the vehicle must reach the terminal station of the request at least once. Moreover, (4.33) ensures that the vehicle traverses the starting station associated with the assigned request. (4.34) is the subtour elimination constraint, with, however a further constraint represented by (4.35), which forces the end station of a request to be visited after the start station. These requirements, furthermore, with this formulation, also ensure that a vehicle can only move from one node to the other only if they left the starting depot

as well.

As an alternative to (4.30) and (4.31), one can also relate the two constraints to the request assignment, which would transform the requirements as following.

$$\sum_{u \in \mathcal{N}_{\underline{s}_a}^+} V_{\underline{s}_a, u}^a = \max_{r \in \mathcal{R}} \{x_{a,r}\} \quad \forall a \in \mathcal{A} \quad (4.36)$$

$$\sum_{u \in \mathcal{N}_{\bar{t}_a}^-} V_{u, \bar{t}_a}^a = \max_{r \in \mathcal{R}} \{x_{a,r}\} \quad \forall a \in \mathcal{A} \quad (4.37)$$

In other words, a vehicle must leave the starting depot and reach the final depot if a request is assigned to it.

4.2.3 Rebalancing

In simpler terms, the rebalancing problem revolves around efficiently redistributing autonomous vehicles (AVs) to optimize their responsiveness to new ride requests while minimizing any existing imbalances in the system. The goal is to fine-tune the positioning of AVs, ensuring they are strategically placed to promptly meet user demands and address any inherent irregularities in the distribution of service requests. This challenge is particularly crucial in ride-sharing systems and transportation systems alike, where the dynamic nature of user requests and varying demand across different locations can lead to imbalances in the fleet's distribution. Effectively tackling the rebalancing problem enhances the overall efficiency of the system, providing users with quicker response times and a more evenly distributed service, ultimately contributing to a smoother and more reliable autonomous transportation network.

The formulation of the rebalancing problem is partially inspired from the works of *Wallar et al.* in [57].

The model previously described allows to reason on the rebalancing model in a novel manner when compared to the literature reviewed in Chapter 3. More specifically one could deal with the rebalancing by leveraging the request arrival rate λ_r and the fact that vehicles are assumed to have a starting and terminal station, \underline{s}_a and \bar{t}_a respectively.

Considering the situation depicted in Fig 4.3 with a set of requests \mathcal{R}' and a sets of vehicles \mathcal{A}' . According to this example, the urban area has been divided in three cells (in red, blue and green). This division has been done arbitrarily in this case, but different strategies can be used ([57], [66]). These regions must be specified, however, in such a way that they contain exactly one node belonging to \mathcal{V}_c and more than $n \in \mathbb{Z}_{\geq 1}$ nodes belonging to \mathcal{V}_n . Informally, this means that those regions are built around each charging station or depot and contain a certain amount of normal nodes. Different strategies could be used to determine this amount, like for e.g. all the nodes within a radius r from each terminal node or within a driving distance T_{max} . Nevertheless, as a result of this, one will obtain a set of regions R . Formally, each region around a node $v \in \mathcal{V}_c$ can

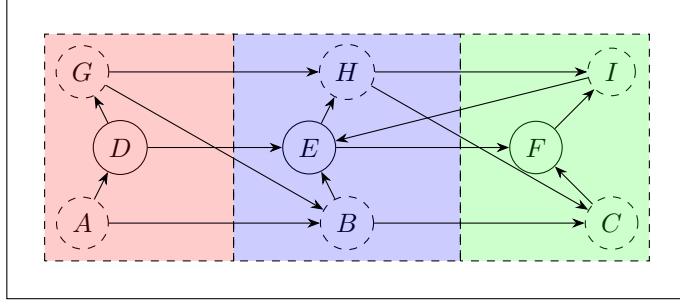


Figure 4.3: Simplified example for the rebalancing strategy. Nodes D, E, F (continuous circles) to be the depot (or charging stations), while the other nodes (dashed circles) to be normal nodes. The graph has been previously divided in three areas (red, blue and green).

be defined as a subgraph $\mathcal{G}_v = \langle \mathcal{V}'_v, \mathcal{E}'_v \rangle$, where

$$\mathcal{V}'_v = \{u \in \mathcal{V}_n : (u, v) \in \mathcal{E} \wedge (v, u) \in \mathcal{E}, f(v, u) = 1\} \cup \{v\} \quad (4.38)$$

$$\mathcal{E}'_v = \{(u, w) \in \mathcal{E} : u, w \in \mathcal{V}'_v\} \quad (4.39)$$

In (4.38), the function $f : \mathcal{V}_c \times \mathcal{V}_n \rightarrow \{0, 1\}$ is used to establish whether a node u belongs to the region of v ($f(v, u) = 1$) or not ($f(v, u) = 0$), as discussed before. Accordingly, one can obtain the total number of requests of region v ($|\mathcal{R}'_v|$) by considering the requests which have a starting node in \mathcal{V}_n . In other words

$$|\mathcal{R}'_v| = \sum_{r \in \mathcal{R}'_v} \lambda_r \quad (4.40)$$

$$\mathcal{R}'_v = \{r \in \mathcal{R} : s_r' \in \mathcal{V}'_v\} \quad (4.41)$$

In simple words, the set of \mathcal{R}'_v indicates all the requests that start within the region of the node $v \in \mathcal{V}_c$.

Analogously, one can provide an alternative definition for \mathcal{R}'_v in terms of the terminal station u .

$$\mathcal{R}'_u = \{r \in \mathcal{R} : \bar{t}_r' = u\} \quad (4.42)$$

Requests, therefore, are distributed over those regions and it is not hard to envision scenarios where requests are unequally distributed, i.e. regions with a higher $|\mathcal{R}'_v|$ than others. As a result, due to the heterogeneous nature of the nodes and the requests, an ATS might experience imbalance, as AVs can be scattered through the whole graph if requests have different terminal nodes. Conversely, the opposite might happen, if all the requests have the same terminal node, but start from different areas. Considering, for example, Fig 4.3, assuming all the requests in \mathcal{R}' to be served, each vehicle's starting position is now either D, E,

or F, i.e. $s_a \in \{D, E, F\} \quad \forall a \in \mathcal{A}'$. This is due to the fact that, according to the transportation network model, since the requests have all been served, the vehicles have all reached their previous ending stations \bar{t}_a , which became their new starting nodes. Therefore, if the choice of the terminal node is not properly made, not enough vehicles could be present in a region to serve the incoming requests. As a result, the solution of the rebalancing problem becomes ensuring that there is a sufficient number of vehicles at the depot nodes to fulfill as many requests as possible in the given area, potentially covering all requests. This, therefore, can be reduced to an assignment problem.

Unconstrained Rebalancing Problem

If one aims at solving the rebalancing problem separately or, more generally, aims at route vehicles towards locations with high requests, this could be achieved as presented in this section. Similarly to Section 4.2.2, a flow variable will be used to indicate whether an idle vehicle, i.e. a vehicle not currently transporting customers or goods, should be moving from a node to another. Formally, this is expressed as follows.

$$y_{u,v}^a = \begin{cases} 1 & \text{if } a \text{ traverses } (u, v) \in \mathcal{E} \\ 0 & \end{cases} \quad \forall a \in \mathcal{A}, \forall u, v \in \mathcal{V}$$

Within the assumptions made at the beginning of this section, it follows that a vehicle will be considered idle if it is currently at its final station.

For the sake of simplicity and brevity, a generic cost function \mathcal{J}' will be considered for the entirety of this section since the choice of a cost function is similar to the one for Section 4.1.

Finally, within the model established in prior sections, the *Unconstrained Rebalancing Problem (UReP)* can be formulated as following.

Algorithm 4.4: (*UReP*) Given a transportation network \mathcal{G} , a set of idle vehicles \mathcal{A}' and a set of nodes belonging to \mathcal{V}_c , defined within the description in section Section 4.1, solve:

$$\min \quad \mathcal{J}'$$

s.t.

$$\sum_{u \in \mathcal{V}} y_{u,v}^a - \sum_{w \in \mathcal{V}} y_{v,w}^a = 0 \quad \forall a \in \mathcal{A}', v \in \mathcal{V} \setminus \{\underline{s}_a, \bar{t}_a\} \quad (4.43)$$

$$\sum_{u \in \mathcal{N}_{\underline{s}_a}^+} y_{\underline{s}_a,u}^a = 1 \quad \forall a \in \mathcal{A}' \quad (4.44)$$

$$\sum_{\bar{t} \in \mathcal{V}_c} \sum_{u \in \mathcal{N}_{\bar{t}}^-} y_{u,\bar{t}}^a = 1 \quad \forall a \in \mathcal{A}' \quad (4.45)$$

$$\sum_{a \in \mathcal{A}'} \sum_{u \in \mathcal{N}_{\bar{t}}^-} y_{u,\bar{t}}^a \leq z_{\bar{t}} \quad \forall \bar{t} \in \mathcal{V}_c \quad (4.46)$$

$$\sum_{a \in \mathcal{A}'} \sum_{u \in \mathcal{N}_{\bar{t}}^-} G_a \cdot y_{u,\bar{t}}^a \geq \sum_{r \in \mathcal{R}'_{\bar{t}}} G'_r \quad \forall \bar{t} \in \mathcal{V}_c \quad (4.47)$$

$$\sum_{a \in \mathcal{A}'} \sum_{u \in \mathcal{N}_{\bar{t}}^-} P_a \cdot y_{u,\bar{t}}^a \geq \sum_{r \in \mathcal{R}'_{\bar{t}}} P'_r \quad \forall \bar{t} \in \mathcal{V}_c \quad (4.48)$$

(4.43) insures the flow conservation. (4.44) indicates that the vehicles leave the starting node only from one edge. (4.45) guarantees a vehicle must reach only one deposit at the time. (4.46) insures there the capacity of the deposits is respected. Finally, (4.47) and (4.48) are used to ensure the vehicles being rebalanced to a certain station have the capacity to fulfill the requests of the area.

4.3 The Complete ATS Management Problem Formulation

While the previously described approaches are suitable for standalone applications, amalgamating routing, dispatching, and rebalancing into a single problem would entail a considerable increase in the number of variables, resulting in an inefficient formulation. However, with the insights obtained in Section 4.2, a more streamlined formulation which aims at solving all of the aforementioned challenges can be defined.

Let

$$b_{\bar{t}}^a = \begin{cases} 1 & \text{if } a \text{ is assigned to } \bar{t} \\ 0 & \end{cases} \quad \forall a \in \mathcal{A}, \forall \bar{t} \in \mathcal{V}_c$$

the *Complete ATS Management problem (CATSM)* can be formulated as follows.

Algorithm 4.5: (*CATSM*) Given a transportation network \mathcal{G} , a set of vehicles \mathcal{A} , a set of total requests \mathcal{R} , defined within the description in section Section 4.1, and the sets of requests per region \mathcal{R}' following (4.41), solve:

$$\min(4.11), (4.12) \text{ or } (4.13)$$

s.t.

$$(4.17) \text{ or } (4.36)$$

$$(4.26), (4.28), (4.29)$$

$$(4.32) - (4.35)$$

$$\sum_{u \in \mathcal{V}} V_{u,v}^a - \sum_{w \in \mathcal{V}} V_{v,w}^a = 0 \quad \forall a \in \mathcal{A}, v \in \mathcal{V} \setminus \{\underline{s}_a\} \setminus \mathcal{V}_c \quad (4.49)$$

$$\sum_{u \in \mathcal{N}_{\bar{t}}^-} V_{u,\bar{t}}^a = b_{\bar{t}}^a \quad \forall a \in \mathcal{A}, \forall \bar{t} \in \mathcal{V}_c \quad (4.50)$$

$$\sum_{\bar{t} \in \mathcal{V}_c} b_{\bar{t}}^a \leq 1 \quad \forall a \in \mathcal{A} \quad (4.51)$$

$$\sum_{a \in \mathcal{A}} b_{\bar{t}}^a \leq z_{\bar{t}} \quad \forall \bar{t} \in \mathcal{V}_c \quad (4.52)$$

$$G_l \geq \sum_{a \in \mathcal{A}} G_a \cdot b_{\bar{t}}^a \geq \sum_{r \in \mathcal{R}'_{\bar{t}}} G'_r \quad \forall \bar{t} \in \mathcal{V}_c - \sum_{a \in \mathcal{A}} \mathbb{1}_{\underline{s}_a == \bar{t}} G_a \quad \forall a \in \mathcal{A}, \forall \bar{t} \in \mathcal{V}_c \quad (4.53)$$

$$P_l \geq \sum_{a \in \mathcal{A}} P_a \cdot b_{\bar{t}}^a \geq \sum_{r \in \mathcal{R}'_{\bar{t}}} P'_r \quad \forall \bar{t} \in \mathcal{V}_c - \sum_{a \in \mathcal{A}} \mathbb{1}_{\underline{s}_a == \bar{t}} P_a \quad \forall a \in \mathcal{A}, \forall \bar{t} \in \mathcal{V}_c \quad (4.54)$$

$$\sum_{\bar{t} \in \mathcal{V}_c} b_{\bar{t}}^a \geq \max_{r \in \mathcal{R}} \{x_{a,r}\} \quad a \in \mathcal{A} \quad (4.55)$$

(4.49), similarly to (4.16), imposes the flow conservation for all nodes except the startind depot and the set of terminal depots. (4.50) impose a vehicle to stop at a terminal depot. (4.51) ensures each vehicle can be assigned maximum to one terminal depot. (4.52) guarantees that the depot parking facility is not exceeded. (4.53) and (4.54) ensure that the vehicles' capacity is enough to serve the next batch of requests. G_l and P_l impose an upperbound for goods and people capacity, respectively, which is useful to reduce the number of rebalancing vehicles in case the aim is not to, directly or indirectly, minimizing it. Finally, (4.55) insures that a vehicle moving to satisfy a requests is being assigned to a final depot. The function $\mathbb{1}_x$ is commonly known as the indicator function, denoting a boolean variable x that can take values true, false. Specifically, $\mathbb{1}_x$ is defined as follows:

$$\mathbb{1}_x = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{if } x \text{ is false} \end{cases} \quad (4.56)$$

In other words, $\mathbb{1}_x$ equals 1 when x is true and equals 0 when x is false, making it a convenient way to express the truth value of the expression x in mathematical notation.

Compared to most of the reviewed literature, the strength of this approach lies in the fact that the rebalancing problem resembles more an assignment problem than routing. As a matter of fact, vehicles are routed in such a way that all the requests are guaranteed to be served and at the same time, the imbalance nature of ATSSs is tackled by ensuring that each region has enough vehicle to deal with future requests.

Furthermore, due to its vehicle-centric approach, this formulation offers significant flexibility across several dimensions: *(i)* it accommodates diverse request types, such as goods or people, *(ii)* it allows for a wide range of vehicle characteristics and specification, like for e.g. different capacities or discharging profiles, and *(iii)* it remains adaptable to various node characteristics, as nodes have different number parking spaces, for example. Simultaneously, it exhibits scalability, with the number of variables increasing linearly in relation to the edges in the graph and the number of vehicles under consideration.

Notably, this formulation solves implicitly two additional issues relating to ATSSs, i.e. ride-sharing and delivery pooling. As a matter of fact, by addressing demands with varying pick-up and drop-off locations, this formulation inherently resolves the two additional challenges associated with ATSSs. Through the capability to efficiently accommodate customers requiring transportation between disparate points, the system seamlessly integrates both ride-sharing, where passengers share a vehicle for travel, and delivery pooling, which consolidates multiple goods deliveries into a single route

4.4 Use Case Analysis

The main purpose of this use case is to demonstrate the applicability of the model in the real world and the feasibility of the solutions presented in Section 4.2 and Section 4.3. This use case is based on real-world data extracted from taxi rides in manhattan in 2016 ([11]).

The data has been prepared in the following way. Since the data retrieved is based on the whole New York City, in order to decrease the complexity, some filters have been applied to only target the Manhattan area. More specifically, the pick-up and drop-off points which were outside of Manhattan have been filtered out. Subsequently, these pick-up and drop-off points have been clustered in 40 locations. Finally, these locations have been mapped to the real Manhattan road network and the shortest path between five neighbours nodes has been traced in order to create a simplified road network, which can be seen in Fig 4.4. The graph considered has 482 locations and 1622 routes.

As a consequence, the complexity of Manhattan's road network has been significantly reduced. Within the context of this project, and specifically in this section, this simplification is deemed acceptable since only proof of concepts are being considered. Furthermore, employing a more intricate representation of

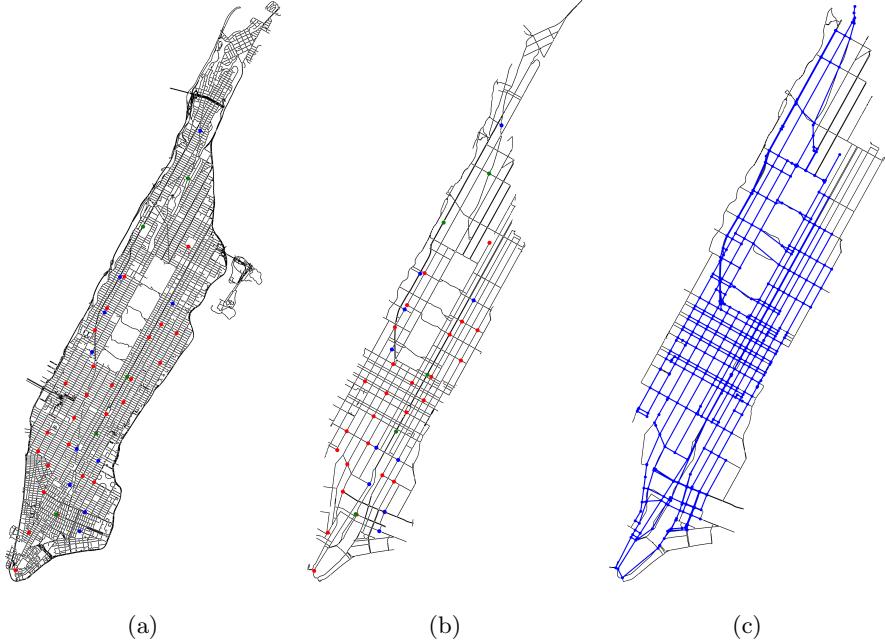


Figure 4.4: Manhattan's Road Network Representation. In (a), red points are used to indicate pick-up stations, while blue delineates drop-off points and green is used to pin-point the fictitious depots. These are obtained by the historical data. Furthermore, pick-up points can also be used as drop-off, and viceversa. (b) shows the semplified road network consistsitnf of only the most important streets. Finally, (c) shows the final version of the road network after the application of the minimum spanning tree algorithm.

the road network would not significantly enhance the objectives at hand. This streamlined approach not only facilitates clearer demonstration of concepts but also expedites the analysis process, enabling a more focused examination of essential elements. Thus, for the current scope of this work, the simplified representation remains suitable and pragmatic. According to the discussion in Section 4.2.3, the region has been divided in multiple areas as well. This can be observed in Fig 4.4a. Although the division in areas it's not explicitely stated, five green-highlighted nodes can be observed. These nodes represent depots and have been created to fulfill the same purpose of Section 4.2.3. In order to improve the performance of the simulation, the road network has been simplified by considering only the most important streets in Manhattan, as shown in Fig 4.4b and Fig 4.4c. This approach optimizes resources while still providing valuable insights into the model performance. Fig 4.4c specifically has been obtained by first applying the Minimum Spanning Tree (MST) algorithm ([35]) to the relevant nodes for the requests. The MST is used to calculate the smallest pos-

sible connected subtree that spans all the vertices in a given graph. In essence, constructing MST involves iteratively selecting edges with the smallest weight while ensuring that no cycles are formed. This process starts with any arbitrary vertex and gradually expands by adding the lightest edge connected to the existing tree until all vertices are encompassed. The connections between the nodes in the resulting MST graph have been restored using the original road network's street information to maintain the connectivity.

