Optimizing Urban Mobility in Multi-Mode Transportation Systems

by

Mohammed Bin Hariz

Thesis submitted to the

Faculty of Engineering

In partial fulfillment of the requirements

For the Ph.D. degree in

Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

The transportation system needs innovative schemes and applications to facilitate mobility in cities. These schemes should be user-friendly, easy, enjoyable, and convenient according to citizens' constraints. Three dynamic multi-mode transportation models are developed in which the passenger, for his/her trip, can use one transportation mode or a combination of transportation forms such as a car, a bus, or a bicycle. The first model is based on the particle swarm optimization (PSO) algorithm and the genetic algorithm (GA), and it is called dynamic mobility traffic (DMT). The second and third models are both based on the Stackelberg game theory with the PSO and GA, but one is a centralized infrastructure called the game theory multi-mode transport (GT-MMT) model, and the second is a community-based infrastructure called the decentralized game-theoretic (DGT) model. All three models are implemented through a realistic scenario in a specific city using OMNET++, OpenStreetMap, Inet, and Veins software tools.

We proposed a new Dynamic Mobility Traffic (DMT) scheme that combines public buses and car ride-sharing. The main objective is to improve transportation by maximizing the riders' satisfaction based on real-time data exchange between the regional manager of the transportation modes, the public buses, the car ride-sharing mode, and the riders.

The GT-MMT and DGT models, which involve multiple modes of transportation, allow the user to be an active prosumer who can travel in the city using public and private forms and make decisions about trip costs. In the two models, the passenger is the leader, and the rest of the modes are followers. The utility functions for each player are different depending on their goals. While the passenger aims to reach the destination in the shortest possible time and for the lowest price, the bus and the car try to have the seats vacant as often as possible, and the bicycle tries to increase the availability for the passenger. We propose these two models to manage passenger needs, public bus interests, car ride-sharing, and bicycle constraints. The analytical and simulation results prove the effectiveness of the proposed schemes. The effectiveness of the decentralized game-theoretic transportation model appears more clearly when compared with the centralized multi-mode dynamic approach in [1], as it gives much better optimization results.

Acknowledgments

First, with gratitude, I thank Allah (SWT), the Holy, the Creator, the Most Gracious, the Most Merciful, the Guide, the Munificent and the wise, who helps me and gives me the strength and patience to finish a significant chapter of my life.

Secondly, I would like to express my deep gratitude to my supervisor, Professor Hussein T. Mouftah, for his advice, encouragement, patience, and constant support. Without him, this thesis would not have been possible to be completed.

Moreover, I would like to thank my parents, Abdullah and Khadeeja Bin Hariz, who always believed in me. They taught and encouraged me always to be myself.

I want to extend my heartfelt thanks to my beloved wife, Nada Bin Hareez, for her patience, understanding, and support. She put up with me through the long years of research and writing. I offer my thanks to the best support person who pushed me hard to finish my work, and I thank her for helping me and always providing a suitable environment for me to finish my work. I thank my daughter Aljoury and my sons Abdullah and Ayedh. You have been a source of joy and inspiration; your presence has reminded me daily why I am committed to this work.

All thanks and gratitude go to my sister, Marwah, and to my brother and best friends in my alienation, Abdulrahman AL-Qudaimi and Abdullah Baras. I hope they will all have the chance to achieve their best in their current and future endeavours.

Finally, I extend my gratitude to everyone who has supported me, directly or indirectly, during this process, especially my uncle Ayedh and my maternal aunts Fatima and Noor. Thank you all for being part of this journey with me.

Dedication

To my father, who has always been behind me in every step of my life. To my mother, for her absolute and endless love.

To my wife Nada, without her encouragement and patience, this work would not have materialized.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronyms	Definition
AV	Autonomous mode
BB	Balanced Budget
BT	Best Time sharing
CB	Customized Bus
CM	Community Manager
CPPPMI	CarPooling Problem with the help of Prematch Information
DApp	Decentralized Application
DGT	Decentralized Game Theory
DMT	Dynamic Mobility Traffic
DQNs	Deep Q Networks
DRTS	Demand Responsive Transit System
FPC	Fuzzy Predictive Control
GA	Genetic Algorithm
GHG	GreenHouse Gases
GIS	Geographic Information System
GPS	Global Positioning System
GRUs	Gated Recurrent Units
GT	Game Theory
GT-MMT	Game Theory Multi-Mode Transportation
IC	Incentive Compatibility
IoT	Internet of Things
IR	Individual Rationality
IRS	Intercity Ride-Sharing
LDMU	Local Decision-Making Unit
ITO	Intelligent Train Operations
ITS	Intelligent Transport System

MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer nonLinear Programming Problem
MIQP	mixed-integer quadratic programming
MPEC	Mathematical Programming with Equilibrium Constraints
NN	Nearest Neighbour problem
ODBRP	On-Demand Bus Routing Problem
OM-CRA	Online Matching with Controllable Rewards and Arrival probabilities
OSM	OpenStreetMap
PDHG	Primal-Dual Hybrid Gradient
PSO	Particle Swarm Optimization
RL	Reinforcement Learning
RS	Strategies Represented
RVRPTW	Robust mode Routing Problem with Time Window
SAMoD	Shared Autonomous Mobility-on-Demand
SBRP	School Bus Routing Problem
SPA	Solution Pooling Technique
SUMO	Simulation Of Urban Mobility
TCS	Tradable Credit Scheme
TS	Transportation System
VMT	mode Miles Travelled
VRP-IRS	IRS mode Routing Problem
VRPTWPD	mode Routing Problem with Time Windows and simultaneous Pickup and Delivery
VRP	mode Routing Problems
VRPSCD	mode Routing Problem with Stochastic Demands and Customers
VSP	mode Sharing Programming
XAR	Xhare-A-Ride
2SPSPP	2 Synchronization Points Shortest Path Problem

List of symbols

Notation	Definition
	Common Notations
A	$E \cup \{S_1^m, D_{nm}^m\}$, where S_1^m is the source and D_{nm}^m is destination points for modes
alg	Counter of iterations of the algorithm for the three models
Ca_m	Capacity of mode m
$\mathcal{C}_{p_i}$	Ordered list of tuples, where each tuple = $(m_{nm}, S_1^{p_i}, D_{np}^{p_i}, TT_{p_i}, TF_{p_i})$
D	Set of delivery points
D_m	Set of destination points of the modes
D_{nm}^m	Destination of mode m
E	Union of all pickup and delivery points
ED	Set of edges for the routes
ED_m	Set of the edge for mode m in the trip
ED_{p_i}	Set of edge for the passenger p_i in his trip
ed_x	Edge (route) id
$G(N, ED)$	Representing the route and road as a graph with nodes and edges
I	Number of the iterations of the algorithm for the three models
$If_k(t)$	State vector of infrastructure k at time t
M	Set of all transportation modes
m	Travel mode: m= c for ride-sharing mode, m= b for public bus mode
	and m= bk for bicycle-sharing mode
MD_m	Maximum travel distance for mode
MT_m	Maximum travel period for mode
N	Set of nodes representing all stations/stops for all modes
Num_m	Number of modes
$Nump_{s_{n+i}}^m = -l_{s_{n+i}}^m$	indicate the change in the number of passengers at location s_{n+i}
$Nump_{s_n}^m = l_{s_n}^m$	indicate the change in the number of passengers at location s_n
O	Set of pickup points
O_m	Set of original points of the modes
$RO(S_1^m, D_{nm}^m)$	Optimal route from mode source S_1^m to mode destination D_{nm}^m

RT	Real-time traffic data
S_1^m	Source of mode m
TD_m	Total trip distance of m in one trip
$TD_{p_i}^m$	Total trip distance of passenger when using m in his trip
$TF_{p_i}^m$	Total trip fare for the passenger when using m in his trip
TT_m	Total trip time of m in one trip
$TT_{p_i}^m$	Total trip time of the passenger when using m in his trip
$V_i(t)$	State vector of mode i at time t
Vs_{max}	Urban speed limit
Vs_{m_i}	Speed of the mode m_i
$x_{s_n}^m$	Binary decision variable = 1 if mode m provides service at location s_n. Otherwise = 0
x_m	Binary indicators (0 or 1) denoting whether the mode is used during the trip or not
θ_{m_i}	Direction of the mode m_i
α, β, γ	Weight factors for the objective function
ϵ	convergence criterion (predefined threshold)

Passengers Notations

AT_{p_i} and BT_{p_i}	Constants to compute the maximum passenger travel time
$D_{np}^{p_i}$	Last destination for the passenger p_i
Ea_{p_i}	Earliest arrival time for the passenger p_i at the last destination $D_{np}^{p_i}$
Ed_{p_i}	Earliest departure time for the passenger p_i from the source point $S_1^{p_i}$
$I_r^{p_i}$	Intermediate stops for the passenger p_i
La_{p_i}	Latest arrival time for the passenger p_i at the last destination $D_{np}^{p_i}$
Ld_{p_i}	Latest departure time for the passenger p_i from the source point $S_1^{p_i}$
$Match$	Set of all matching options available for the riders
$MD_{p_i}^m$	Maximum allowed distance for the passenger
MF_{p_i}	Maximum trip fare for the passenger
MT_{p_i}	rider's maximum trip time
MT_{p_i}	Maximum trip period for the passenger
MW_{p_i}	Maximum passenger waiting time

p_i	Rider id
R	Set of rider request
SR	Set of all passenger points
$S_1^{p_i}$	Source point for the passenger p_i
TD_{p_i}	Total distance of passenger in the whole trip
TF_{p_i}	Total trip fare for the passenger in the whole trip
TT_{p_i}	Total trip time of the passenger in the whole trip
$t(I_r^{p_i}, I_{r+d}^{p_i})$	Time period from point to point
$t(S_1^{p_i}, D_{np}^{p_i})$	Straight-line travel time from location $S_1^{p_i}$ to $D_{np}^{p_i}$
WT_{p_i}	Actual passenger waiting time
$T^m_{D_{np}^{p_i}}$	Time when the passenger arrived to his destination
$T^m_{I_r^{p_i}}$	Time when the passenger arrived to the intermediate point
$st^m_{I_r^{p_i}}$	Time required to serve the passenger at point $I_r^{p_i}$ by mode m
$T^m_{I_r^{p_i}}$	Time of starting service at point $I_r^{p_i}$ by mode m
$Twalking_{I_r^{p_i}}$	Walking time for the passenger
$x(p_i, Ma)$	Binary variable that is 1 if rider p_i is matched with option Ma and 0 otherwise