While information regarding passenger demands are relatively easy to obtain, other data must be crafted or estimated. In particular, details about the constraints outlined in Section 4.1 can be challenging to gather as they are often specific to certain scenarios or dependent on external factors not captured by the model. For instance, determining the threshold in the congestion model is complex, especially considering it could easily reach thousands in a densely populated city like NYC. Likewise, the time windows of a request is also very specific to the application. For the rest of this work, this information will be derived from the available data and created, if required. For example, the traveling time for each link $T_{u,v}^a$ can be derived with the assumption of a constant speed. Trivially, since the distance is known, the travelling time can be calculated as follows:

$$T_{u,v}^a = \frac{d_{u,v}}{l_{u,v}} \quad (4.57)$$

where $l_{u,v}$ is the speed limit of the link $\langle u, v \rangle$.

Although data on customer mobility is readily accessible, obtaining precise information on goods delivery poses a challenge. This includes details on the specific category of goods being transported and the exact dropoff locations. Multiple might be the reasons for it, such as privacy, commercial sensitivity or due to the highly heterogenous nature of this field. Nevertheless, a potential approach to glean insights into goods delivery could involve leveraging publicly accessible datasets such as the GetFood Historical Data ([38]). Amid the COVID-19 pandemic, initiatives like GetFoodNYC have been instrumental in facilitating emergency home food delivery for residents unable to access food through conventional means. Although the data provided does not contain delivery location, this can be simulated by picking a random location within the road network and the quantity can be estimated by the available data.

The simulations have been carried out using the CBC ([50]) and Gurabi ([19]) solvers used in combination with PuLP ([12])².

4.4.1 Rebalancing-Free ATS Analysis

This use-case refers to the RDR (Algorithm 3) with the alternatives defined in (4.36)-(4.37) and the goal is to serve as many requests as possible, i.e. to minimize the cost function in (4.8). For this use-case, since the rebalancing is not considered, the end depots are chosen randomly at the beginning of the

²On a 2018 Intel i7 MacbookPro

simulation and only the starting and ending depots of transporting vehicles are exchanged at every iteration. As a result, transporting vehicles move from starting to end depot at one iteration and viceversa at the next iteration. Two main aspects are considered, namely charging time and congestion constraints. These two variables, which greatly influence the system's performance, have been varied to analyse how the system behave with (*i*) high roads capacity low requests rate, i.e. high charging times, since vehicles can start moving after their battery has been charge with an additional 40%; (*ii*) same road capacity, but high request rate (additional 5 % of charging); (*iii*) low road capacity and high charging rate; finally (*iv*) lower capacity, but higher charging rate. The details of the simulations are shown in Table 4.2 and the starting distribution of the vehicles in the depot is described in Table 4.3. The overall performance is summarized in Table 4.4³.

	Sim. 1	Sim. 2	Sim. 3	Sim. 4	Sim. 5
Vehicles #	24	-	-	-	-
Iterations	10	-	-	-	-
Requests / <i>i</i>	10	-	-	-	-
Charging % / <i>i</i>	60	40	30	10	30
$c_{u,v} / i$	50	-	-	5	5

Table 4.2: Details of the Rebalancing-less Simulation. “*i*” stands for iteration and “-” means the entry has not been changed from the previous one

Depot ID	1	2	3	4	5
# Vehicles	3	6	8	4	3

Table 4.3: Depot Distribution Used During the Simulation

	Sim. 1	Sim. 2	Sim. 3	Sim. 4	Sim. 5
Total Distance (m)	731993	749006	824307	585955	718618
Average Distance (m)	30500	312089	34346	24415	29942
Total Time (s)	14640	14980	16486	11719	14372
Average Time (s)	610	624	687	488	599
Unique Road Used	1171	1137	1244	1021	1170
Request Served (%)	64	63	68	57	60

Table 4.4: Overview of the System's Performance for Routing-less Simulation. The unique road used entry counts how many different roads have been used during the simulation. In other words, it measures the sum of routes that have been travelled by at least one vehicle.

³Further information can be found in Appendix A

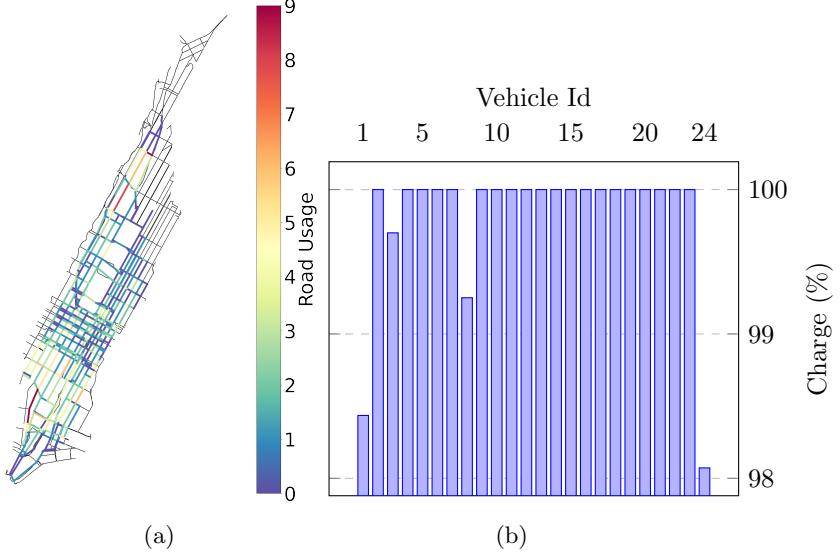


Figure 4.5: (a) illustrates the utilization of the entire road network. Road usage is determined by aggregating the total number of vehicles traversing each road segment throughout the shift duration. The most used roads are shown in red, while blue indicates low activity. (b) displays the average vehicle charge during the shifts.

In a system characterized by wide charging times and absence of congestion limits ($c_{u,v} > |\mathcal{A}| \quad \forall (u,v) \in \mathcal{E}$), the performance depicted in Fig 4.5 unfolds. Notably, the objective is not focused on minimizing overall travel time (4.11) or total distance traveled (4.12), but rather on maximizing the number of served requests. Consequently, with more charging time and less road constraint, vehicles tend to cover more distance. This inclination is further facilitated by the lack of real capacity constraints on roads and the freedom for vehicles to traverse them at any time.

As a consequence, the utilization of the road network is uneven, as illustrated in Fig 4.5a, where the most-utilized links are highlighted in red. This uneven distribution can be attributed partly to the concentration of drop-off and pick-up points, as well as depots, primarily within downtown Manhattan. However, the presence of a depot uptown causes an area of relatively high vehicles activity, although there are not many relevant location there, as shown in Fig 4.4b. This suggests that if the aim is to mitigate vehicle usage and road congestion, this region shouldn't exhibit high vehicular activity. This consideration becomes significant when addressing rebalancing strategies.

Furthermore, Fig 4.5b suggests that most of the vehicles have not been used during the shift. This can be inferred by observing that, for half of the vehicles, the average charge is exactly 100%, i.e. the same as the starting battery charge. This suggests that the fleet of vehicles could be reduced in number and

maintaining the same capacity.

Fig 4.6 shows the same metrics for the third simulation. Although the charging time is reduced in half, this simulation results in a much more efficient use of the road network. First of all, as shown in Table 4.4, more requests could be served in the same amount of time. Furthermore, although with less charge, vehicles tend to travel more. Though it may seem counterintuitive, the reason could be that the solver has been limited to ten minutes. As a result of this restriction, it is likely that, within this time limit, the solver has been able to prune out unfeasible solutions thanks to the more stringent requirement. Furthermore, from Fig 4.6a, contrary to Fig 4.5a, it is evident that the road network is used more evenly. As expected, there are still roads that stand out as the most used ones, which reflects the result obtained previously and it is still to associate with the fact that terminal depots are not assigned properly. On the contrary, as shown in Fig 4.6b, the fleet usage is similar when compared to previous approaches. Although the distribution of average charge appears to be slightly more uneven compared to Fig 4.5b , with some vehicles maintaining an average charge of around 90%, most vehicles are not used entirely. This suggests that the fleet could be reduced in this case as well. Although vehicles do not fully recharge at the end of each shift, according to their evolution over the simulation period ⁴, since the fleet is generously big, other vehicles can replace charging vehicles for the shift. On the one hand, this results on more vehicles used, however the overall average is well above 80% at the end of the shift.

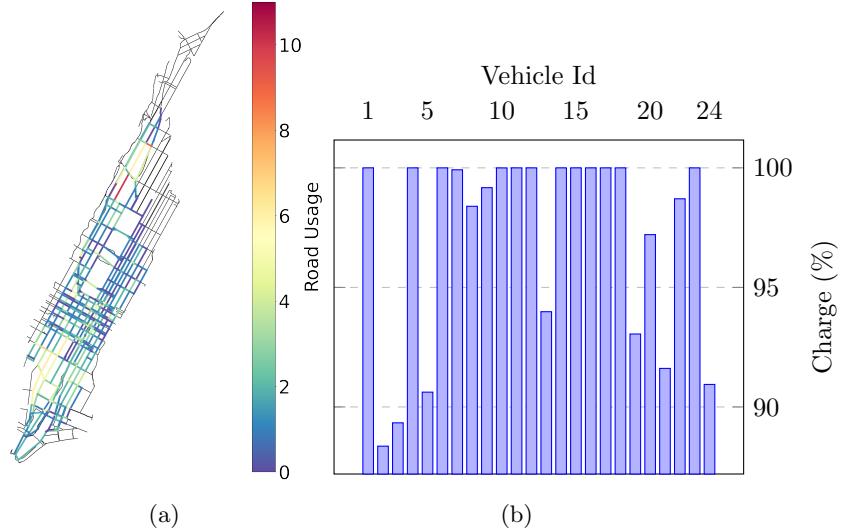


Figure 4.6: Overview of System's Performance for Routing-less Simulation 3.

The performance of simulation number four can be observed in Fig 4.7. In this case, a more stringent approach for the congestion limit has been con-

⁴The evolution of a set of vehicles can be found in Fig A.4a

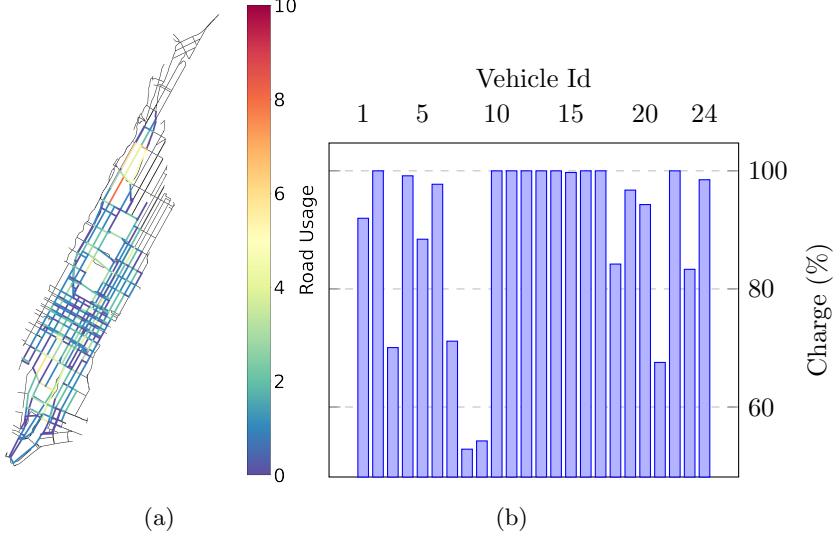


Figure 4.7: Overview of System’s Performance for Routing-less Simulation 4.

sidered, i.e. no more than five vehicles can traverse a link at the same time. Furthermore, vehicles have only been charged up to ten per cent after each iteration. As a result, the fleet has been able only to serve up to 57%. While the ten minutes limit on the solver is contributing, the strict capacity on the streets in combination with the low charging rate is resulting in a major downgrade in performance. The vehicles tend to travel less, for less time and can not reach all the requests. Since the graph is an abstraction of the real street network in Manhattan, one way roads are also modelled, therefore, reaching a particular node, might be hindered by a common link not being accessible by more than four vehicles at the same time. Furthermore, road capacity and charging limits have also a high influence on the fleet usage. As shown in Fig 4.7b, more vehicles tend to be used, as the overall charge is lower, with an average charge as low as below 60% . The evolution of the vehicles with the lowest average charge over time shown in Fig A.4b suggests two main phenomena: *i* vehicles tend to be used for up to 50% of the shift duration or until their charge is enough to cover the requests and *(ii)* as soon as their charge is not enough to cover a batch of requests, they are not considered for a relatively long period of time. These two phenomena are clearly fleet dependend, since more vehicles in the fleet will accentuate them more and a lower one would imply a bigger need for every vehicle to be active.

A similar effect can be found by analysing simulation number five⁵

⁵The same metrics can be found in Fig A.2 and Fig A.3

4.4.2 Rebalancing-Integrated System Analysis

To ensure a comprehensive and equitable comparison of vehicle rebalancing within the simulations, majority of the parameters have been kept consistent. Table 4.5 provides a summary of the primary parameters utilized in this simulation. The introduction of the rebalancing strategy, as outlined in Section 4.2.3, was motivated by the primary objective of enhancing service fulfillment rates. By strategically redistributing vehicles across the transportation network, the aim was to ensure a more consistent and reliable service delivery to customers. However, this strategic move was not without its anticipated drawbacks, notably in the form of an expected increase in travel statistics, especially since the length of the route is not included in the system's objective. Therefore, the system's objective is a trade-off between the number of requests served and the number of rebalancing vehicles in order to decrease unnecessary travels.

	Sim. 1	Sim. 2	Sim. 3	Sim. 4
Charging % / i	60	30	10	30
$c_{u,v} / i$	50	10	5	5

Table 4.5: Details of the Simulation with Vehicle Rebalaсing. Vehicles' number, iterations and requests for iterations have not been changed. Furthermore, the same requests have been picked for a more fair comparison.

	Sim. 1	Sim. 2	Sim. 3	Sim. 4
Total Distance (m)	806280	874937	871167	972776
Average Distance (m)	33595	36456	36299	40532
Total Time (s)	16126	17499	17423	19456
Average Time (s)	672	729	726	811
Unique Road Used	1193	1206	1186	1278
Request Served (%)	81	78	81	79

Table 4.6: Performance of the Simulation with Vehicle Rebalaсing

The hypothesis has been confirmed by the simulation. Table 4.6 contains a summary of the performance of this simulation. When comparing simulations with similar parameters, the impact of rebalancing on travel statistics becomes evident, with notable increases observed across all factors. Conversely, the number of served requests has notably risen across all simulations. Specifically, performance among simulations incorporating rebalancing remained relatively consistent, hovering around 80%. This highlights the effectiveness of the rebalancing strategy in enhancing service fulfillment rates despite the associated increase in travel metrics. An additional goal associated with rebalancing was a more efficient vehicle use road network usage. Fig 4.8 shows the results measure for Simulation 2 and 3. Starting from Fig 4.8a and Fig 4.8c, when comparing it to previous results, the road usage did not improve. On the contrary, since

vehicles are travelling to different depots potentially at every iteration, different routes must be taken. Interestingly, when observing the number of unique road used, this does not increase dramatically from the routing-less simulation. Furthermore, the road network analysis indicates a similar imbalance, where downtown and uptown manhattan have the highest usage. Furthermore, the similar roads in both simulation measured a high usage. While this is against the previous hypothesis, i.e. that rebalancing would have decrease the usage uptown due to a lower amount of nodes, it offers another important insight. By observing the road usage in both cases, one is able to determine the most important streets and, therefore, able to make more informed decisions on how to manage the road network in the future.

Fig 4.8b and Fig 4.8d show the evolution of the state of charge for a set of vehicles. Compared to previous results, as expected, the charge of the vehicles is less consistent during the shift. During the process of rebalancing, vehicles often travel to various depots, resulting in fluctuating distances traveled. Consequently, this can lead to variations in the final charge of the vehicles, either lower or higher. While vehicles have been picked to show this fact, it is still worth noting that, as could be observed in Fig 4.8b, the state of charge of some vehicles remains well above 90% during the shift, which suggests the fleet could be potentially be reduced in number and still offer the same quality of service.