Ride-sharing Notations

AD_{c_j},BD_{c_j}	Constant to calculate the ride-sharing maximum travel distance
AT_{c_j}, BT_{c_j}	Constant to calculate the mode's maximum travel time
C	Set of the ride-sharing mode
Ca_{c_j}	Capacity of the ride-sharing c_j
c_j	Ride-sharing mode id
$D_{nc}^{c_j}$	Last destination for the ride-sharing mode c_j
Ed_{c_j}	Earliest departure time for c_j
$Fadd_{c_j}$	Additional charges that vary depending on specific conditions
$Fbase_{c_j}$	Fixed base fare cost applied to all rides
Fd_{c_j}	Fare rate per unit distance (kilometre)
Ft_{c_j}	Fare rate per unit time (minutes)
$I_a^{c_j}$	Intermediate stops for the mode c_j
La_{c_j}	Latest arrival time for the c_j at the last destination $D_{nc}^{c_j}$
MD_{c_j}	Ride-sharing maximum trip distance

MT_{c_j}	Ride-sharing maximum trip time
MW_{c_j}	Maximum ride-sharing waiting time
$Nump(S_1^{c_j}, D_{nc}^{c_j})$	Total number of passengers from the pickup point $S_1^{c_j}$ to the point $D_{nc}^{c_j}$
$Nump_{e_x}^{c_j}$	Number of the passengers inside the ride-sharing on each route
$S_1^{c_j}$	Source point for the ride-sharing mode c_j
TD_{c_j}	Total trip distance of c_j in one trip
$TD_{p_i}^{c_j}$	Total trip distance of the passenger when he rides c_j
TF_{c_j}	Total trip fare of c_j in one trip
$TF_{p_i}^{c_j}$	Total trip fare of the passenger when he rides c_j
$Tr_{p_i}^{c_j}$	Time when the passenger request the ride
$TT_{p_i}^{c_j}$	Total trip time of the passenger when he rides c_j
TT_{c_j}	Total trip time of c_j in one trip
σ_{c_j}	Acceptable time detour in the trip

Public Bus Notations

$at_{s_u}^{b_k}$	Arrival time for the bus
B	Set of buses
b_k	Bus id
Ca_{b_k}	Capacity of the bus b_k
$D_{s_u}^{b_k}$	Demand at stop s_u
$D_{\text{avg}}^{b_k}$	Average demand fro the bus
$D_{nb}^{b_k}$	Last destination for the bus b_k
$dt_{s_u}^{b_k}$	Departure time for the bus b_k
Ed_{b_k}	Earliest departure time for b_k
f_{s_u}	Buses frequency (variations in demand)
$I_u^{b_k}$	Intermediate stations for the bus b_k
La_{b_k}	Latest arrival time for the b_k at the last destination $D_{nb}^{b_k}$
$Nump_{e_x}^{b_k}$	Number of the passengers inside the bus on each route
R_{b_k}	Set of routes for the bus b_k
r_{b_k}	A route for the bus b_k
$S_1^{b_k}$	Source station for the bus b_k
S_{b_k}	Set of stations for b_k

$scht_{s_u}^{b_k}$	Schedule time for the bus
So_{b_k}	Optimized schedule for the bus
$T_{\text{tolerance}}$	Allowable deviation from the schedule
TD_{b_k}	Total trip distance of b_k in one trip
$TD_{p_i}^{b_k}$	Total trip distance of the passenger when he rides b_k
$TF_{p_i}^{b_k}$	Total trip fare of the passenger when he rides b_k
TT_{b_k}	Total trip time of b_k in one trip
$TT_{p_i}^{b_k}$	Total trip time of the passenger when he rides b_k
TF_{b_k}	Total trip fare of b_k per hour
ΔT_{b_k}	Bus time interval for peak hours

Bicycle-sharing Notations

av_{bk_n}	Boolean number to determine if bicycle bk_n is available or not
bk_n	Bike-sharing id
$D_{nbk}^{bk_n}$	Last destination of the journey using bk_n
Ed_{bk_n}	Earliest departure time for bk_n from the source point $S_1^{bk_n}$
$Fma_{s_{bk}}$	Cost associated with maintenance at location s_{bk}
$Fre_{s_{bk}}$	Cost associated with redistribution at location s_{bk}
La_{bk_n}	Latest arrival time for bk_n at the last destination $D_{nbk}^{bk_n}$
$Ma_{s_{bk}}$	Number of maintenance activities at location s_{bk}
$Nbike_{\text{av}}^{s_{bk}}$	Number of bicycles at docking station s_{bk}
$Nbike_{\text{use}}^{s_{bk}}$	Number of bicycles currently in use at station s_{bk}
nbk	Number of locations or docking stations
$Nbike_{\text{min}}^{s_{bk}}$	Minimum number of bicycles available at stop s_{bk}
$Nbike_{\text{min}}^{s_{bk}}$	The adjusted demand at station s_{bk} after redistribution
$Numbk_{use}(I_l)$	Denotes the number of bicycles in use at location I_l
$Numbk_{av}(I_l)$	Total number of bicycles available at location I_l
$Numbk_{av}^{s_{bk}}$	Number of available bike in the location s_{bk}
$Numbk_{re}(s_j, s_{bk})$	Number of bicycles relocated from station s_j to station s_{bk}
$Numbk_{re}(s_{bk}, s_k)$	Number of bicycles relocated from station s_{bk} to station s_k
$Numbk_{re}$	Maximum allowable re-balancing effort
$Opt_{max^{s_{bk}}}$	Maximum allowed operations per station

$Re_{s_{bk}}$	Number of redistribution activities at location s_{bk}
$S_1^{bk_n}$	Source point of the being journey for the bk_n
$Tcost_{max}^{s_{bk}}$	Maximum budget for operational costs
TD_{bk_n}	Total trip distance of bk_n in one trip
$TD_{p_i}^{c_j}$	Total trip distance of the passenger when he rides bk_n
TF_{bk_n}	Total trip fare of bk_n in one trip
$TF_{p_i}^{bk_n}$	Total trip fare of the passenger when he rides bk_n
TT_{bk_n}	Total trip time of bk_n in one trip
$TT_{p_i}^{bk_n}$	Total trip time of the passenger when he rides bk_n
U_{bk_n}	Utility function for the usage rate of bicycles

Additional Notations for DGT

$\mathbf{A}_{\text{global}}^*$	Global optimized strategies
Com	Community in the DGT model
Co_{min}	Minimum required connectivity
$Co(NO_i, F_k)$	where C represents connectivity between node NO_i and fog node F_k
G	geofenced area
$LO(m_i)$	Location of the mode m_i
$\mathbf{S}_i^*$	Optimized local strategies
$TD(NO_i, NO_j)$	where TD is the distance function, NO_i and NO_jrepresent individual nodes, and W is the Wi-Fi range
T_{global}	Maximum global iterations
T_{local}	Maximum local iterations
TS	Convergence threshold in the DGT model

Genetic algorithm Notations

CH	Each candidate solution in the population (chromosome) which represents the combination of strategies
$CH_{mutated}$	New chromosome (strategies) after the mutation process
CH_{new}	offspring chromosome
CH_i	Parent chromosome that is selected from the population to contribute to the creation of the next generation
CH_{selected}	Chromosome (strategies) selected from the tournament process

$Fintess$	Evaluates the quality of each solution in GA (Fitness Function)
Ge	Elements of a chromosome representing decision variables (Genes)
Gt	Generation of the GA process
M	Crossover mask
mu	Mutation probability
NCH	Number of all chromosomes in the algorithm
PO_o	Set of candidate solutions (population)
PS_{ga}	Population size for the GA (number of chromosomes in one population)
Ts	Tournament size
z and β	Weighting coefficients that prioritize passenger satisfaction and service efficiency

PSO Notations

co	cognitive learning factor in PSO
$gbest$	The near-optimal solution found by any particle in the swarm (Global Best)
NP_{pso}	Number of particles in the swarm
Pa_i	Each particle represents a set of strategies (Particles)
$pbest_i$	The near-optimal solution found by a specific particle (Personal Best)
st	Random number uniformly distributed in the range [0, 1], introducing stochastic behaviour
$Swarm$	Collection of particles
v_i	The rate of change of the particle's position (Velocity)
w	Inertia weight, which controls the influence of the previous velocity of the particle
$Xpso_i$	The current solution of the particle (Position)

Stacklberg Notations

A	Set of the strategies of all players
A_m	The optimal strategy chosen by the follower in response to the leader's strategy
$A^*_m(A_p)$	Optimal strategy of the follower given the leader's strategy
A_{p_i}	The optimal strategy chosen by the leader
$Fitness(A)$	Total fitness for all the players in the model
Followers (m)	The decision-makers who move after observing the leaders' decisions (transportation modes)
I_m	Preference set of mode m

Leaders (p_i)	The decision-makers who move first (passengers)
U_m	Utility function of mode m
U_{p_i}	The utility function for leader player (Payoff Function)

Chapter 1 Introduction

1.1 Background

Today, governments are committed to bringing about urban and modern development, increasing their citizens' satisfaction, and improving their citizens' living standards regarding services such as transportation. Recent studies indicate that the world's cities are experiencing a huge increase in the number of vehicles. This number leads to congestion problems on the roads. As we know, road congestion [6] [7] has numerous negative impacts, for example, wasting the time of the passenger [8] [9] , environmental air contamination [6], and expanded fuel utilization [10] [11] [12]. Currently, transportation systems, which are the second highest source of pollution in cities [13], must deal with many issues to meet user expectations in terms of efficient, healthy, environmentally friendly, and convenient trips. For example, GreenHouse Gases gas (GHG) emissions [14], road congestion [15], and accidents [16] place a heavy burden on governments to provide an adequate environment for their population [17]. With growing concerns, there is a greater need for sustainable transportation solutions. This involves minimizing the impact on the environment, reducing emissions, and encouraging the use of eco modes of transportation such as transit, cycling and walking.