4.5 Conclusion

This chapter demonstrated how an autonomous transportation system can be modeled, organized and evaluated in order to offer a better service to the customers. The dispatching, routing and rebalancing problems have been formulated and different versions have been proposed. Furthermore, a use case is demonstrated and a possible analysis has been proposed, with the goal of formulating considerations for a better fleet and road network usage. While this is to consider only as an example, multiple outcomes have been reached, which could be used to improve the quality of life of residents of traffic-cities cities such as NYC. Even if it focused on customer transportation and door-to-door delivery for urban cities, however, the analysis of a more bespoken model can be adapted for business logistics and transportation. Anticipated consequences included both negative effects, like heightened road and vehicle utilization due to rebalancing, and positive outcomes, such as improved service efficiency. However, unexpected outcomes, such as increased delivery demands with road capacity constraints, were also observed. The system, furthermore, can still be improved, as described in Section 4.1.1 and observed in Section 4.4. The latter, specifically, highlighted the need of a better cohesive strategy during the shift which would make optimal decision for the present also in regard of the future. Despite this, the analysis serves as a foundational framework to inform decisions aimed at fostering a more sustainable and livable environment through optimized road usage and reduced vehicle emissions.

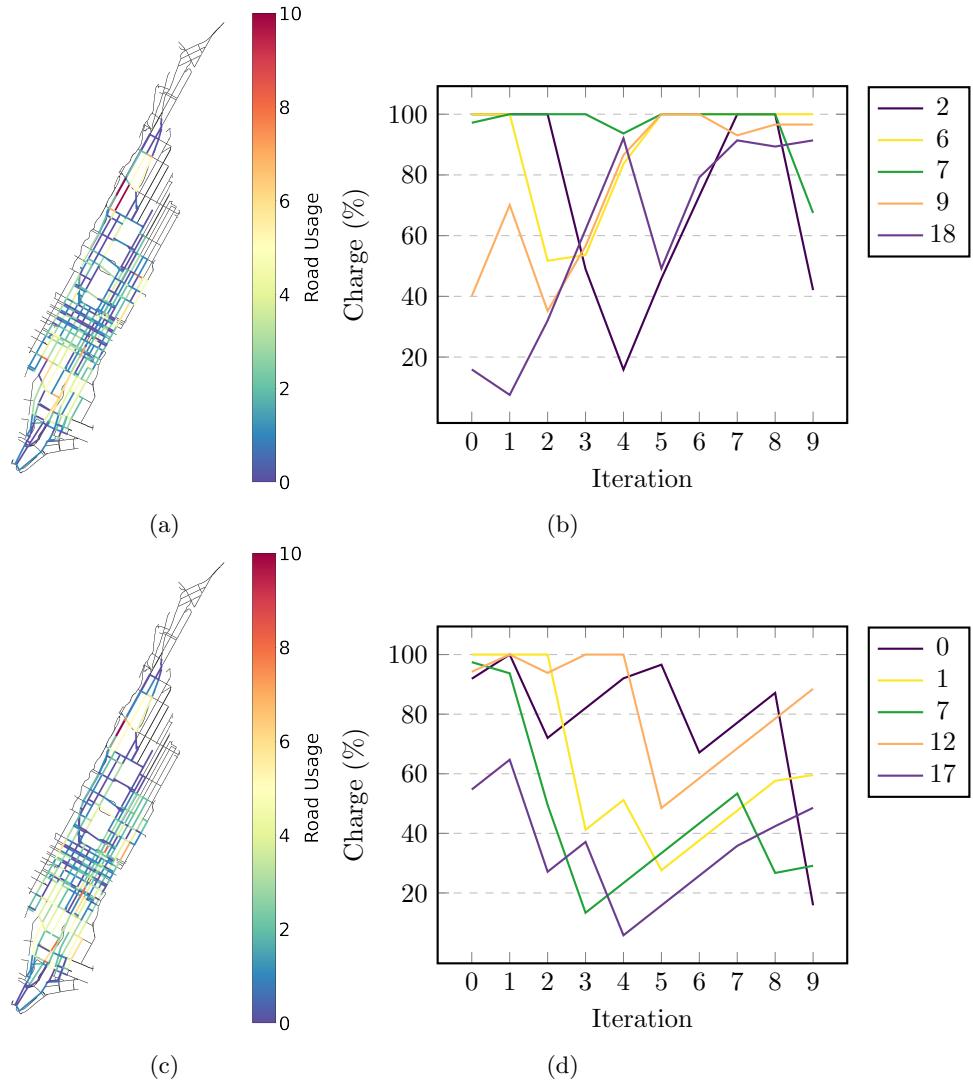


Figure 4.8: Road Usage and State of Charge of an exemplary Set of Vehicles for the Simulation 2 and 3 with Rebalancing. (a) and (b) refer to Simulation 2, while (c) and (d) depict Simulation 3.

Chapter 5

Model Predictive Control of an ATS

Motivated by the insights obtained in the previous chapter, which suggested that making decisions in the present have a great impact on the future, a novel model for an ATSs is proposed which, coupled with an ad-hoc designed MPC, aims at optimizing long-term outcomes by dynamically adjusting strategies in response to evolving circumstances. Furthermore, in order to improve the real-time performance of the controller, graph transformation systems are used to derive a more abstract version of the road network, named Reduced Connectivity Schema (RCS), to reduce the complexity of the model. The chapter is organized as follows: In Section 5.1, a novel model for the system is developed. Subsequently, Section 5.2 defines the MPC problem and includes a thorough discussion on the terminal set and cost function, which are essential to prove stability. In order to achieve this goal, the feasibility of the terminal and of the states set are demonstrated. Section 5.3 introduces the set of GTS rules and their condition aiming at the creation of a dynamic, adaptive and abstract representation of the road network. Finally, in Section 5.4, a simulation is proposed to evaluate the overall performance of the system.

5.1 Linear-Discrete Time Model

Consider a transportation network composed of $|\mathcal{V}|$ stations connected by a set of $|\mathcal{N}|$ roads and $|\mathcal{A}|$ multi-occupancy, goods-carrying vehicles. Within this context, customers are assumed to request rides only from the abovementioned stations.

Similarly to the model described in Chapter 4.1, other than carrying goods or people, vehicle are expected to *(i)* serve clients from one station to another and *(ii)* reach locations in \mathcal{V} in order to avoid system imbalance. For the rest of the section, a generic road $\langle i, j \rangle \in \mathcal{N}$ will be considered.

Let $d_{ij} \in \mathcal{R}_{\geq 0}$ be the total length of a road $\langle i, j \rangle$. Let $v_{ij}^a(t) \in \{0, 1\}$ indicate

whether a transporting vehicle $a \in \mathcal{A}$ is moving from station i to station j ($v_{ij}^a(t) = 1$) and, likewise, $w_{ij}^a \in \{0, 1\}$ whether an empty vehicle is moving from i to j ($w_{ij}^a(t) = 1$). Let $V_{ij}(t) \in \{x \in \mathbb{N}_0 : x \leq |\mathcal{A}|\}$ being the total number of vehicles currently circulating on the street $\langle i, j \rangle$, this can be trivially expressed as in (5.1).

$$V_{ij}(t) = \sum_{a \in \mathcal{A}} v_{ij}^a(t) + w_{ij}^a(t) \quad (5.1)$$

When a vehicle is in transit, it is essential to monitor the anticipated duration until it reaches its destination. In order to do so, let's assume the road $\langle i, j \rangle$ to have a speed limit $l_{i,j}$. Accordingly, since the system is dealing with fully autonomous vehicles, in a typical driving scenario, one can safely assume this to be the cruising speed as well. In other words, one can assume the vehicles to be driving with a speed $l_{i,j}$ over $\langle i, j \rangle$ in normal road conditions. Motivated by safety concerns, the cruising speed cannot be consistently maintained at $l_{i,j}$ due to various factors. These factors may include road conditions, weather conditions, traffic density, or any unforeseen circumstances. Therefore, the actual cruising speed during the journey over road $\langle i, j \rangle$ may vary based on these dynamic elements, ensuring that the autonomous vehicles can adapt to changing conditions and prioritize safety over a fixed cruising speed. One factor which is directly controllable is the traffic density. Therefore, taking inspiration from the BPR model (Fig 3.1), one can approximate the cruising speed according to the amount of vehicles on the road. More specifically, one can modify (3.2) to reflect this condition, and therefore rewriting it as

$$s_{ij}(V_{ij}) = \begin{cases} l_{ij} & \text{if } V_{ij} \in [0, V_{ij}^{th}] \\ l_{ij} - b \cdot (V_{ij}^{th} - V_{ij}) & \text{if } V_{ij} \in [V_{ij}^{th}, V_{ij}^{max}] \\ 0 & \text{if } V_{ij} \geq V_{ij}^{max} \end{cases} \quad (5.2)$$

with $b = \frac{l_{ij} - \epsilon}{V_{ij}^{th} - V_{ij}^{max}}$ and $\epsilon \in (0, 1)$

From the definition, it follows that s_{ij} is defined in the following domain $s_{ij} : \mathbb{N}_0 \rightarrow \mathbb{R}_{\geq eps}$. An example can be seen in Fig 5.1. The last case, i.e. when $V_{ij} \geq V_{ij}^{max}$, is inspired from the congestion model used in Section 4.1 and in this case is modeling stale traffic. Intuitively, if too many cars are on the road, these will not be able to circulate with a high speed and eventually stop-and-go traffic will be created. In this case, vehicles will be modelled as circulating at a very low speed.

This allows to effectively track the position of the vehicle in terms of time as well. However, the position should be tracked only if the vehicle is currently driving on the street. Therefore, the equation propagating the position of a over

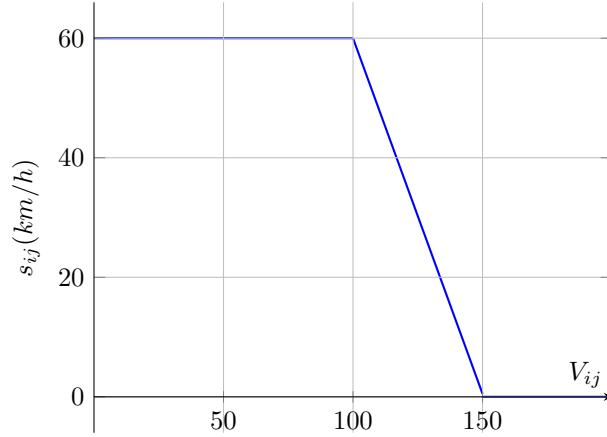


Figure 5.1: Cruising speed in function of traffic density. In this example, $l_{ij} = 60 \text{ km/h}$, $V_{ij}^{th} = 100$, $V_{ij}^{max} = 150$ and $\epsilon = 0.5$.

$\langle i, j \rangle$ must take care of this.

$$x_{ij}^a(t + \delta) = \begin{cases} x_{ij}^a(t) + s_{ij}(V_{ij}(t)) \cdot \delta & \text{if } v_{ij}^a(t) + w_{ij}^a(t) = 1 \\ 0 & \text{if } v_{ij}^a(t) + w_{ij}^a(t) = 0 \\ 0 & \text{if } i = j \end{cases} \quad (5.3)$$

As mentioned above, (5.3) tracks the position of a moving vehicle, i.e. takes care of the state of moving vehicles, and resets it in case a vehicle is not moving anymore.

In addition, the model also needs to track the number of vehicles currently stationed in i . This is achieved by introducing an additional variable $f_i^a(t)$, which indicates whether a vehicle arrived at a station i . It is defined as following.

$$f_i^a(t) = \begin{cases} 1 & \text{if } x_{ji}^a(t) \geq d_{ji} \\ 0 & \text{otherwise} \end{cases} \quad (5.4)$$

This variable can be propagated by introducing two more variables, i.e. l_{ji}^a and k_{ji}^a , to indicate whether a vehicle has departed or arrived at a station i , respectively. As a result, f_i^a is propagated as follows.

$$f_i^a(t + 1) = f_i^a(t) + \sum_{j \in \mathcal{V}} (k_{ji}^a(t) - l_{ji}^a(t)) \quad (5.5)$$

As an additional benefit, (5.5) also ensures that a vehicle performs an action on the road $\langle i, j \rangle$ only when stationed at i .

The variables l_{ji}^a and k_{ji}^a are binary variables as well and their definition is as

follows:

$$l_{ji}^a(t) = \begin{cases} 1 & \text{if } x_{ji}^a(t) \leq d_{ji} \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

$$k_{ji}^a(t) = \begin{cases} 1 & \text{if } x_{ji}^a(t) \geq d_{ji} \\ 0 & \text{otherwise} \end{cases} \quad (5.7)$$

The propagation of k_{ji}^a follows a straightforward pattern: if the variable is zero at time t , it can become either one or zero at $t + 1$, representing the possibility of the vehicle reaching a station or not. However, if $k_{ji}^a(t) = 1$ at time t , then $k_{ji}^a(t + 1) = 0$, indicating a reset. The variable l_{ji}^a is not propagated, however its definition depends on time. Initially, at $t = 0$, l_{ji}^a is set to zero, implying no vehicle can start at time zero. At instant $t + 1$, $l_{ji}^a = 1$ only if the vehicle has moved at time t , but was stationed at time $t - 1$, otherwise must remain zero¹. The definition in (5.7) imposes that the vehicles must stop if they reached a station. This is translated in (5.8) and (5.9)

$$w_{ij}^a(t + 1) = \neg v_{ij}^a(t) \wedge \neg k_{ji}^a(t) \wedge \neg v_{ij}^a(t) \quad (5.8)$$

$$v_{ij}^a(t + 1) = \neg w_{ij}^a(t) \wedge \neg k_{ji}^a(t) \wedge \neg w_{ij}^a(t) \quad (5.9)$$

This formulation leverages the fact that all the above variables are binary. In simple words, this is ensuring that a vehicle can only be rebalancing or driving if it was performing the same action on the previous instance or if it has not arrived at a station yet. In addition, it also insures that if the vehicle reached a station, then it would stop moving, i.e. $v_{ij}^a(t) = w_{ij}^a(t) = 0$ ².

Crucially, for the abovementioned constraint to work, the vehicles can not perform the aforementioned actions, i.e. waiting, routing and rebalancing, at the same time. Furthermore, the vehicles can not perform actions on multiple stations or roads. Therefore, the following constraint is necessary.

$$\sum_{i \in \mathcal{V}} (f_i^a(t + 1) + \sum_{j \in \mathcal{V}} v_{ij}^a(t) + \sum_{j \in \mathcal{V}} w_{ij}^a(t)) = 1 \quad (5.10)$$

This also implies that, if a vehicle is stationed at a station i , it can not have a position anywhere else.

$$\sum_{i \in \mathcal{V}} (f_i^a(t) + \mathbb{1}_{x_{ji}^a(t) \neq 0}) = 1 \quad (5.11)$$

In other words, if a vehicle is travelling through an edge $\langle ji \rangle$, it can only be stationed at i if it finished travelling.

If, on the one hand, it is interesting to know which station is currently hosting

¹For a more comprehensive analysis and explanation, please refer to Section C.1.1 and Section C.1.2.

²Please refer to Section C.1.3 for the derivation of these two constraints.

which vehicle, it is might also be important to know the amount of vehicles currently stationing in the system. This can be achieved by flipping the idea from (5.1) and therefore calculating how many vehicles out of the total number is currently not driving.

$$F(t) = |\mathcal{A}| - \sum_{(i,j) \in \mathcal{E}} V_{i,j}(t) \quad (5.12)$$

Denoted by $p_{ij}(t)$ and $g_{ij}(t)$ are the transportation and goods delivery requests respectively starting from station i and headed to j . Let $o_{ij}^p(t)$ and $o_{ij}^g(t)$ denoting the number of outstanding requests from i to j for people or goods, respectively, one can describe its propagation as follows:

$$\begin{aligned} o_{ij}^p(t+1) &= o_{ij}^p(t) + p_{ij}(t) - \sum_{a \in \mathcal{A}} P_a \cdot v_{ij}^a(t) \\ o_{ij}^g(t+1) &= o_{ij}^g(t) + g_{ij}(t) - \sum_{a \in \mathcal{A}} G_a \cdot v_{ij}^a(t) \end{aligned} \quad (5.13)$$

It follows then that vehicles can not transport more than what requested, i.e.:

$$\begin{aligned} P_a \cdot v_{ij}^a(t) &\leq o_{ij}^p(t) + p_{ij}(t) \forall a \in \mathcal{A} \\ G_a \cdot v_{ij}^a(t) &\leq o_{ij}^g(t) + g_{ij}(t) \forall a \in \mathcal{A} \end{aligned} \quad (5.14)$$

Accordingly, vehicles can rebalance only if there are still requests to be served.

$$w_{ij}^a(t) \leq \sum_{i,j \in \mathcal{E}} (o_{ij}^p(t-1) + o_{ij}^g(t)) \quad \forall a \in \mathcal{A} \quad (5.15)$$

At this point, the definition of the control and the state of the system is complete. More specifically, the state of the system is described by the number outstanding requests, the position of each moving vehicle $x_{ij}(t)$ and the position of each idle vehicle $f_i^a(t)$. Let the vector $\mathbf{x}(t)$ be the column vector created by reshaping and concatenating o_{ij}^p , o_{ij}^g , x_{ij}^a , f_i^a , V_{ij} , k_{ij}^a and l_{ij}^a , the set of feasible states \mathcal{X} is defined as follows.

$$\mathcal{X} := \left\{ x \mid \begin{array}{l} o_{ij}^p \in (\mathbb{N}_0)^{|\mathcal{N}|}, o_{ij}^g \in (\mathbb{N}_0)^{|\mathcal{N}|}, \\ k_{ij}^a, l_{ij}^a, f_i^a \in \{0, 1\}^{|\mathcal{A}||\mathcal{N}|} \\ x_{ij}^a \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{N}|}, V_{ij} \in \{a \in \mathbb{N}_0 : a \leq |\mathcal{A}|\}^{|\mathcal{N}|} \end{array} \right\} \quad (5.16)$$

where $x = [o_{ij}^p, o_{ij}^g, x_{ij}^a, f_i^a, V_{ij}, k_{ij}^a, l_{ij}^a]^T$

Note that $o_{ii}^p = 0$ and $o_{ii}^g = 0$.