Today, commuters expect transportation systems that are not only efficient in terms of travel time but also user-friendly. This includes access, comfort, affordability and reliability. An efficient and user-friendly transportation system can greatly enhance the quality of life in areas. Fundamentally, mobility is ensured by a three-component system: vehicle, driver, and transport infrastructure (which is named, in a broader sense, environment). Over time, vehicles and infrastructure have evolved spectacularly through technological upgrades and improvements, with the human driver remaining the only constant of the system [18]. Smart mobility is a broad term, so far, with no fixed definition. But according to the reference [19], a definition can be provided as follows: "the use of Information and Communication Technology in modern transport technologies to improve urban traffic.". As we know, ride-sharing services contribute to helping people with disabilities in their mobility more so than public transportation. However, according to the reference [20], the developers of those types of services need to add many necessary features to make these services more convenient and more accessible for those with disabilities.

As we know, riding in private cars is more comfortable than taking public transportation such as a bus. However, a bus can provide more seats for passengers and thereby limit traffic crowding, especially during rush hours. Also, when one private car is used by only one or two people, many seats are empty. Replacing a private vehicle with a carpooling service will eliminate this inefficiency [21]. In general, buses have the lowest fare in comparison to other forms of public transport. Today's bus passengers can receive real-time information such as departure and arrival trip time from the stations [22]; however, buses are still not efficient in terms of trip optimization. As a solution, we can improve the current mobility traffic system and meet passenger requirements if bus movement and carpooling are combined into one system.

The mobility system in the city is an important pillar for urban and modern development [23]. Currently, the transportation forms (e.g., public and private)are disjoint and lack coordination, which makes them more and more stressful. Moreover, as the human population grows, the huge increase in vehicle numbers on the roads leads to many challenges such as road congestion, wasting time for passengers and air pollution, etc. These issues negatively impact citizens' daily lives, business productivity, health, and the environment. Today, government and city planners are focusing on building smart transportation solutions to optimize transportation services and improve mobility in the city.

The concept of shared mobility based on new technologies can improve the reliability and efficiency of the traditional transportation system. Indeed, dynamic ride-share systems allow passengers to take other routes, possibly reducing a trip's time and cost. At present, road network developers are focusing on integrating the two modes of transportation to provide passengers with more dynamic transportation and on-demand mobility services [24]. In recent times, transit forms have developed significantly due to tremendous progress in the dynamic multi-mode transportation system, which has the ability to collect data in near real-time. At the same time, developments in computing technologies, vehicular communication, and sensing make it possible for cars to inform their neighbours about traffic data very quickly.

Recently, there has been an increased tendency to establish new innovative schemes and applications to carry out user-friendly, secure and value-added services in TSs. For example, mobility-asa- service is a new promising application that will save the user several resources such as parking space, time and money [25]. Car-sharing has also become popular for various reasons, such as improving traffic conditions by decreasing the number of cars on the road, enabling the user to enjoy the ride, improving the social links between riders and providing a cost-effective way to increase citizens' mobility [26]. Another example is bike-sharing programs that offer affordable short-term rentals throughout a defined geographical area [27]. These bike-sharing programs are run by private and public companies such as Divvy (Chicago), Citi Bike (New York), BIXI (Montreal) and GOBIKE (Ottawa).

The decentralised system can be described as being community-based, a term that clarifies the idea of moving from absolute central authority to community participation to make the TS [28] increasingly efficient and convenient in meeting a community's needs. In the last decade, the sharing economy has emerged as one of the most critical sectors to have formed accelerated growth in various service fields [29]. In this context, many transportation companies, such as Uber, tend to share the economy by allowing drivers and passengers to contribute to determining the amount of profit and cost. The big leaps in smart cities and decentralised systems have contributed to the growth and development of the concept of a sharing economy [30] and improving the citizens' satisfaction regarding proposed public and private services.

It should be noted that ride-sharing requires a group of persons using the same car and sharing the trip cost (time and price), while the bicycle requires a group of riders to rent the bike and share it for a specific time. According to [31], many residents prefer to use bicycles for their short commutes. Cycling helps to reduce the incidence of many health problems, such as obesity while helping to maintain a healthy heart.

Current mobility transport projects usually focus solely on one form of transportation and a static mobility model of carpooling service. In the static mobility model, passengers send a sharing request and indicate the time they want; next, the driver arranges the trip for these specifications. The same is true for public buses, so departure and arrival times are also known and fixed. While static ride-sharing is satisfactory for a long trip, it is not efficient for journeys within an urban area. In this case, dynamic ride-sharing is more suitable because it provides a real-time approach to the driver and the passenger [32]. Dynamic ride-sharing establishes a model that can give travellers the possibility of forming ride-sharing instantaneously [33].