Similarly, considering the control inputs $v_{ij}^a(t)$ and $w_{ij}^a(t)$, one can derive the

set of feasible control set $\mathcal{U}(t)$ as follows.

$$\mathcal{U}(t) := \left\{ u = [v_{ij}^a, w_{ij}^a]^T \middle| \begin{array}{l} v_{ij}^a \in \{0, 1\}^{|\mathcal{A}| |\mathcal{N}|}, v_{ii}^a = 0 \\ w_{ij}^a \in \{0, 1\}^{|\mathcal{A}| |\mathcal{N}|}, w_{ii}^a = 0 \\ (5.10), (5.14), (5.15) \end{array} \right\} \quad (5.17)$$

Since (5.8) and (5.9) depend on time, $\mathcal{U}(t)$ is also time-dependent.

Thanks to these formulations, the system can be written as a linear time-dependent system of the form

$$x(t+1) = Ax + Bu(t) \quad (5.18)$$

where $x \in \mathcal{X}$, $u(t) \in \mathcal{U}(t)$ and A and B are the matrix associated to the coefficients of (5.3), (5.5) and (5.13).

5.1.1 Objectives

Clearly, the main objective is to serve all the outstanding requests.

$$J_1(x(t)) = \sum_{i,j \in \mathcal{V}} (o_{ij}^p(t) + o_{ij}^g(t)) \quad (5.19)$$

However, the performance of the system can be optimized if other aspects are considered as well.

Since the speed of the vehicles, according to the model, is determined by the number of vehicles on the street, limiting it will improve performance. Vehicles serving requests are responsible for the main objective and their number is directly proportional to the number of outstanding requests. On the other hand, rebalancing vehicles, while help reducing the number of requests, they also contribute to congestions. Therefore, by reducing the number of free vehicles, the speed of the transporting ones is optimized as a result.

$$J_2(x(t)) = \sum_{i,j \in \mathcal{V}} w_{ij}^a(t) \quad (5.20)$$

Therefore, the final stage cost will be a combination of the two described above.

$$I(x(t)) = J_1(x(t)) + \lambda \cdot J_2(x(t)) \quad (5.21)$$

where λ is the weight used to determine the influence the rebalancing must have in the system.

5.1.2 Model Evaluation

Some comments are in order. Requests are treated considering customers and goods as single entities. More specifically, if a group consisting of three customers reach a station, this is treated as three individual requests, therefore $p_{ij}(t) = 3$. While this simplifies the model description and reflects the requirements for goods delivery, it would create an inconvenient scenario for traveling customers. As a matter of fact, it is not hard to envision situations where a group of travelers would like to travel together. However, it is out of the scope of this work to treat this scenario. Similarly, this model does not take into account scenarios where $o_{ij}^p < \min_{a \in \mathcal{A}}(P_a)$. This can be tackled by tracking the capacity that vehicles have left at their disposal.

The system described in (5.18) implicitly assumes the customer arrival to be known a priori. In other words, this is not treated as noise, but rather as a known quantity.

$$x(t+1) = Ax(t) + Bu(t) + w(t) \quad (5.22)$$

where $x(t) \in \mathcal{X}$, $u(t) \in \mathcal{U}(t)$, $w(t) = [p_{ij}(t) \ g_{ij}(t) \ 0 \ 0 \ 0]^T$ and A and B are the matrix associated to the coefficients of (5.3), (5.4) and (5.13).

(5.22) treats customer arrival as noise, or disturbance, which makes reasoning on the system considerably harder. Several techniques ([6], [28]) have been proposed to deal with this type of MPC systems, i.e. Robust MPC. Alternatively, some work propose solutions to estimate this quantity using Deep Learning techniques ([3], [9]). However, the analysis of those techniques and their performance in this scenario is out of the scope of this work. Similarly to Section 4.1, the model must take into account that stations do not have unlimited amount of parking spots. As a consequence, the number of vehicles stationed must be limited accordingly and this is achieved as follows.

$$\sum_{a \in \mathcal{A}} f_i^a(t) \leq C_i \quad (5.23)$$

where C_i indicates the number of parking spots available at i .

Furthermore, (5.2) imposes an implicit capacity limit on the streets. If $\epsilon = 0$ and the limit V_{ij}^{max} was exceeded, than the speed would also be zero. In this condition, the time would not be within the defined bounds, i.e. a constraint is violated.

Furthermore, one can also calculate the required time $T_{ij}(t)$ based on the speed approximation, as shown in (5.24).

$$T_{ij}^a(t) = \begin{cases} \frac{d_{ij} - x_{ij}^a(t)}{s_{ij}(V_{ij}(t))} & \text{if } x_{ij}^a(t) \in (0, d_{ij}) \\ 0 & \text{otherwise} \end{cases} \quad (5.24)$$

Since this function fails to satisfy the properties of homogeneity and additivity, it is not linear. Consequently, incorporating this function in the model will result in a non-linear model as well. Therefore, it has been decided to exclude it from the model.

5.2 Problem Formulation

The MPC for the ATS problem is formulated as follows.

Definition 5.1: (*MPC for ATS*)

Given $x(t) \in \mathcal{X}$, determine the controls $u(t), \dots, u(t+N)$ according to the following optimization problem.

$$\begin{aligned} \min_{u(t), \dots, u(t+N)} \quad & J_f(x(N)) + \sum_{t=0}^{N-1} I(x(t)) \\ \text{s.t.} \quad & x(t+1) = Ax(t) + Bu(t) \\ & x(t) \in \mathcal{X}, \quad u(t) \in \mathcal{U} \\ & x(N) \in \mathcal{X}_f \end{aligned} \quad (5.25)$$

where $J_f(x(t+N))$ is the terminal cost function and \mathcal{X}_f is the terminal set.

As the main goal is to prove the stability of (5.25), a proper definition of the final cost and terminal set will facilitate this objective. The strategy used to prove stability is the one described in Section 2.2.1. For this purpose, the terminal set is required to be defined around an equilibrium point x_E . A good candidate for the equilibrium point in such systems can be found by observing that the system remains at a equilibrium whenever no more requests arrive and there are no more outstanding requests within the system. In other words, when

$$\begin{aligned} p_{ij}(t) &= 0 \\ g_{ij}(t) &= 0 \quad \forall i, j \in \mathcal{V} \\ o_{ij}^p(t) &= 0 \\ o_{ij}^g(t) &= 0 \end{aligned}$$

By (5.13), this implies that, eventually, no more vehicles will transport goods or people anymore. Furthermore, one can also conclude that vehicles will also stop rebalancing themselves. Therefore, o_{ij}, x_{ij}^a and V_{ij} all assume value zero. However, there is no way of knowing exactly where vehicles are going to be stationed. Upon inspection, however, one can conclude that vehicles must be stationed somewhere, since they are not driving. In other words, f_i^a could assume any value in $\{0, 1\}^{|\mathcal{A}||\mathcal{V}|}$ except $\{0\}^{|\mathcal{A}||\mathcal{V}|}$. Furthermore, at equilibrium, all the vehicles are stationed. That means, the set of possible values for f_i^a is further limited accordingly. To define this set, let's consider a function n :

$\mathcal{D}_{f_i^a} \rightarrow N$, where $\mathcal{D}'_{f_i^a} = \{0, 1\}^{|\mathcal{A}||\mathcal{V}|} \setminus \{0\}^{|\mathcal{A}||\mathcal{V}|}$, which sums the number of 1s in the set. With this addition, one can fully define the set of values of f_i^a at equilibrium.

$$\mathcal{D}_{f_i^a} := \left\{ x \mid x \in \{0, 1\}^{|\mathcal{A}||\mathcal{V}|}, n(x) = |\mathcal{A}| \right\} \quad (5.26)$$

While the set is indeed smaller, it is still quite hard to exactly pin-point a specific equilibrium point, as already mentioned above. The difficulty arises because, out of all the elements in (5.26), one can not directly identify the one which describes the system fully without knowing how the system progressed in time. However, one can conclude that the equilibrium points are all elements of the set described in (5.27).

$$\mathcal{E} := \left\{ e = [o_{ij}^p, o_{ij}^g, x_{ij}^a, f_i^a, V_{ij}]^T \middle| \begin{array}{l} o_{ij}^p \in \{0\}^{|\mathcal{N}||\mathcal{A}|}, o_{ij}^g \in \{0\}^{|\mathcal{N}||\mathcal{A}|} \\ k_{ij}^a, l_{ij}^a \in \{0\}, f_i^a \in \mathcal{D}_{f_i^a} \\ x_{ij}^a \in \{0\}^{|\mathcal{N}||\mathcal{A}|}, V_{ij} \in \{0\}^{|\mathcal{N}|} \end{array} \right\} \quad (5.27)$$

While this could be considered as a candidate for the terminal set \mathcal{X}_f , it turns out to be too restrictive. This is due to the fact that it would require all vehicles to be stationed and, therefore, not moving. Some of the conditions, however, can be relaxed by considering some implicit assumptions made during the development of the model. Mainly, vehicles are assumed to be perfect and to never break, therefore, one can consider a request to be satisfied the moment it has been picked up. As a result, while the number of requests is still zero, the conditions for x_{ij}^a , f_i^a , k_{ij}^a and V_{ij} can be relaxed. Furthermore, (5.5) is slightly changed so that vehicles do not leave the station as they arrive. This is achieved by simply setting $\sum_{j \in \mathcal{V}} (l_{ji}^a(N-1)) = 0$.

$$f_i^a(N) = f_i^a(N-1) + \sum_{j \in \mathcal{V}} (k_{ji}^a(N-1)) \quad (5.28)$$

$$\mathcal{X}_f := \left\{ x_f \mid \begin{array}{l} o_{ij}^p \in \{0\}^{|\mathcal{N}||\mathcal{A}|}, o_{ij}^g \in \{0\}^{|\mathcal{N}||\mathcal{A}|} \\ k_{ij}^a, f_i^a \in \{0, 1\}^{|\mathcal{N}||\mathcal{A}|}, l_{ij}^a \in \{0\}^{|\mathcal{N}||\mathcal{A}|} \\ x_{ij}^a \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{N}|}, V_{ij} \in \{a \in \mathbb{N}_0 : a \leq |\mathcal{A}|\}^{|\mathcal{N}|} \end{array} \right\} \quad (5.29)$$

where $x_f = [o_{ij}^p, o_{ij}^g, x_{ij}^a, f_i^a, V_{ij}, k_{ij}^a, l_{ij}^a]^T$

By construction, therefore, $\mathcal{E} \subset \mathcal{X}_f \subset \mathcal{X}$.

In addition, thanks to this definition, another desiderable conditions apply. Since $x_{ij}^a \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{N}|}$ and $V_{ij} \in \{a \in \mathbb{N}_0 : a \leq |\mathcal{A}|\}^{|\mathcal{N}|}$, empty vehicles can

also be navigating, which allows rebalancing, too. Moreover, since vehicles are not required to be stationed and the minimum requirement for a request to be satisfied is to be picked up, as long as the capacity of the vehicle stationed in i is enough to cover the request, than the number of outstanding requests is still zero.

$$\begin{aligned} \sum_{a \in \mathcal{A}} P_a \cdot f_i^a(t) &\geq p_{ij}(t) \\ \sum_{a \in \mathcal{A}} G_a \cdot f_i^a(t) &\geq g_{ij}(t) \quad \forall i, j \in \mathcal{V} \\ o_{ij}^p(t) &= 0, o_{ij}^g(t) = 0 \end{aligned} \tag{5.30}$$

Furthermore, at each time, it is required that the capacity of the vehicle leaving a station is equal to the new demand, as the number of outstanding requests is zero.

$$\begin{aligned} \sum_{a \in \mathcal{A}} P_a \cdot v_{ij}^a(t) &= o_{ij}^p(t) \\ \sum_{a \in \mathcal{A}} G_a \cdot v_{ij}^a(t) &= o_{ij}^g(t) \end{aligned} \tag{5.31}$$

Vehicles, however, can only leave a station if they are present at that station. On the same note, since there must be enough vehicles at every station to serve the requests, those must be rebalanced in such a way that, once a request arrives, a vehicle is ready to serve it.

Furthermore, as a vehicle leaves a station, another vehicle must take its place. Therefore, the follow relation must also hold.

$$\begin{aligned} P_a \cdot v_{ij}^a &\leq \sum_{j \in \mathcal{V}} \sum_{a' \in \mathcal{A} \setminus \{a\}} P_{a'} \cdot w_{ji}^{a'} \\ G_a \cdot v_{ij}^a &\leq \sum_{j \in \mathcal{V}} \sum_{a' \in \mathcal{A} \setminus \{a\}} G_{a'} \cdot w_{ji}^{a'} \end{aligned} \tag{5.32}$$

(5.32) is used to ensure that, if a vehicle leaves a station, it must be replaced by one or more vehicles with at least the same capacity. A remark is required. (5.31) and (5.32) indeed allow vehicles to be replaced and, therefore, potentially take care of the requests. However, in addition to the assumption made for (5.31), the the system will remain in \mathcal{X}_f only as long as the requests are made within a time interval corresponding to the time required to travel from any station containing a free vehicle to the station where the request is starting. In other words, this is a very particular case of extrogenous requests. Therefore, for the rest of this section, the external arriving requests will be considered as zero, i.e. $o_{ij}^p(t) = o_{ij}^g(t) = 0$, therefore treating the system as undisturbed. In this case, one can assume that, at the end of a long enough horizion, all the requests are served and no more transporting vehicles v_{ij}^a leaves a station and, likewise, since no more requests are going to arrive, then the number of new rebalancing vehicles w_{ij}^a should also be zero.

5.2.1 Controller Stability

As indicated in Section 2.2.1, the three main requirements for controller stability are (*i*) that the control law is feasible considering \mathcal{U} , (*ii*) the terminal cost function is Lyapunov and (*iii*) the set \mathcal{X}_f remains feasible.

The control law that must be designed is a function mapping \mathcal{X}_f to a subset of \mathcal{U} , i.e. $\kappa_f : \mathcal{X}_f \rightarrow \mathcal{U}_{\kappa_f} \subseteq \mathcal{U}$. Due to (5.14) and (5.15), $\mathcal{U}_{\kappa_f} = \mathcal{U}$, which satisfies the first condition.

While it is usually difficult to construct a Lyapunov terminal cost function, since the system considered is undisturbed, one can construct a cost function satisfying the Lyapunov properties by observing the vehicles still in motion at the end of the horizon. Since all the requests are taken care of, new vehicles are not put in motion. Hence, all the vehicles still moving, will eventually come to a stop to a station.

In this scenario, two candidates can be determined. Firstly, if the traveling time T_{ij}^a has been included in the model, this can be used to determine a cost function $J_t(x)$, which can be proved to be Lyapunov³. Alternatively, an immediate candidate is the number of vehicles still moving in the system. The terminal cost function $J_v(x)$ is trivial to define.

$$J_v(x(N)) := \sum_{i,j \in \mathcal{V}} V_{ij} \quad (5.33)$$

With these assumptions proving $J_v(x)$ to be Lyapunov is straight forward.

Proposition 5.1: *Within the definition of \mathcal{X}_f , (5.33) is a Lyapunov Function in \mathcal{X}_f*

Proof. Considering an equilibrium point $x_{\mathcal{E}} \in \mathcal{E}$, three conditions must be met.

1. The function must be strictly positive except at $x_{\mathcal{E}}$, i.e.

$$\sum_{i,j \in \mathcal{V}} V_{ij}(t) > 0$$

This is indeed satisfied by definition of \mathcal{X}_f . More specifically, when the system is not at $x_{\mathcal{E}}$, then there are vehicles still moving, hence $\sum_{i,j \in \mathcal{V}} V_{ij} > 0$.

2. Secondly, the function must assume the value of zero at equilibrium. In other words, given the point $x_{\mathcal{E}}$

$$J_v(x_{\mathcal{E}}) = 0$$

At equilibrium, there are no vehicles moving. Clearly, the terminal cost function is zero.

³The definition and proof can be found in Section C.2.1

3. J_v must decrease $\forall x \in \mathcal{X}_f$.

$$J_v(x(k+1)) - J_v(x(k)) \leq 0$$

Let's consider a system with only one vehicle a still moving. If the vehicle is not rebalancing or transporting a customer, it can be considered at equilibrium, hence $\sum_{i,j \in \mathcal{N}} V_{ij} = 0$. J_v does not increase nor decrease with time, hence the condition is satisfied. If the vehicle is moving, since it can not move backwards, as time increases, by propagation of x_{ij}^a , its position always increases until $x_{ij}^a \geq d_{ij}$. During this time window, $\sum_{i,j \in \mathcal{N}} V_{ij} = 1$, eventually reaching the station j and therefore not being in motion anymore, hence the condition remains satisfied. For a system composed of more vehicles, regardless of whether on the same link or not, the same discussion applies. More specifically, since the number of moving vehicles can not increase, can only decrease or remaining the same, hence the condition remains satisfied.

In order to prove stability of (5.25), it is imperative to first demonstrate the feasibility of the state sets \mathcal{X} and \mathcal{X}_f ⁴.

For the rest of this section, the notation will be eased as follows.

$$\begin{aligned} x &= x(t) \\ u &= u(t) \\ x^+ &= x(t+1) \end{aligned}$$

Proposition 5.2: (*Feasibility of \mathcal{X}*) Let $x \in \mathcal{X}$ and $u \in \mathcal{U}(t)$, then $x_+ \in \mathcal{X}$

Proof. Let $x^+ = [o_{ij}^{p+}, o_{ij}^{g+}, x_{ij}^{a+}, f_i^{a+}, V_{ij}^+, g_{ij}^{a+}, p_{ij}^{a+}]^T$ and $u = [v_{ij}^a, w_{ij}^a]^T$. Since $u \in \mathcal{U}$, then (5.14) is satisfied, hence $o_{ij}^{p+}, o_{ij}^{g+} \in (\mathbb{N}_0)^{|\mathcal{V}|}$. Trivially, by construction V_{ij}^+ is also $\in \mathcal{X}$. Likewise, $g_{ij}^{a+}, p_{ij}^{a+} \in \{0, 1\}$ because of (5.7) and (5.6).