1.2 Motivation

Most of the previous studies, especially those carried out between 1999 and 2023, focused on traditional ride-sharing schemes with a static itinerary, where the passenger shares his trip with other passengers with the same origin and/or destination [34] [35] [36]. Nowadays, dynamic ride-share systems allow passengers to take another route with the possibility of reducing the trip's time and cost. At the same time, road network developers are focusing on integrating multiple forms of transportation to provide passengers with more dynamic transportation and a mobility-on-demand service. The paper [37] proposed a model that combined carpooling with public transport to obtain multi-modal mobility planning. However, this system combines only two modes of transport and is designed for the sole purpose of finding the shortest route for the user to reach his destination without considering the cost of the trip. Smart transportation is an important pillar of smart city deployment [38]. Expanding the traditional Transportation System (TS) by increasing the number of buses and cars on the road could potentially risk the fulfilment of environmental, social and economic requirements for the Citizens' well-being in cities, especially considering population growth as well as diversity in terms of geographic location, weather conditions and specific user needs.

One of the new trends in the modern transportation system planning models in smart cities is the Game Theory (GT) concept to solve congestion problems with multi-player objectives. The GT, with its different strategies, helps dynamic transport system developers to design a model that provides the shortest trip time and the lowest cost for transporting passengers [39]. Game theory can be considered an optimization method. In many cases, the optimization algorithm and game theory are used together to solve optimization issues in the daily problems in our lives. Recently, game theory has been used with optimization algorithms to solve several problems by finding near-optimal solutions. However, until now, this trend has been rarely seen in the field of intelligent transportation development [40] [41] [42]. As we know, game theory is a set of analytical tools that are used to find the near-optimal solution to choice-making problems.

The deployment of dynamic TS is complex and requires critical decisions to be made in response to the following questions:

- How can the researcher's revolutionise the transportation service in cities to make mobility as simple as possible?

- How can the author enhance the user-friendly nature and security of the TS?
- How will the user obtain maximum benefits without disturbing the service provider's interest?

These questions can be solved by developing efficient transportation schemes customised for a specific community's needs and constraints [43] [44] [45].

1.3 Objectives

Our thesis has several objectives in the field of a dynamic traffic system for multi-mode transportation in smart cities:

1. To propose a dynamic mobility transportation model that integrates multiple transportation modes (ride-sharing and buses) to reduce travel time, minimize costs, and improve overall transportation efficiency in urban.
2. To design a central decision-making dynamic mobility scheme based on game theory to solve some of the transportation trip problems.
3. To develop a transportation model that is also based on game theory but for a community-based transportation system, helping optimize the trip.
4. To incorporate real-scenario data collection using simulation tools and advanced optimization algorithms into multi-mode transportation models. The objective is to improve the adaptability of these models to dynamic urban environments, ensuring that transportation systems can respond swiftly to changes in traffic conditions and passenger demand, thereby enhancing the overall efficiency and user satisfaction of urban transportation networks.
5. To figure out and improve the roles and goals of leaders (passengers) and followers (ride-sharing, bus, and bicycle modes) in a centralized and community-based multi-mode transportation system based on the game theory concept. The goal is to keep travel time and costs as low as possible for passengers while allowing modes of transportation to change their strategies to improve system efficiency, coordination, and user satisfaction.

1.4 Thesis Contributions

The thesis makes several contributions to the field of dynamic multi-mode transportation systems in a smart city. More specifically, we focus on ride-share and public buses as modes of transportation in this thesis. Consequently, our contributions to this thesis are as follows:

1. We introduced a novel dynamic mobility transportation (DMT) model to enable passengers to use buses or cars either exclusively or together during their trip to minimize travel time and costs. It uses Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) for real-time route and schedule optimization. This model lays the basic foundations for reaching the project's main objective in [46]. The model will be compared with existing heuristic multi-load models to demonstrate its effectiveness in reducing travel time, minimizing costs, and improving overall transportation efficiency in smart cities.

2. We developed a new dynamic mobility traffic model called game theory multi-mode transportation (GT-MMT) based on the GT concept. In the GT-MMT scheme, bicycles are added to the DMT framework, and game theory, specifically the Stackelberg game theory, is used to model how passengers (leaders) and transportation modes (followers) make strategic decisions that improve the system based on these decisions.

3. We proposed a novel community-based transportation scheme called Decentralised Game-Theoretic(DGT). The DGT model decentralizes control, allowing each transportation mode to independently optimize its operations using local information and game theory principles to ensure coordination and efficiency without a central control system. The model enhances flexibility, scalability, and robustness in dynamic, real-time urban transportation environments. To prove its effectiveness, we compared the DGT with another multi-mode model in the reference [5].

4. Many simulations are run for the three models to implement the proposed model and compare it with other models to ensure the effectiveness of our models.

1.5 Thesis Outline

The remainder of the thesis is organized as follows:

Chapter 2 presents state-of-the-art knowledge of smart cities' multi-mode and single-mode transportation systems. This chapter also provides an overview of the optimization algorithms and the approaches used to enhance trips in urban areas. The most relevant issues and challenges in the transportation system are highlighted also in this chapter. Chapter 3 discusses the design and evaluation of the DMT, GT-MMT, and DGT models, focusing on data management, network topology, parameters, and simulation tools. In Chapter 4, we describe the DMT model's development, integration, and effectiveness. Chapter 5 presents the multi-mode game optimization model, GT-MMT, based on a Stackelberg and the hybrid optimization algorithm. In Chapter 6, we propose the community-based model called DGT. Finally, conclusions and future directions are presented in Chapter 7.

1.6 Papers published

1. M. B. Hariz, D. Said, and H. T. Mouftah, "Mobility traffic model based on combination of multiple transportation forms in the smart city," in 15th International Wireless Communications Mobile Computing Conference (IWCMC). IEEE, pp. 14–19 [46].

2. M. Bin Hariz, D. Said, and H. T. Mouftah, "A dynamic mobility traffic model based on two modes of transport in smart cities," Smart Cities, vol. 4, no. 1, pp. 253–270 [47].

3. M. Bin Hariz, D. Said, and H. T. Mouftah, "Game theoretic approach for public multi-mode transportation in smart cities," IET Networks, vol. 10, no. 5, pp. 201–216 [1].

4. M. B. Hariz, D. Said, and H. T. Mouftah, "Game theoretic approach for a multi-mode transportation in smart cities," in International Symposium on Networks, Computers and Communications (ISNCC). IEEE, pp. 1–6 [48].

5. M. Bin Hariz, D. Said, and H. T. Mouftah, "Decentralised game-theoretic management for a community-based transportation system," IET Smart Cities, vol. 2, no. 4, pp. 181–190 [49].

Chapter 2 State of the Art

2.1 Introduction

At present, regular taxi services are facing fierce competition from car-sharing services that have entered the market very strongly. This is due to the participation of vehicles to the wishes of passengers and drivers in providing demand-driven travel dynamically.

This section reviews previous studies of dynamic mobility-traffic systems developed by integrating ride-sharing and public transit models. The multimodal route planning problem is a combination of two or more different modes of transport in a single layout that allows the traveller to move by more than one means of transport to reach his goal. In [50], the authors investigate the potential advantages of merging ride-sharing and public transportation. In particular, ride-sharing and public transit can complement one another. While ride-sharing can function as a feeder system connecting less densely populated regions to public transportation, the public transit system can expand the reach of ride-sharing and decrease drivers' need for re-routing. As a result, integrating public and ride-sharing transportation may assist in overcoming itinerary incompatibilities between drivers and passengers by facilitating their matching. However, the authors have only focused on the most straightforward (from the driver's perspective) issue, i.e., the driver's willingness to pick up and drop off two passengers at the same transit stop. Currently, research related to the smart city concept for the Transportation System (TS) has focused on how to decrease CO2 emissions, travel costs and travel time. Clearly, the task of planning advanced public transport networks is not as easy and straightforward. Designing systems for advanced public transportation requires special techniques to develop a proper model that takes into account the static track of buses and trains dependent on a specific time [51] [52]. The ride-sharing optimization problem focuses on finding the shortest path or the shortest trip interval to arrive at one's destination [53] [54]. This optimization leads to enhancing the quality of life of the person who uses car ride-sharing or a public bus to travel within a city.

2.1.1 Importance of Dynamic Systems and Transportation Optimization

Dynamic mobility in smart cities, defined as the movement of people and vehicles or anything, is designed to be dynamic. As such, it can respond intelligently and often automatically to changes and surrounding traffic. Dynamic systems play a role in the realm of modal transportation because of their ability to adapt to changing conditions and demands, in real time. This adaptability is vital for managing the interplay of different modes of transportation and ensuring optimal service delivery. Now lets explore the intricacies of why dynamic systems so pivotal in transportation within a multi-modal context [55]. Dynamic transportation systems offer a multitude of benefits. We can conclude it and compare it with the static one in the table 2.1.

The ride-sharing optimization problem focuses on finding the shortest path or the shortest trip interval to arrive at one's destination This optimization leads to enhancing the quality of life of the person who uses car ride-sharing or a public bus to travel within a city. The main benefits of the transportation optimization models can be concluded in the table 2.2.

2.1.2 Role of Game Theory in Transportation Optimization

Game theory is a framework that was developed to study how different decision-makers (agents) interact strategically. It has applications in economics, political science, psychology, operations research, and transportation planning. In the field of transportation optimization, game theory plays a role in understanding and predicting the behaviours of all parties involved, such as commuters, transport operators, and passengers [62].

Game theory helps describe interactions among actors who may have certain objectives or varying motivations when using public transportation systems [63]. By studying these interactions, it becomes possible to anticipate policy changes or system modifications that could affect transport operations and their expected outcomes [64].