The condition $f_i^{a+} \in \{0, 1\}^{|\mathcal{A}||\mathcal{V}|}$ can be proven with the help of (5.5). At first vehicles are either present at a station i or not. If vehicles are not stationed, then no action can be taken due to (5.10) (if $u \in \mathcal{U}$, then it is satisfied), which implies no vehicle is departing or arriving at the station. Otherwise, if vehicles are indeed stationed, then $f_i^a = 1$. If a vehicle moves, i.e. if $w_{ij}^a(t) = 1$ or $v_{ij}^a(t) = 1$, because of the first condition of (5.3), then because of (5.6) and (5.7) being mutually exclusive, at the next step $f_i^{a+} = 0$. If, on the other hand, the vehicle is approaching the station, i.e. $f_i^a = 0$ and either $v_{ji}^a(t) = 1$ or $w_{ji}^a(t) = 1$, then the following enfolds. If $x_{ij}^a \leq d_{ij}$, $f_i^{a+} = 0$. Assuming $x_{ij}^a \geq d_{ij}$, then $g_{ij}^{a+} = 1$, due to (5.5), then either $f_i^{a+} = 1$ if $v_{ji}^a(t) = w_{ji}^a(t) = 0$, or $f_i^{a+} = 0$, if $v_{ji}^a(t) = 1$ or $w_{ji}^a(t) = 1$, which would imply $p_{ij}^{a+} = 1$.

$x_{ij}^{a+} \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{V}|}$ is proved by observing x_{ij}^a in (5.3). In case $v_{ij}^a(t) + w_{ij}^a(t) = 0$,

⁴Section C.2.2 contains the same proof for the sets including traveling time as well. The proof in this section works in with those definitions as well

the condition is satisfied, since $x_{ij}^{a+} = 0$. On the other hand, in case of $v_{ij}^a(t) + w_{ij}^a(t) = 1$, the condition is satisfied by definition of s_{ij} and $V_{ij}(t)$.

□

Proposition 5.3: (*Feasibility of \mathcal{X}_f*) Let $x \in \mathcal{X}_f$ and $\kappa_f(x) \in \mathcal{U}_{\kappa_f}(t)$, then $x_+ \in \mathcal{X}_f$

Proof. Let $x^+ = [o_{ij}^{p+}, o_{ij}^{g+}, x_{ij}^{a+}, f_i^{a+}, V_{ij}^+, g_{ij}^{a+}, p_{ij}^{a+}]^T$ and $\kappa_f(x) = [v_{ij}^a, w_{ij}^a]^T$. Since the system is treated as undisturbed, $o_{ij}^{p+} = o_{ij}^{g+} = 0$. Because $\kappa_f(x) \in \mathcal{U}_{\kappa_f}(t)$, then (5.31) is satisfied, implying there is no vehicle leaving the station to serve a request. By construction V_{ij}^+ is also $\in \mathcal{X}_f$. Likewise, $g_{ij}^{a+} \in \{0, 1\}$ because of (5.7).

If $w_{ij}^a = 0$, then nothing happens within the system, therefore all the vehicles remain at the station, then $f_i^{a+} \in \{0\}^{|\mathcal{V}||\mathcal{A}|} \forall i \in \mathcal{V}$. Consequently $x_{ij}^{a+} = 0$, i.e. $\in \mathbb{R}_{\geq 0}$.

If vehicles are in movement, i.e. $w_{ij}^a = 1$, a similar argument to the one proposed for the feasibility of \mathcal{X} can be made. Because of (5.5), $f_i^{a+} = 0$ and $x_{ij}^{a+} \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{V}|}$, by definition of s_{ij} and $V_{ij}(t)$. Eventually, as the vehicle approaches the station, $d_{ji} = x_{ji}^a(t)$, then $f_i^{a+} = 1$ due to (5.5) and $p_{ij}^{a+} = 0$. Since (5.10) is respected ($\kappa_f(x) \in \mathcal{U}_{\kappa_f}(t)$), then (5.11) is also respected, as $w_{ij}^{a+} = 0 \implies x_{ij}^{a+} = 0$. Condition (5.32) is always respected, assuming there is no new requests.

□

Proving that \mathcal{X} and \mathcal{X}_f are feasible provides several insights and assurances regarding the performance of the controller as well. First of all, it implies the controller can always find a feasible solution within these sets, given any initial condition and prediction horizon. This is crucial for ensuring that the controller can operate under various circumstances without violating any constraints. Furthermore, it suggests that the controller can adequately predict future system behavior and plan control actions to ensure that the system remains within safe operating limits. Additionally, it suggests that even in the presence of disturbances or uncertainties, the MPC controller can maintain feasibility by adjusting control actions within the given constraints, which contributes to the robustness of the control system.

More importantly, these can be used to prove the closed loop stability of the controller.

Proposition 5.4: (*Stability of 5.25*)

Given \mathcal{X}_f and J_f defined in (5.29) and (5.33), respectively, and let $\kappa_f : \mathcal{X} \rightarrow \mathcal{U}_{\kappa_f}$, then the controller defined in (5.25) is stable in the sense of Lyapunov.

Proof. To prove stability, one must (1) prove the recursive feasibility of \mathcal{X}_f and (2) that the optimal cost function J^* is Lyapunov .

1. As stated by Proposition 5.2, \mathcal{X} is feasible. Let $x \in \mathcal{X}$ and $[u_0^*, u_1^*, \dots, u_{N-1}^*]$ be an optimal control sequence calculated at x . At x^+ the control sequence $[u_0^*, u_1^*, \dots, \kappa_f(x_N^*)]$ is feasible. This is because $x_N \in \mathcal{X}_f$ and, therefore, $x_+ = Ax^* + B\kappa_f(x_N^*) \in \mathcal{X}_f$, since \mathcal{X}_f is feasible, as proven in Proposition 5.3. This proves recursive feasibility.
2. Given the optimal cost function

$$J^*(k) = J_f(x_N^*) + \sum_{i=0}^{N-1} I(x_i^*, u_i^*)$$

At $x(k+1) = x_1^*$, the following needs to be shown

$$J^*(k+1) \leq \tilde{J}(k)$$

where $\tilde{J}(k)$ is the candidate function and calculated at $\tilde{U} = \{u_1^*, u_2^*, \dots, \kappa_f(x_N^*)\}$. Therefore

$$\begin{aligned} J^*(k+1) &\leq \sum_{i=1}^{N-1} I(x_i^*, u_i^*) + I(x_N^*, \kappa_f(x_N^*)) + J_f(Ax_N^* + B\kappa_f(x_N^*)) \\ J^*(k+1) &\leq \sum_{i=1}^{N-1} I(x_i^*, u_i^*) + I(x_0^*, u_0^*) - I(x_0^*, u_0^*) + I(x_N^*, \kappa_f(x_N^*)) + J_f(Ax_N^* + B\kappa_f(x_N^*)) \\ J^*(k+1) &\leq \sum_{i=0}^{N-1} I(x_i^*, u_i^*) - I(x_0^*, u_0^*) + I(x_N^*, \kappa_f(x_N^*)) + J_f(Ax_N^* + B\kappa_f(x_N^*)) \\ \text{Since } J^*(k) &= J_f(x_N^*) + \sum_{i=0}^{N-1} I(x_i^*, u_i^*) \\ J^*(k+1) &\leq J^*(k) - I(x(k), u_0^*) + \underbrace{J_f(Ax_N^* + B\kappa_f(x_N^*)) + I(x_N^*, \kappa_f(x_N^*)) - J_f(x_N^*)}_{\leq 0 \text{ because } J_f \text{ is a Lyapunov function}} \\ \implies J^*(k+1) - J^*(k) &\leq -I(x(k), u_0^*) \quad I(x, u) > 0 \text{ for } x, u \neq 0 \end{aligned}$$

As a result, the optimal cost is a Lyapunov function.

Hence, one can conclude that the system is asymptotically stable. \square

5.3 Reduced Connectivity Schema

The Reduced Connectivity Schema (RCS) serves to introduce an additional layer of abstraction above the road network, aimed at minimizing the nodes and edges within the graph representation of the network. This abstraction is driven by the understanding that in optimization models, which is the case for the MPC in this work, a decrease in the number of variables facilitates the identification of an optimal control input set. The RCS enables more scalable solutions, particularly in large urban areas or networks with complex topology. It allows optimization algorithms to handle larger networks without sacrificing performance or requiring prohibitively high computational resources. Furthermore, simplifying the network representation through RCS can enhance the robustness of control strategies against uncertainties or disturbances in the system. With fewer variables to optimize, the control system may be more resilient to unexpected events such as sudden changes in traffic conditions or road closures. To achieve these objectives, the RCS must be dynamically constructed depending on how the system evolves. For this purpose, the RCS will be constructed using GTS. Formally, RCS is defined as follows.

Definition 5.2: *Reduced Connectivity Schema (RCS)*

Let $G = (\mathcal{V}, \mathcal{E}, \mathcal{L}^{\mathcal{V}}, \mathcal{L}^{\mathcal{E}}, s, t)$ be a labelled graph representing a road network of an urban environment. The Reduced Connectivity Schema (RCS) is itself a graph $H = (\mathcal{V}_h, \mathcal{E}_h, \mathcal{L}_h^{\mathcal{V}}, \mathcal{L}_h^{\mathcal{E}}, s_h, t_h)$, where $\mathcal{V}_h \subseteq \mathcal{V}$, $\mathcal{E}_h \subseteq \mathcal{E}$, $\mathcal{L}_h^{\mathcal{V}} \subseteq \mathcal{L}^{\mathcal{V}}$, $\mathcal{L}_h^{\mathcal{E}} \subseteq \mathcal{L}^{\mathcal{E}}$ and $s_h, t_h : \mathcal{E}_h \rightarrow \mathcal{V}_h$ are the source and target functions restricted to edges in \mathcal{E}_h , obtained according to a sequence of transformation rules \mathcal{T} .

Edges and nodes are labelled according to a simple schema. Let $\mathcal{L}^{\mathcal{V}}$ be the set of node labels, the latter are labelled as necessary, in case they represent a drop-off or a pick-up point, or as simple nodes. On the other hand, $\mathcal{L}^{\mathcal{E}}$ is the set of edge-labels and include all the distances from each node.

The transformation rules in the sequence \mathcal{T} are defined as follows.

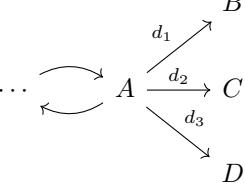
Rule 1: Important Node Reconnection

If a node is labelled as necessary, then all of its immediate connections are restored.

LHS	RHS
$\dots \xrightarrow{\text{...}} A$	$\dots \xrightarrow{\text{...}} A \xrightarrow{d_2} C \begin{cases} \xrightarrow{d_1} B \\ \xrightarrow{d_3} D \end{cases}$
NAC: A is a dropoff or pick-up node	

Rule 2: Simple Node Reconnection

If a node is labelled as simple, then three of its immediate connections are restored.

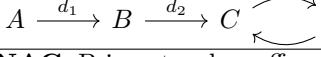
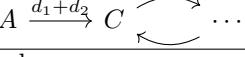
LHS	RHS
	
NAC: A is neither a dropoff nor pick-up node	

Critically, the difference between Rule 1 and Rule 2 lies in the fact that, while Rule 1 restores all connections, Rule 2 restores only a fixed amount, which in this case is three. These rules are meant to be applied to expand the connections between necessary with the aim of reducing the unnecessary edges from the original graph.

Although this would effectively reduce the number of edges and nodes, other rules can be used to further trim the unnecessary complexity.

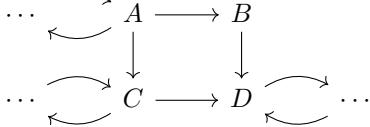
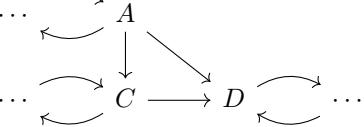
Rule 3: Straight Line Simplification

If three consecutive nodes (A, B, C) form a straight line and B has degree 2, remove node B.

LHS	RHS
	
NAC: B is not a dropoff nor pick-up node	

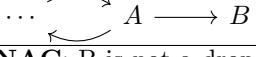
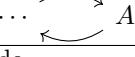
Rule 4: Node Removal based on Neighborhood

If there exist four nodes (A, B, C, D), each connected by exactly one edge, then any node with a degree of two can be removed⁵.

LHS	RHS
	
NAC: B is not a dropoff or pick-up node	

Rule 5: Dead-End Removal

If node B is labelled as simple and has a degree of one, then it can be removed.

LHS	RHS
	
NAC: B is not a dropoff or pick-up node	

⁵This rule is applied to the four vertices, for clarity the definition of only one case is shown

As a result of the application of the abovementioned rules, the RCS is constructed and can be used to manage the ATS.

Although these rules are highly context-dependent and, as a consequence, other rules could be easily constructed, the defined transformation rules offer a versatile framework for tailoring the construction of the RCS to specific urban environments and optimization objectives. By allowing customization based on factors such as traffic patterns, infrastructure layout, and control requirements, the RCS can effectively capture the unique characteristics of different transportation networks. This flexibility ensures that the schema remains adaptable to evolving urban landscapes and diverse application scenarios.

An important trade-off between complexity and performance must be considered. By carefully balancing the reduction of unnecessary edges and nodes with the preservation of essential connections, the RCS optimizes network representation to enhance computational performance without compromising control effectiveness. This balance ensures that the resulting schema remains efficient and scalable, capable of supporting real-time decision-making in dynamic urban environments. The balance, however, is difficult to find and, while it is usually determined through iterative refinement and validation, in practice it is difficult to establish a metric or an automatic process for it.

5.4 Use Case Analysis

The use case has been developed using the same data Section 4.4 with two main goals in mind. First, the theoretical insights about the stability of the controller need to be confirmed. The hypothesis is that, if the model has full knowledge of the future requests, then an optimal control law should be found such that minimizes the cost function defined in (5.19) and all the vehicles end at a station, i.e. no more moving vehicle is present. Secondly, the model's performance needs to be evaluated using real-world data to determine if it's feasible for application. A summary of the parameters and the performance of the simulations can be found in Table 5.1.

As shown in Table 5.1a, all parameters are uniform between the two simulations except for the length of the road, where, for Simulation 1, has been chosen a fixed value for the road length, while for Simulation 2 the real length is used. As a result, the performance of the simulations are slightly different. Although in both cases the same number of requests has been fulfilled, the simulation using the real network demonstrates slightly better results. This discrepancy stems from the fact that, while the average length remains consistent, specific roads exhibit significantly shorter distances. This means, in the same shift duration, vehicles might travel less to fulfill the requests and likewise they need to travel less to rebalance in order to fulfill future requests. Nevertheless, although the vehicles travel less, the number of served requests did not improve. As a matter of fact, the two simulation show that only around 20% of the total number of travelers and goods could be transported, which is the result of a short horizon. Due to the randomic nature of the requests, their destination

	1	2	3
AVs #	30	-	-
Horizon (h)	3	-	-
Threshold (km/h)	60	-	-
Requests	240	-	-
Road (km)	30	R	-

(a)

	1	2	3
ATT (%)	33	36	3
ART (%)	17	14	3
Required AVs	19	12	10
Carrying AVs	13	11	4
Rebalancing AVs	11	12	9

(b)

Table 5.1: Various information about the MPC Simulations. (a) includes the main details of the various MPC Simulations. “-” means the entry has not been changed from the previous one, while “R” indicates the real road length is used. All the vehicles can host two travelers and carry 10 pieces of cargo. (b) summarizes the performance of the controller during the simulations. The entry “ATT” and “ART” stand for “Average Transporting Time” and “Average Rebalancing Time”, respectively, and indicate the percentage of time which vehicles spend transporting requests, or rebalancing, on average, while the entry “Required Vehicles” represents the number of unique vehicles required during the whole shift, therefore it is not the sum between “Carrying” and “Rebalancing Vehicles”.

could be considerably far from where vehicle was stationed. Although a lot of solutions proposed either assume vehicles to be scattered around the area and/or the fleet to be infinitely large, in this work vehicles are stationed only within specified location within the road network, namely the depots. This is more representative of a real world scenario, especially if road parking is wished to be reduced. Furthermore, this situation also highlights problems which would not otherwise be presented, such as in this case. More specifically, most of the requests could not be fulfilled as their pickup point, or destination, is too far from where vehicles are stationed in terms of traveling times. It is worth noting that this problem is more present within large urban environment, such as NYC, and it is less visible within small centers. While, intuitively, this could be solved by extending the horizon, this would be deleterious for the performance of the controller, especially for large road network and real-time operations. In other words, extending the horizon is not a scalable solution. This further motivates the introduction of the RCS described in Section 5.3. This technique offers a means to further streamline the road infrastructure by identifying redundant or inefficient connections, as well as trimming the unnecessary connections which do not bring to a point of interest, thereby enhancing the overall efficiency and performance of the controller. By trimming the unrequired roads from the network, the number of variables is drastically reduced, which decreases the computational requirements of the optimization problem, allowing to consider a considerably longer horizon. In order to obtain the RCS, the rules included in Section 5.3 are applied according to the order and frequency described in Table 5.2. The results and the details of the simulation are once again summarized in Table 5.1, specifically as simulation number three.

Rule	Frequency
Rule 1	1
Rule 2	3
Rule 3	-
Rule 4	-
Rule 5	-

Table 5.2: Rule Application Sequence and Frequency for the RCS. “-” means that the rule has been applied until no more all the nodes are removed.

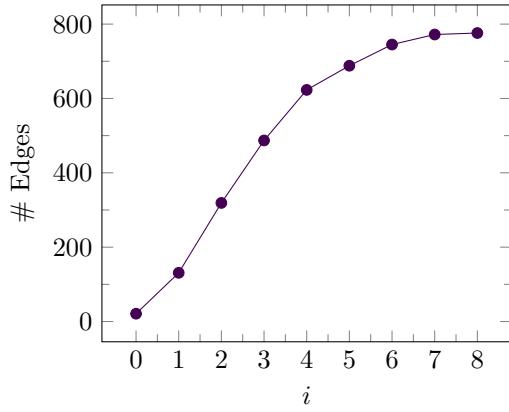


Figure 5.2: RCS Dimensions for Different Application of Rule 2. “ i ” refers to the amount of iterations applying Rule 2. All the other rules have been applied with the same parameters.

Considering the same requests, the RCS used results in a 60% reduction of the total dimension of the orginal road network, which signifies a notable reduction in complexity for both the model and the optimization problem. As a result, the time performance has also noticeably improved, while the overall performance did not decrease. As a matter of fact, not only the same number of requests have been served, but vehicles tend to travel less, since less roads are considered, and, on average, vehicles tend to be used less as well.

The main benefit of the RCS, however, is that its dimension is, contrary to the original representation, related to the how far the requests nodes are from the depots. In other words, thanks greatly to its flexibility, the RCS can be dynamically varied in size depending on this distance. In practice, this translates to the possibility of constructing subgraphs around the depots in order to efficiently satisfy the requests. This leads to expect two main effects. Firstly, if the requests fall within the graph radius, then they should be satisfied. Secondly, the computational time is expected be greatly reduced, as result of the reduced number of nodes and edges in the graph. In order to control the dimension of the RCS, a key parameter is the number of iterations for Rule 2. In the previ-

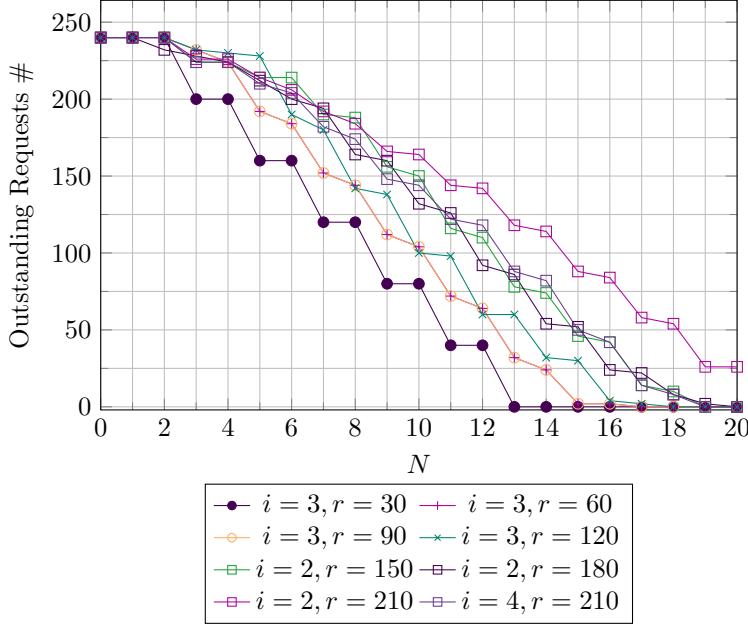


Figure 5.3: Decrease in Number of Requests Over Time. “ i ” refers to the amount of iterations applying Rule 2 and “ r ” is the requests radius in kilometers.

ous discussion, this rule has been applied for seven iterations. However, other values can be considered and their influenced is showed in Fig 5.2. Originally, the graph had 1200 edges and, as proven by Fig 5.2, it can be reduced to as little as 18%⁶. However, this number greatly depends on two main factors: (i) how close the requests pick-up and drop-off nodes are to the depots and (ii) the actual distance associated with the edge. For this use case, for simplicity, each edge has a length of 30 km, but, in a real-world application, various other considerations should be made in this regard. Nevertheless, thanks to the reduction in complexity allows to better study the performance of the controller in other settings. Fig 5.4 and Fig 5.3 depict the performance of the systems as the graph is changed as well. In order to obtain these results, only requests within a certain radius have been accepted. While it is understood that this is an artificial setting, it allows to obtain insights which would otherwise be difficult to capture. All the simulations use a fleet of 20 vehicles for an horizon of 10 hours.

Before delving into the details of those figures, however a particular detail must be noted. Since all the requests are artificially picked to be within the considered radius, it was expected the system to be able to serve most of the requests. This is indeed confirmed by Fig 5.3. This depicts how the number of outstanding requests decreases over the horizon time. As expected, as long as the

⁶Its evolution is depicted in Fig D.1

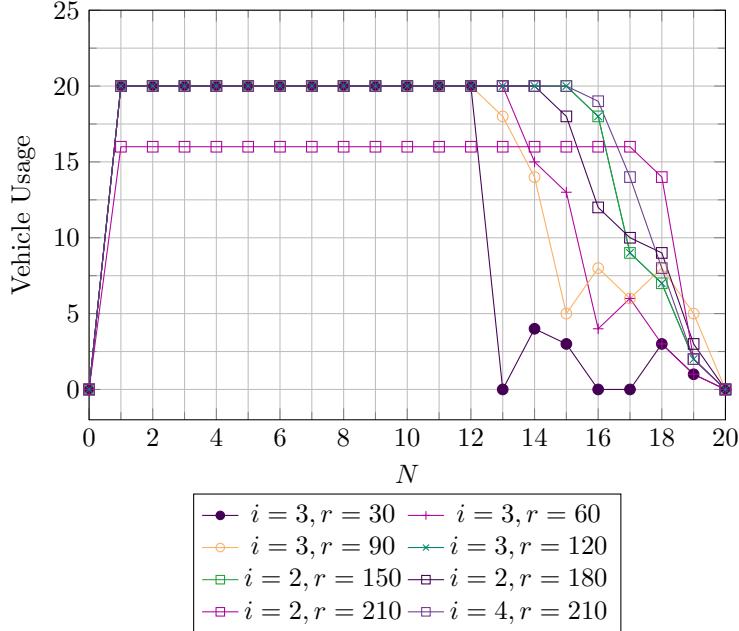


Figure 5.4: Evolution of The Vehicle Usage Over Time The vehicle usage is calculated as the sum of the vehicles on the street at time t . “ i ” refers to the amount of iterations applying Rule 2 and “ r ” is the requests radius in kilometers.

requests are within the RCS radius, the system is able to fulfill all of them. Some cases are faster than other due to the distance from the depot to the request locations. This is indeed the case in real-world situations as well. On the other hand, if a request falls outside the RCS radius, the the system is not able to fulfill it. This can also be inferred by Fig 5.4. For example, if requests are over 210 kilometers from the depot, the controller using RCS can not serve them. As a result, not all vehicles will be used, as in the case for $i = 2$ and $r = 210$ in Fig 5.4. Furtermore, by analyzing Fig 5.4, a particular trend emerges. As the radius increases, more vehicles are required to fulfill the requests and this can be deduced by the fact that, as r increase, so do the situations where all the vehicles are deployed. Note that in case of $i = 2$ and $r = 210$, all the deployable vehicles are indeed deployed for 80% of the time. If the radius is small, on the other hand, the full capacity of the fleet is deployed for less, as it is in the case of $i = 3$ and $r = 30$. Specifically in this case, vehicles are also used although, for the interval before, no vehicle was deployed.

While obtaining real-world data on goods transportation for direct comparison with the simulation may be challenging, using official NYC taxi data can provide insights into the actual performance of the system. According to the NYC government ([37]), considering a ten hour shift, taxis carry less than a passenger per day, more specifically, on average taxis carry 0.354 passengers per

taxi-hour⁷. Considering this as the baseline, one can derive a comparison with the proposed method in this chapter. Considering the same ten hour shift, following the same calculations, the controller is able to serve 0.83 passengers per taxi-hour, which is a 134% increase in transporting rate.

5.5 Conclusion

Although the controller showed promising results, both theoretically and on a practical use case, some considerations are still in order. Firstly, while the stability of the undisturbed controller is necessary, a real world application will likely deal with a noisy system, which, in this context, translates in various aspects. Intuitively, the first aspect is extrogenous requests which are not considered in advanced, which hypothetically, could still be served, albeit with a longer horizon. Secondly, in a more practical view, the model tracks the speed and the position of the vehicles only based on the presence of other vehicles on the road. In reality, other factors are influencing the vehicles' position. However, regarding the model, the primary limitation arguably comes from the model's exclusive focus on the controllable vehicles, neglecting the presence and behaviors of pedestrians and other road users, such as other vehicles or bikers. Integrating additional layers of complexity to account for the dynamic nature of urban environments, such as incorporating pedestrian behavior, road conditions, and interactions with other vehicles, could enhance the model's predictive capabilities and its utility in practical settings. Furthermore, other elements of viability and of the infrastructure should also be taken into account when dealing with urban environments. For example, vehicles such as ambulances, police cars and firefighters, as well as emergency vehicles in general, must be treated differently than other vehicles. For instance, correctly modeling and controlling traffic in situations of emergency could potentially help saving lives. Additionally, other less urgent, but more common, situations should also be modeled, such as interactions with railways and other means of transports. Notably, trains have strict deadlines which one must adhere to. While this could potentially be solved with the addition of time windows constraints, as already done in the previous chapter, other ways should be explored to create a more efficient model.

The simplistic nature of the model, however, does not necessarily imply limited applicability. On the contrary, although in simulation, the use case demonstrated a robust and performant controller, with the potential of being applied in a closed-loop scenario, where real vehicles can be efficiently guided towards making the best decisions to improve traffic flow in smart cities.

⁷See Section D.1 for more details

Chapter 6

Summary and Outlook

The escalating demands on transportation infrastructure in urban areas necessitate efficient traffic management. Beyond simply facilitating movement, it profoundly impacts economic productivity, environmental sustainability, and societal well-being. Effective traffic control mechanisms are crucial for mitigating the negative effects of an always more dynamic society, by optimizing vehicle flow, reducing congestion, and enhancing economic productivity and air quality. Additionally, they promote social inclusion and improve overall quality of life for urban residents. Motivated by these reasons, this work aims to laying the groundwork for a more efficient and powerful techniques to manage and control transportation systems, which include both people and goods mobility, in the context of smart cities. This work investigates the fundamental methods to achieve this ambitious goal, which are used to cover different aspects of mobility in dense urban environment. Firstly, in Chapter 4, a time-invariant model of an ATS is proposed, which takes into consideration the numerous aspects which influence the impact and performance of such a system. Furthermore, in line with the overall scope of the thesis, this model is developed with expansion in mind, namely considering future direction, such as different charging profile for the vehicles and different vehicles category for multi-cargo situations. Furthermore, while in this thesis it was mostly used for urban environment, the model is flexible enough to tackle different domains, such as logistics. This model is then used to reason about the most considered challenges for an ATS, which leads to the formulation of the CATSM problem (Section 4.3). Finally, the model and the CATSM are simulated using real-world data to prove their effectiveness and their performance on dense urban environment situations.

Motivated by the insights achieved in the previous chapter, specifically regarding the real-time controlling performance, a novel approach for controlling ATS is proposed in Chapter 5. While the use of MPC for controllining ATS has already been investigated in literature, the novelty of this chapter consists of a new linear discrete-time model, which gives the ability to control the speed and track the position of the vehicles in real time. This model is then used to formulate an MPC, which is proved to be stable using an ad-hoc terminal state

set and cost function. Furthermore, to further improve real-time performance, a completely new approach is proposed in Section 5.3 to dynamically reduce the complexity of the road network based on the drop-off and pick-up locations of the requests. Namely, by leveraging graph transformations systems, the RCS, acronym for reduced connectivity schema, is constructed, which is a subset of the original road network. This condensed representation can be utilized to enhance the efficiency of traffic control, improving both overall outcomes and computational efficiency.

Although multiple contributions have been made and the most fundamental basis have been established, being this its main goal, this thesis is to consider as the first milestone towards a complete ATS managing systems and multiple research directions are generated as a result. Firstly, while the solution considered in this work to solve the dispatching problem is naturally integrated in further problems' formulation, this highly impacts the computational performance of the solver. This motivates further investigations into other algorithms. An example approach involves devising an ad-hoc algorithm. Requests could be assigned to vehicles greedily based on various heuristics, such as proximity to the nearest neighbor or considering factors like vehicle status (e.g., state of charge, capacity) and current traveling path. It's important that assigned requests align with the vehicle's previous path (refer to Fig 4.2 for a visualization). Previous calculated paths can serve as input for this assignment process, if a step-wise solution for the main challenges is considered. On the same note, a similar consideration for rebalancing vehicles can be done. Furthermore, similarly to the next chapter, the introduction to the RCS in this approach would greatly improve the overall performance. Furthermore, the introduction of GTS to the vehicle-centric model could lead to completely new research directions. For example, the model could be revisited to be used, according to ad-hoc defined GTS rules, to prove liveness and safety property, such as the absence of incidents or that all requests can be served. Furthermore, GTS could also be used to derive precise requirements regarding elements of the infrastructures, such as the capacity of the road or the charging equipment. Infrastructure elements are also to be considered in this context, since they could be controlled by similar techniques in order to further improve the overall viability of the city. Chapter 5 is also source of potential research horizons. Firstly, the model proposed can be further expand to consider ride-sharing. This could be achieved by keeping track of the vehicle capacities as the system progress and, subsequently, obtain an optimal control law on how to distribute passengers and goods among the vehicles. Its effect should be studied, both in terms of stability and performance. Furthermore, in a smart city scenario, with V2X communication and smart infrastructure, the model can be further expanded to manage multiple factors which have not been considered so far. For example, traffic lights' routine could be defined in real-time based, based on the system's current state, in order to reduce traffic and further optimize road usage. While this is arguably not necessary in a condition of full autonomous vehicles navigating through the streets, this becomes a beneficial addition in condition of mixed traffic. This scenario also leads to consider the behaviour of non-vehicles circulating through

the streets to increase their safety.

This potential capillarity is also increased if more powerful model are considered as well, which go beyond linearity, and that consider other aspects of the vehicle, such as its dynamics. This could be used to make optimal local decisions, such as when to cross an intersection, based on the behaviour of other vehicles and their current state. On a macroscopic level, the same method can also be used to take into account other transportation networks, such as railway, to insure no deadlines are missed by safely controlling the speed of the vehicles accordingly. These latter aspects could be considered by applying GTS differently than what proposed in this work. More precisely, new rules could be defined such that the model is dynamically expanded to consider them as well. This discussion outlines the main benefit of the solution proposed in Chapter 5 lies in the fact that it allows for a more capillary integration within a smart city, when compared to other methods, since it can account for aspects which are outside of the ATS scope.

Furthermore, the application of GTS also offers multiple directions. An intriguing concept involves automating the design and adaptation of vehicles based on anticipated requests. By estimating the specific requirements for different locations, such as hospitals, vehicles can be configured accordingly. For instance, a vehicle designated for hospital services might need capabilities to *(i)* halt traffic due to priority, *(ii)* accommodate critically ill patients, and *(iii)* facilitate on-board medical staff. This approach allows for exploration of the "design space" for vehicle configurations. Similarly, infrastructure can be modified or created to align with these needs. For example, updating traffic lights with appropriate transmission equipment like V2X can facilitate efficient traffic control.

In summary, the insights gained from our discussion, along with the proposed solutions, represent a significant initial step in advancing modern urban landscapes towards safer and more efficient traffic, viability, and transportation systems. Although the challenges ahead are considerable, this thesis establishes sturdy groundwork for future advancements. By utilizing these foundations, we can work towards tangible enhancements in the quality of life for urban residents. This research sets the platform for ongoing innovation and enhancement, providing promising paths for further exploration and implementation. As we move forward, it's crucial to stay dedicated to the overarching objective of developing cities that are not only technologically sophisticated but also supportive of the well-being and prosperity of their inhabitants.

Bibliography

- [1] Ahmed Abdulaal, Mehmet H. Cintuglu, Shihab Asfour, and Osama A. Mohammed. Solving the multivariant ev routing problem incorporating v2g and g2v options. *IEEE Transactions on Transportation Electrification*, 3(1):238–248, 2017.
- [2] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017.
- [3] Bahman Askari, Tai Le Quy, and Eirini Ntoutsi. Taxi demand prediction using an lstm-based deep sequence model and points of interest. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1719–1724, 2020.
- [4] J. Beck, Patrick Prosser, and E. Selensky. Graph transformations for the vehicle routing and job shop scheduling problems. volume 2502, pages 60–74, 10 2002.
- [5] Luca Brodo. Autonomous mobility-on-demand: A novel framework to study amod in smart cities. Paper not yet submitted for publication, 2023.
- [6] Peter J. Campo and Manfred Morari. Robust model predictive control. *1987 American Control Conference*, pages 1021–1026, 1987.
- [7] Carlos Carrion and David Levinson. Value of travel time reliability: a review of current evidence. *Transportation Research Part A: Policy and Practice*, 46:720–741, 2012.
- [8] Andrea Carron, Francesco Seccamonte, Claudio Ruch, Emilio Frazzoli, and Melanie N. Zeilinger. Scalable model predictive control for autonomous mobility-on-demand systems. *IEEE Transactions on Control Systems Technology*, 29(2):635–644, 2021.
- [9] Kai Fung Chu, Albert Y.S. Lam, and Victor O.K. Li. Travel demand prediction using deep multi-scale convolutional lstm network. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1402–1407, 2018.

- [10] Juan de Lara, Hans Vangheluwe, and Pieter J. Mosterman. Modelling and analysis of traffic networks based on graph transformation. 2005.
- [11] B. Donovan and D. B. Work. New york city taxi trip data (2010-2013). <http://dx.doi.org/10.13012/J8PN93H8>, 2014.
- [12] Iain Dunning, Stuart Mitchell, and Michael O'Sullivan. Pulp: A linear programming toolkit for python. *Department of Engineering Science, The University of Auckland*, September 2011.
- [13] Aurélien Froger, Ola Jabali, Jorge E. Mendoza, and Gilbert Laporte. The electric vehicle routing problem with capacitated charging stations. *Transportation Science*, 56(2):460–482, 2022.
- [14] M. Fürst, J. Götz, M. Otto, et al. Automation of gearbox design. *Forsch Ingenieurwes*, 86:409–420, 2022.
- [15] German Aerospace Center (DLR). Nemo Bili - Mobility from the First to the Last Mile, Year of access. Accessed on: 04.01.2024.
- [16] J Gross and S Rudolph. Geometry and simulation modeling in design languages. *Aerospace Science and Technology*, 54:183–191, 2016.
- [17] J Gross and S Rudolph. Modeling graph-based satellite design languages. *Aerospace Science and Technology*, 49:63–72, 2016.
- [18] J Gross and S Rudolph. Rule-based spacecraft design space exploration and sensitivity analysis. *Aerospace Science and Technology*, 59:162–171, 2016.
- [19] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.
- [20] Kyoungseok Han, Tam W. Nguyen, and Kanghyun Nam. Battery energy management of autonomous electric vehicles using computationally inexpensive model predictive control. *Electronics*, 9(8), 2020.
- [21] Reiko Heckel. Graph transformation in a nutshell. *Electronic Notes in Theoretical Computer Science*, 148(1):187–198, 2006. Proceedings of the School of SegraVis Research Training Network on Foundations of Visual Modelling Techniques (FoVMT 2004).
- [22] Deborah A. Hennessy and Donald L. Wiesenthal. Traffic congestion, driver stress, and driver aggression. *Aggressive Behavior*, 25:409–423, 1999.
- [23] Andreas Holzapfel, Heinrich Kuhn, and Michael G. Sternbeck. Product allocation to different types of distribution center in retail logistics networks. *European Journal of Operational Research*, 264(3):948–966, 2018.
- [24] M. Hyland and H. S. Mahmassani. Dynamic autonomous vehicle fleet operations: optimization-based strategies to assign avs to immediate traveler demand requests. *Transportation Research Part C: Emerging Technologies*, 92:278–297, 2018.

- [25] Brian Kallehauge, Jesper Larsen, Oli Madsen, and Marius Solomon. *Vehicle Routing Problem with Time Windows*, pages 67–98. 03 2006.
- [26] S. R. Kancharla and G. Ramadurai. Electric vehicle routing problem with non-linear charging and load-dependent discharging. *Expert Systems With Applications*, 160:113714, 2020.
- [27] Hans-Joerg Kreowski, Renate Klempien-Hinrichs, and Sabine Kuske. Some essentials of graph transformation. *Studies in Computational Intelligence*, 25, 01 2006.
- [28] W. Langson, I. Chryssochoos, S.V. Raković, and D.Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.
- [29] G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43:408–416, 2009.
- [30] Chungmok Lee. An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. *Journal of the Operational Research Society*, 72:1–24, 03 2020.
- [31] Michael W. Levin, Kara M. Kockelman, Stephen D. Boyles, and Tianxin Li. A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application. *Computers, Environment and Urban Systems*, 64:373–383, 2017.
- [32] C Robin Lindsey and Erik T Verhoef. Congestion modelling. Technical report, Tinbergen Institute Discussion Paper, 1999.
- [33] Haining Liu, Ijaz Haider Naqvi, Fajia Li, Chengliang Liu, Neda Shafiei, Yulong Li, and Michael Pecht. An analytical model for the cc-cv charge of li-ion batteries with application to degradation analysis. *Journal of Energy Storage*, 29:101342, 2020.
- [34] Alejandro Montoya, Christelle Guéret, Jorge E. Mendoza, and Juan G. Villegas. The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, 103:87–110, 2017. Green Urban Transportation.
- [35] NetworkX Development Team. `networkx.algorithms.tree.mst.minimum_spanning_edges`. NetworkX 2.6.2 documentation, 2021. Retrieved 2024-02-11.
- [36] Heinz Neuburger. The economics of heavily congested roads. *Transportation Research*, 5(4):283–293, 1971.
- [37] New York City. Taxi of tomorrow: New york city's taxi history. https://www.nyc.gov/html/totweb/taxioftomorrow_history-links_time2.html#:~:text>New%20York%20now%20has%20only,almost%201%2C000%2C000%20passengers%20a%20day., Date of Access. Accessed: February 29, 2024.

- [38] New York City Mayor's Office of Food Policy. GetFoodNYC historical data. <https://www.nyc.gov/site/foodpolicy/reports-and-data/getfood-historical-data.page>. Accessed: February 12, 2024.
- [39] Zi-Hao Nie, Qiang Yang, En Zhang, Dong Liu, and Jun Zhang. Ant colony optimization for electric vehicle routing problem with capacity and charging time constraints. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 480–485, 2022.
- [40] Bureau of Public Roads. Traffic assignment manual. Technical report, U.S. Dept. of Commerce, Urban Planning Division, 1964.
- [41] P. Pant and R. M. Harrison. Estimation of the contribution of road traffic emissions to particulate matter concentrations from field measurements: a review. *Atmospheric Environment*, 77:78–97, 2013.
- [42] Marco Pavone, Stephen L Smith, Emilio Frazzoli, and Daniela Rus. Robotic load balancing for mobility-on-demand systems. *The International Journal of Robotics Research*, 31(7):839–854, 2012.
- [43] Mark P.H. Raadsen, Michiel C.J. Bliemer, and Michael G.H. Bell. Introducing pattern graph rewriting in novel spatial aggregation procedures for a class of traffic assignment models. 2016-12-01.
- [44] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Pub., 2020.
- [45] Gabriele Taentzer Reiko Heckel. Graph transformation for software engineers. 01 2020.
- [46] Federico Rossi, Rick g, and Marco Pavone. Autonomous vehicle routing in congested road networks. *CoRR*, abs/1603.00939, 2016.
- [47] Grzegorz Rozenberg. Handbook of graph grammars and computing by graph transformation. 01 1997.
- [48] Mauro Salazar, Matthew Tsao, Izabel Aguiar, Maximilian Schiffer, and Marco Pavone. A congestion-aware routing scheme for autonomous mobility-on-demand systems. pages 3040–3046, 06 2019.
- [49] David Schrank, Bill Eisele, and Tim Lomax. Texas transportation institute 2012 urban mobility report. Technical report, Texas Transportation Institute, Texas A&M University, College Station, TX, 2012.
- [50] Dirk Schumacher, Jan Ooms, Bakhtiyor Yapparov, and John O. Hansson. *rcbc: COIN CBC MILP Solver Bindings*, 2022. R package version 0.1.0.9001.

- [51] Stephen L. Smith, Marco Pavone, Mac Schwager, Emilio Frazzoli, and Daniela Rus. Rebalancing the rebalancers: optimally routing vehicles and drivers in mobility-on-demand systems. In *2013 American Control Conference*, pages 2362–2367, 2013.
- [52] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, 2014.
- [53] Paolo Toth and Daniele Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002.
- [54] Rejane Arinos Vasco and Reinaldo Morabito. The dynamic vehicle allocation problem with application in trucking companies in brazil. *Computers and Operations Research*, 76:118–133, 2016.
- [55] Erik T Verhoef. Time, speeds, flows and densities in static models of road traffic congestion and congestion pricing. *Regional Science and Urban Economics*, 29(3):341–369, 1999.
- [56] C Voss, F Petzold, and S Rudolph. Graph transformation in engineering design: an overview of the last decade. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 37(e5):1–17, 2023.
- [57] Alex Wallar, Menno Van Der Zee, Javier Alonso-Mora, and Daniela Rus. Vehicle rebalancing for mobility-on-demand systems with ride-sharing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4539–4546, 2018.
- [58] Salomón Wollenstein-Betech, Arian Houshmand, Mauro Salazar, Marco Pavone, Christos G. Cassandras, and Ioannis Ch. Paschalidis. Congestion-aware routing and rebalancing of autonomous mobility-on-demand systems in mixed traffic. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2020.
- [59] Salomón Wollenstein-Betech, Ioannis Ch. Paschalidis, and Christos G. Cassandras. Joint pricing and rebalancing of autonomous mobility-on-demand systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 2573–2578, 2020.
- [60] James J. Q. Yu and Albert Y. S. Lam. Autonomous vehicle logistic system: Joint routing and charging strategy. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2175–2187, 2018.
- [61] James J. Q. Yu, Wen Yu, and Jiatao Gu. Online vehicle routing with neural combinatorial optimization and deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3806–3817, 2019.
- [62] Gioele Zardini, Nicolas Lanzetti, Marco Pavone, and Emilio Frazzoli. Analysis and control of autonomous mobility-on-demand systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):633–658, 2022.

- [63] Jannik Zgraggen, Matthew Tsao, Mauro Salazar, Maximilian Schiffer, and Marco Pavone. A model predictive control scheme for intermodal autonomous mobility-on-demand. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1953–1960, 2019.
- [64] Rick Zhang. *MODELS AND LARGE-SCALE COORDINATION ALGORITHMS FOR AUTONOMOUS MOBILITY-ON-DEMAND*. Ph.d. thesis, Stanford University, Department of Aeronautics and Astronautics, 2016. Submitted to the Department of Aeronautics and Astronautics.
- [65] Allan Zhao, Jie Xu, Mina Konaković Luković, Josephine Hughes, Andrew Speilberg, Daniela Rus, and Wojciech Matusik. Robogrammar: Graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.
- [66] Pengbo Zhu, Isik Sirmatel, Giancarlo Trecate, and Nikolas Geroliminis. Idle-vehicle rebalancing coverage control for ride-sourcing systems. pages 1970–1975, 07 2022.

Appendix A

Additional Information on Rebalancing-Free Simulation

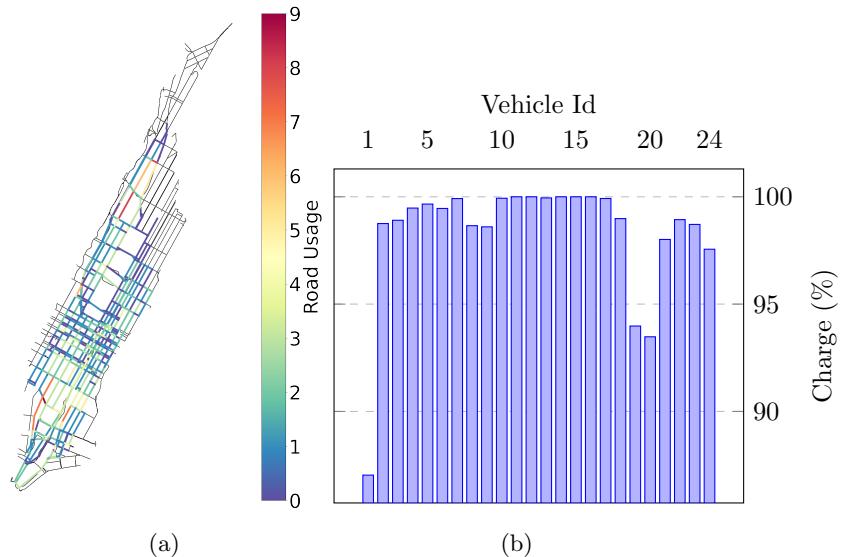


Figure A.1: Overview of System's Performance without Congestion Limits and Less Charging Time. (a) illustrates the utilization of the entire road network during the whole shift, while (b) shows averaging charging percentage of the vehicles.

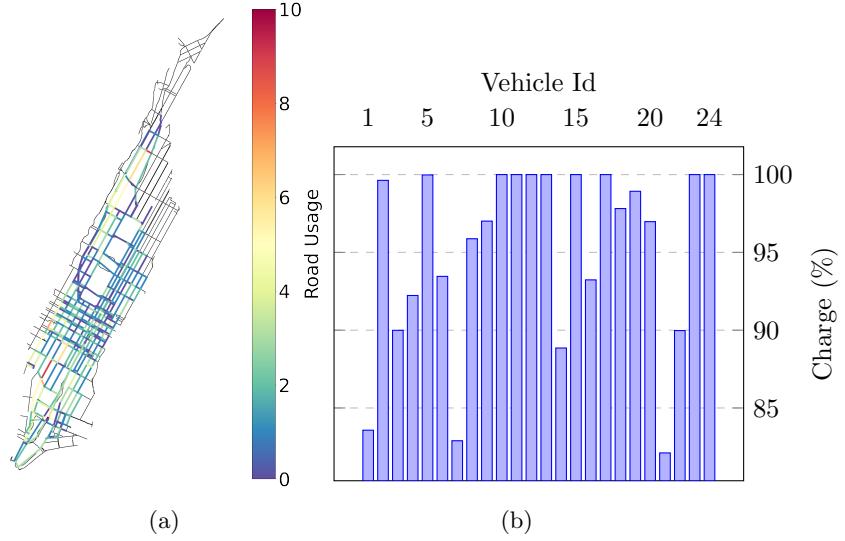


Figure A.2: Overview of System’s Performance without Congestion Limits and Less Charging Time. (a) illustrates the utilization of the entire road network during the whole shift, while (b) shows averaging charging percentage of the vehicles.

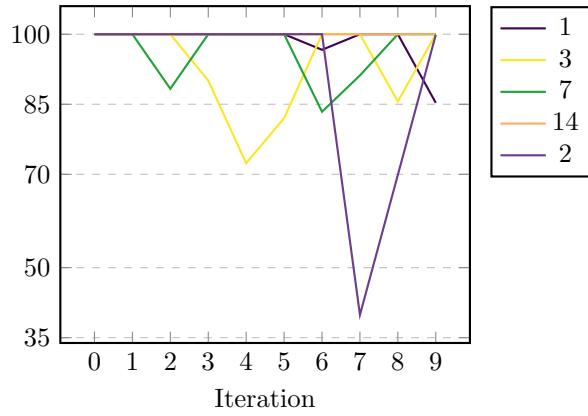
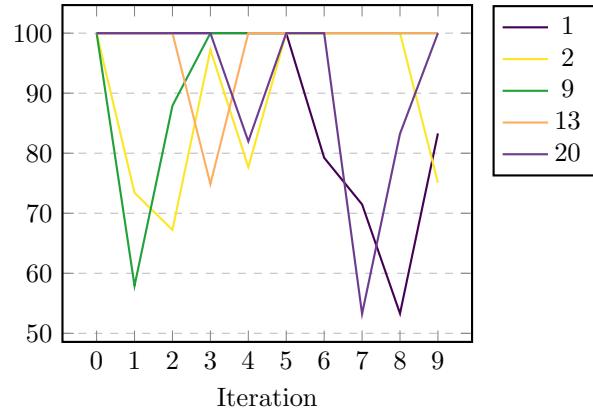
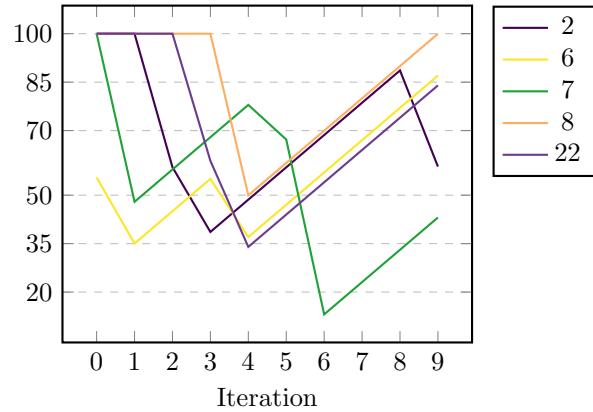


Figure A.3: State of Charge of an exemplary Set of Vehicles for Simulation 5



(a)



(b)

Figure A.4: (a) is the SOC of simulation three, while (b) represents run number four.

Appendix B

Additional Information on the Simulation with Rebalancing

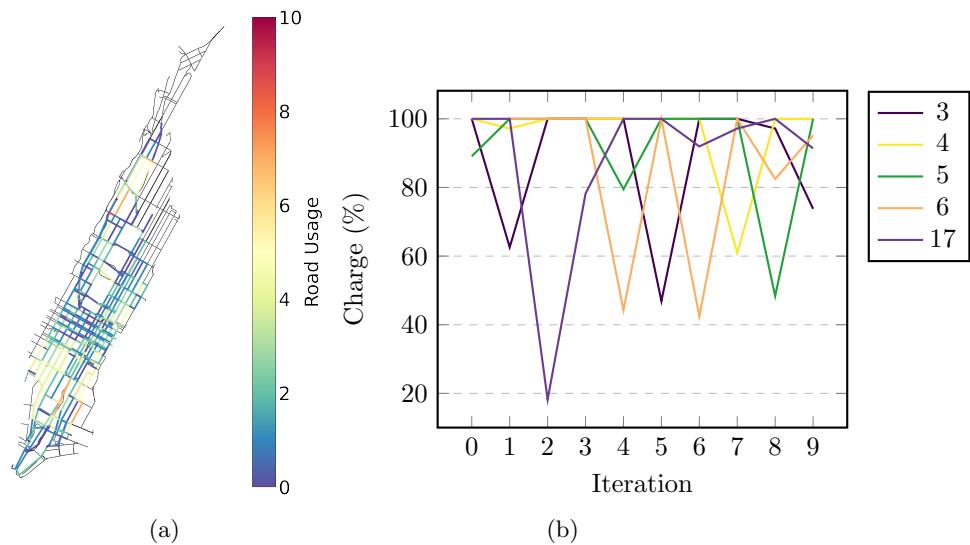


Figure B.1: Road Usage and State of Charge of an exemplary Set of Vehicles for the Simulation 1 with Rebalancing

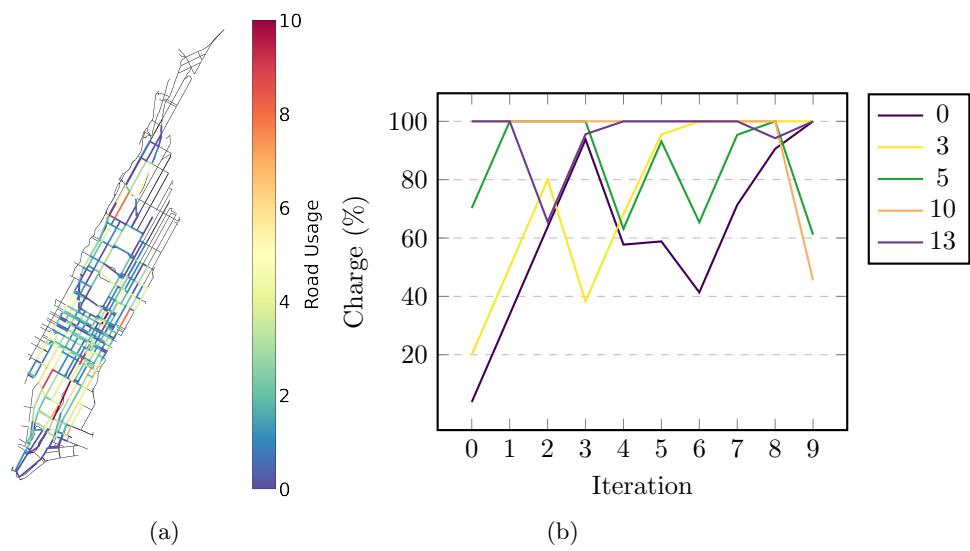


Figure B.2: Road Usage and State of Charge of an exemplary Set of Vehicles for the Simulation 4 with Rebalancing

Appendix C

Additional Information on the MPC for ATS

C.1 Propagation and Definition of the variables in \mathcal{X}

C.1.1 Propagation of g_{ji}^a

The propagation of g_{ji}^a is implemented with the following two constraints.

$$\begin{aligned} g_{ji}^a(t+1) &\geq g_{ji}^a(t) - 1 \\ g_{ji}^a(t+1) &\leq 1 - g_{ji}^a(t) \end{aligned}$$

This is basically the implementation of the following truth table.

$g_{ji}^a(t)$	$g_{ji}^a(t+1)$
0	0
0	1
1	0

Table C.1: Truth table for the propagation of g_{ji}^a

C.1.2 Definition of p_{ji}^a

The propagation of g_{ji}^a follows the truth table in Table ???. For simplicity, let $a = v_{ij}^a(t-1) + w_{ij}^a(t-1)$ and $b = v_{ij}^a(t-2) + w_{ij}^a(t-2)$. Note that this equality works because $v_{ij}^a(t-1) + w_{ij}^a(t-1)$ is either 0 or 1. Following the conjunctive normal form, this is simplified as $p_{ji}^a(t) = a\bar{b}$.

a	b	$p_{ji}^a(t)$
0	0	0
0	1	0
1	0	1
1	1	0

Table C.2: Truth table for the propagation of p_{ji}^a

$w_{ij}^a(t)$	$v_{ij}^a(t)$	$g_{ji}^a(t)$	$w_{ij}^a(t+1)$	$v_{ij}^a(t+1)$
0	0	0	0	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0

Table C.3: Truth table for the constraints on w_{ji}^a and v_{ji}^a

C.1.3 Constraints on w and v

The constraints on w and v follow the idea that can be set to 1 at any time if they are zero, since vehicles can start moving at any time instance, and they have to remain in motion until they arrived, i.e. $g_{ji}^a = 1$. This is described by the following truth table. Table C.3 can be simplified using the conjunctive normal form. For instance, in case of $v_{ij}^a(t+1)$, its constraint is described in (C.1)

$$\begin{aligned} v_{ij}^a(t+1) &= \neg w_{ij}^a(t+1) \wedge \neg g_{ji}^a(t) \wedge \neg w_{ij}^a(t)[v_{ij}^a(t) \vee \neg v_{ij}^a(t)] \\ &= \neg w_{ij}^a(t+1) \wedge \neg g_{ji}^a(t) \wedge \neg w_{ij}^a(t) \end{aligned} \quad (\text{C.1})$$

C.2 Further Proof of Stability

C.2.1 Proof that $J(x)_t$ is a Lyapunov Function

Intuitively, one can consider the time that the vehicles will spend on the road. At first glance, according to the definition found in (5.24), this would be difficult to prove to be Lyapunov. As a matter of fact, the speed of the vehicles depends on the amount of vehicles on the road, which would mean the function is not guaranteed to strictly decrease. In this situation, however, since the system is undisturbed near its equilibrium, new vehicles will not be put in motion. In other words, at time t , if there are $\sum_{i,j \in \mathcal{V}} V_{ij}(t)$ vehicles in the whole system, this number will only decrease as the system progresses, since eventually vehicles will become stationed. As a consequence, the speed of the vehicles will also

improve and, therefore, the time will decrease.

Therefore, the final cost function $J(x)_t : \mathcal{X}_f \rightarrow R_+$ can be constructed in the following way.

$$J_t(x(N)) := \sum_{i,j \in \mathcal{V}} \sum_{a \in \mathcal{A}} T_{ij}^a \quad (\text{C.2})$$

Proposition C.1: *Within the definition of \mathcal{X}_f , (C.2) is a Lyapunov Function in \mathcal{X}_f*

Proof. Considering an equilibrium point $x_{\mathcal{E}} \in \mathcal{E}$, three conditions must be met.

1. The function must be strictly positive except at $x_{\mathcal{E}}$, i.e.

$$\sum_{i,j \in \mathcal{V}} \sum_{a \in \mathcal{A}} T_{ij}^a > 0$$

This is indeed true by definition of T_{ij}^a . More specifically, when the system is not at $x_{\mathcal{E}}$, then there are vehicles moving ($\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \sum_{a \in \mathcal{A}} w_{ij}^a > 0$). Then, at least one vehicle is moving. As a result, $x_{ij}^a > 0$ and consequently $T_{ij}^a > 0$.

2. Secondly, the function must assume the value of zero at equilibrium. In other words, given the point $x_{\mathcal{E}}$

$$J_t(x_{\mathcal{E}}) = 0$$

At equilibrium, there are no vehicles moving. Clearly, the terminal cost function is zero.

3. J_t must decrease $\forall x \in \mathcal{X}_f$.

$$J_t(x(k+1)) - J_t(x(k)) \leq 0$$

Let's consider a system with only a vehicle a . If the vehicle is not rebalancing or transporting a customer, it can be considered at equilibrium, hence $T_{ij}^a = 0$. J_v does not increase nor decrease with time. More specifically, their sum is equal to 0, hence the condition is satisfied.

If the vehicle is moving, since it can not move backwards, as time increases, by propagation of x_{ij}^a , its position always increases until $x_{ij}^a \geq d_{ij}$. As x_{ij}^a increases, T_{ij}^a decreases, since the speed of the vehicles can not increase, as discussed above. If $x_{ij}^a \geq d_{ij}$, then $T_{ij}^a = 0$, falling in the scenario above. For a system composed of more vehicles, regardless of whether on the same link or not, the same discussion applies. More specifically, since the number of vehicles can not increase, since there are no more outstanding requests, the amount vehicles tend to decrease, i.e. the overall system speed increases, which as a result brings the time to decrease.

□

C.2.2 Feasibility of \mathcal{X}^t and \mathcal{X}_f^t

C.2.2.1 Feasibility of \mathcal{X}^t

$$\mathcal{X}^t := \left\{ x \mid \begin{array}{l} o_{ij}^p \in (\mathbb{N}_0)^{|\mathcal{N}|}, o_{ij}^g \in (\mathbb{N}_0)^{|\mathcal{N}|}, \\ g_{ij}^a, p_{ij}^a, f_i^a \in \{0, 1\}^{|\mathcal{A}||\mathcal{N}|} \\ x_{ij}^a \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{N}|}, V_{ij} \in \{a \in \mathbb{N}_0 : a \leq |\mathcal{A}|\}^{|\mathcal{N}|} T_{ij}^a \in [0, \frac{d_{ij}}{\epsilon}]^{|\mathcal{V}|} \end{array} \right\}$$

$$\text{where } x = [o_{ij}^p, o_{ij}^g, x_{ij}^a, f_i^a, V_{ij}, g_{ij}^a, p_{ij}^a, T_{ij}^a]^T \quad (\text{C.3})$$

Proposition C.2: (*Feasibility of \mathcal{X}_t*) Let $x \in \mathcal{X}^t$ and $u \in \mathcal{U}(t)$, then $x_+ \in \mathcal{X}^t$

Proof. Let $x^+ = [o_{ij}^{p+}, o_{ij}^{g+}, x_{ij}^{a+}, f_i^{a+}, V_{ij}^+, g_{ij}^{a+}, p_{ij}^{a+}, T_{ij}^{a+}]^T$ and $u = [v_{ij}^a, w_{ij}^a]^T$. Since $u \in \mathcal{U}$, then (5.14) is satisfied, hence $o_{ij}^{p+}, o_{ij}^{g+} \in (\mathbb{N}_0)^{|\mathcal{V}|}$. Trivially, by construction V_{ij}^+ is also $\in \mathcal{X}$. Likewise, $g_{ij}^{a+}, p_{ij}^{a+} \in \{0, 1\}^{|\mathcal{A}||\mathcal{V}|}$ because of (5.7) and (5.6).

The condition $f_i^{a+} \in \{0, 1\}^{|\mathcal{A}||\mathcal{V}|}$ can be proven with the help of (5.5). At first vehicles are either present at a station i or not. If vehicles are not stationed, then no action can be taken due to (5.10) (if $u \in \mathcal{U}$, then it is satisfied), which implies no vehicle is departing or arriving at the station. Otherwise, if vehicles are indeed stationed, then $f_i^a = 1$. If a vehicle moves, i.e. if $w_{ij}^a(t) = 1$ or $v_{ij}^a(t) = 1$, because of the first condition of (5.3), then because of (5.6) and (5.7) being mutually exclusive, at the next step $f_i^{a+} = 0$. If, on the other hand, the vehicle is approaching the station, i.e. $f_i^a = 0$ and either $v_{ji}^a(t) = 1$ or $w_{ji}^a(t) = 1$, then the following enfolds. If $x_{ij}^a \leq d_{ij}$, $f_i^{a+} = 0$. Assuming $x_{ij}^a \geq d_{ij}$, then $g_{ij}^{a+} = 1$, due to (5.5), then either $f_i^{a+} = 1$ if $v_{ji}^a(t) = w_{ji}^a(t) = 0$, or $f_i^{a+} = 0$, if $v_{ji}^a(t) = 1$ or $w_{ji}^a(t) = 1$, which would imply $p_{ij}^{a+} = 1$.

$x_{ij}^{a+} \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{V}|}$ is proved by observing x_{ij}^a in (5.3). In case $v_{ij}^a(t) + w_{ij}^a(t) = 0$, the condition is satisfied, since $x_{ij}^{a+} = 0$. On the other hand, in case of $v_{ij}^a(t) + w_{ij}^a(t) = 1$, the condition is satisfied by definition of s_{ij} and $V_{ij}(t)$.

Finally, $T_{ij}^{a+} \in [0, \frac{d_{ij}}{\epsilon}]$ can be proven as a result of the discussion made previously.

For the first case, i.e. $T_{ij}^a \geq 0$, it is necessary to prove the following.

$$\begin{aligned}\frac{d_{ij} - x_{ij}^{a+}}{s_{ij}(V_{ij}^+)} &\geq 0 \\ d_{ij} - x_{ij}^{a+} &\geq 0 \\ d_{ij} &\geq x_{ij}^{a+}\end{aligned}$$

By construction, $s_{ij}(V_{ij}(t)) \neq 0$.

Since the variable x_{ij}^{a+} tracks the position of the vehicle on the road, this can not be bigger than the road itself. Furthermore, due to (5.10), this is also safeguarded, as the vehicles becomes stationed if it reaches the end of the road. Moreover, $T_{ij}^{a+} = 0$ if $x_{ij}^{a+} \notin (0, d_{ij})$ by definition.

The second case, i.e. $T_{ij}^{a+} \leq \frac{d_{ij}}{\epsilon}$, is indeed always true since $\epsilon \leq s_{ij} \leq l_{ij}$ and only $x_{ij}^{a+} < d_{ij}$ is considered in the first case.

□

C.2.2.2 Feasibility of \mathcal{X}_f^t

$$\mathcal{X}_f^t := \left\{ x_f \quad \left| \begin{array}{l} o_{ij}^p \in \{0\}^{|\mathcal{N}||\mathcal{A}|}, o_{ij}^g \in \{0\}^{|\mathcal{N}||\mathcal{A}|} \\ g_{ij}^a, f_i^a \in \{0, 1\}^{|\mathcal{N}||\mathcal{A}|}, p_{ij}^a \in \{0\}^{|\mathcal{N}||\mathcal{A}|} \\ x_{ij}^a \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{N}|}, V_{ij} \in \{a \in \mathbb{N}_0 : a \leq |\mathcal{A}|\}^{|\mathcal{N}|} T_{ij}^a \in [0, \frac{d_{ij}}{\epsilon}]^{|\mathcal{V}|} \end{array} \right. \right\}$$

$$\text{where } x = [o_{ij}^p, o_{ij}^g, x_{ij}^a, f_i^a, V_{ij}, g_{ij}^a, p_{ij}^a, T_{ij}^a]^T \tag{C.4}$$

Proposition C.3: (*Feasibility of \mathcal{X}_f*) Let $x \in \mathcal{X}_f^t$ and $\kappa_f(x) \in \mathcal{U}_{\kappa_f}(t)$, then $x_+ \in \mathcal{X}_f^t$

Proof. Let $x^+ = [o_{ij}^{p+}, o_{ij}^{g+}, x_{ij}^{a+}, f_i^{a+}, V_{ij}^+, g_{ij}^{a+}, p_{ij}^{a+}]^T$ and $\kappa_f(x) = [v_{ij}^a, w_{ij}^a]^T$. Since the system is treated as undisturbed, $o_{ij}^{p+} = o_{ij}^{g+} = 0$. Because $\kappa_f(x) \in \mathcal{U}_{\kappa_f}(t)$, then (5.31) is satisfied, implying there is no vehicle leaving the station to serve a request. By construction V_{ij}^+ is also $\in \mathcal{X}_f$. Likewise, $g_{ij}^{a+} \in \{0, 1\}$ because of (5.7).

If $w_{ij}^a = 0$, then nothing happens within the system, therefore all the vehicles remain at the station, then $f_i^{a+} \in \{0\}^{|\mathcal{V}||\mathcal{A}|} \forall i \in \mathcal{V}$. Consequently $x_{ij}^{a+} = 0$, i.e. $\in \mathbb{R}_{\geq 0}$. If vehicles are in movement, i.e. $w_{ij}^a = 1$, a similar argument to the

one proposed for the feasibility of \mathcal{X} can be made. Because of (5.5), $f_i^{a+} = 0$ and $x_{ij}^{a+} \in (\mathbb{R}_{\geq 0})^{|\mathcal{A}||\mathcal{V}|}$, by definition of s_{ij} and $V_{ij}(t)$. Eventually, as the vehicle approaches the station, $d_{ji} = x_{ji}^a(t)$, then $f_i^{a+} = 1$ due to (5.5) and $p_{ij}^{a+} = 0$. Since (5.10) is respected ($\kappa_f(x) \in \mathcal{U}_{\kappa_f}(t)$), then (5.11) is also respected, as $w_{ij}^{a+} = 0 \implies x_{ij}^{a+} = 0$. Condition (5.32) is always respected, assuming there is no new requests.

Finally, in case $w_{ij}^a = 0$, then $T_{ij}^{a+} = 0$. If $w_{ij}^a = 1$ and the vehicle reached the station, then $g_{ij}^{a+} = 1$, which implies $w_{ij}^a = T_{ij}^{a+} = 0$. Otherwise, since $x_{ij}^{a+} \in (\mathbb{R}_{\geq 0})$, a similar argument to Proposition C.2 can be used to prove $T_{ij}^{a+} \in (\mathbb{R}_{\geq 0})$. \square

Appendix D

Additional Information on the MPC Use Case

D.1 Statistics on Taxis in NYC

D.1.1 Passengers per Hour

To calculate the number of passengers per hour, one needs to estimate the number of hours during which taxis operate in New York City. The data is taken from [37]. Typically, taxis operate around the clock, but for simplicity, let's assume they operate for 20 hours a day on average (accounting for downtime, maintenance, etc.). Let's first calculate the number of passengers per hour:

$$P_h = \frac{1000000 \text{ passengers}}{24 \text{ hours}} \approx 41667$$

To calculate the average number of passengers served by each taxi per hour, one can use the following formula:

$$P/T_h = \frac{P_h}{T_h}$$

where $T_h = N \times s$, the number of taxis N and the shift duration s . Considering a 10 hours shift, i.e. $s = 10$, and $N = 11772$.

$$P/T_h = \frac{P_h}{s \times N} = \frac{41667}{10 \times 11772} = 0.354$$

D.1.2 Average Passengers Carry in NYC

Assuming each taxi operates for 10 hours total number of passengers per day to be 1000000. The average number of passengers per taxi is calculated as follows.

$$\text{Passengers per taxi} = \frac{1000000 \text{ passengers per day}}{11772 \text{ taxis} \times 10 \text{ hours}} \approx 8.49 \text{ passengers per taxi}$$

D.2 RCS Evolution

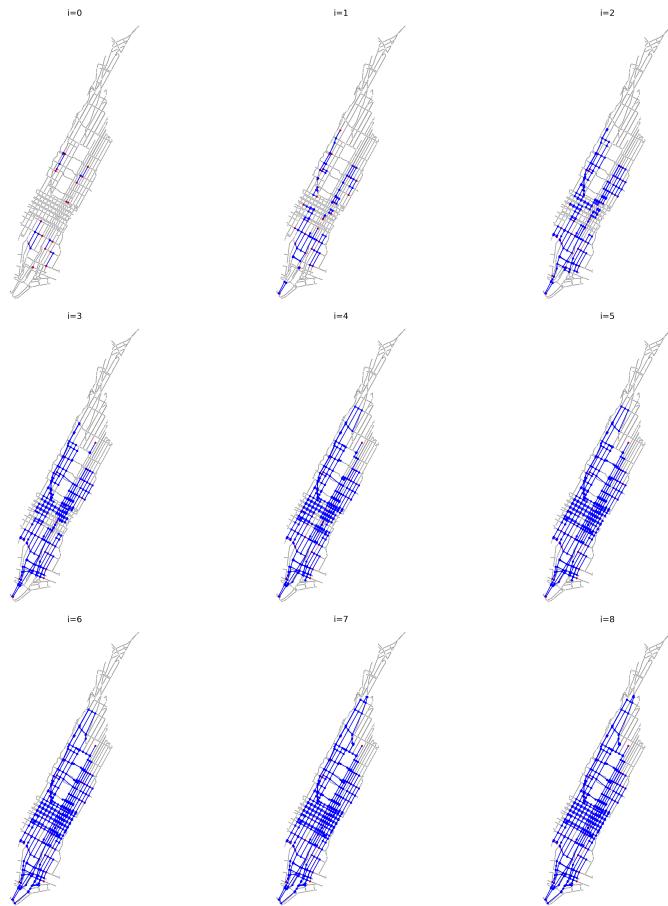


Figure D.1: Depiction of The Evolution of the RCS