Within the framework of game theory, the notion of competitive scenarios involving transport providers competing for market share or individual drivers seeking the shortest routes during rush hours can be modelled as competition. For cooperative scenarios, simulations can also replicate

Table 2.1: Comparison of Dynamic and Static Transportation Systems

Aspect	Dynamic Transportation	Static Transportation
Definition	Real-time, adaptive routing and scheduling of transportation modes based on current conditions and demands.	Pre-determined routes and schedules that do not change in response to real-time conditions.
Flexibility [56]	High flexibility; routes and schedules can adapt to real-time traffic conditions, passenger demand, etc.	Low flexibility; routes and schedules are fixed and do not change regardless of real-time conditions.
Passenger Interaction [57]	Passengers can make real-time decisions and adjustments to their travel plans.	Passengers follow a fixed schedule with limited ability to make changes once a trip is planned.
Optimization Algorithms Used	Genetic Algorithms (GA), Particle Swarm Optimization (PSO), real-time data integration.	Heuristic or simple optimization algorithms, often without real-time data integration.
Response to Traffic Conditions [58]	Can respond dynamically to changes in traffic, avoiding congested areas and optimizing travel time.	Cannot adjust to real-time traffic conditions, leading to potential delays and inefficiencies.
Cost Efficiency	Can optimize routes and modes to minimize costs dynamically.	Fixed costs based on static routes and schedules, which may not always be cost-efficient.
Environmental Impact [59]	Potentially lower environmental impact by optimizing routes and reducing idle times and emissions.	Generally higher environmental impact due to less efficient routing and longer idle times.
User Satisfaction [60]	Higher satisfaction due to personalized and optimized travel experiences.	Lower satisfaction as users have to adhere to fixed schedules and routes.
Scalability [61]	Highly scalable as it can adjust to varying levels of demand and traffic conditions.	Limited scalability due to the rigidity of fixed routes and schedules.

cooperation such as when people share rides or transportation providers coordinate their efforts [65]. In game theory, equilibrium concepts such as Nash equilibrium come into play when no participant benefits from changing their strategy. In transportation terms, this means finding the usage of routes and transit resources based on demand patterns [66]. Game theory can guide the design of mechanisms that promote behaviour across the system. For example, congestion pricing can be viewed as a game where drivers choose routes based on pricing signals to reduce congestion [67] [68].

Analyzing modes of transport such as cars, buses, and trains is an approach that can provide valuable insight into their respective strategies as tactical moves in a game [69] [36]. Game theory aids in predicting how individuals will adjust their travel behaviours in response to policy or infrastructure changes [67] [70]. Game-theory-inspired algorithms can be developed for real-time transportation optimization purposes. This includes creating pricing models for toll roads or ride-sharing services [71] [72].

2.2 Centralized vs. Decentralized Systems

2.2.1 Centralized Systems

Centralized transportation systems are systems where one entity or group makes decisions regarding all elements within a network. For instance, a government transportation agency or a major transit operator might make these calls. These systems can often be found in urban environments where there is a need to manage complex networks of buses, trains, and other forms of public transport. Their main characteristics are central control over scheduling, routing, fare-setting, and policy enforcement. Centralized systems offer several distinct benefits, such as economies of scale, efficient decision-making procedures, consistent service provisioning, and comprehensive network-wide planning capabilities. Centralized transportation systems may also be less adaptable and slower in responding to local needs or sudden shifts, as well as bureaucratically inefficient [73].

Real-world examples demonstrate the efficacy of centralized systems like Transport for London (TfL), which oversees public transit modes in London through one centralized framework. Studies on such systems highlight both their efficiency gains as well as any downsides such as inflexibility

Table 2.2: Comparison of Optimization and Non-Optimization Algorithms in Transportation Systems

Aspect [53] [54]	**Optimization Algorithms/Models**	**Non-Optimization Models**
Definition	Use advanced mathematical techniques to find near-optimal solutions for transportation problems.	Use straightforward, often heuristic-based approaches without formal optimization processes.
Efficiency	Highly efficient in terms of minimizing travel time, costs, and resource usage.	Less efficient; may result in higher travel times and costs due to lack of optimization.
Adaptability	Can adapt dynamically to changing conditions and real-time data inputs.	Static; does not easily adapt to changes or real-time data.
Complexity	High computational complexity; requires significant computational resources and expertise.	Lower complexity; easier to implement and manage.
Implementation Cost	Higher initial cost due to complexity and need for specialized software/hardware.	Lower initial cost; can use existing infrastructure and simpler software tools.
Scalability	Scalable; can handle large-scale transportation networks with multiple variables.	Limited scalability; may struggle with large-scale networks and complex scenarios.
Real-Time Application	Effective in real-time applications, providing quick responses to dynamic conditions.	Ineffective in real-time applications; static nature leads to slower responses to changes.
User Satisfaction	Higher user satisfaction due to optimized routes and schedules, leading to reduced travel time and cost.	Lower user satisfaction due to longer travel times and higher costs.

or difficulty meeting diverse user demands [74].

2.2.2 Decentralized Community-Based Systems

Decentralized transportation systems consist of many independent agents or entities (e.g. drivers, local authorities, or private companies) who make decisions based on local information and individual objectives. Such systems often utilize modern technologies like blockchain or distributed ledgers for coordination without central authority—which may make the systems more resilient due to no single point of failure and more responsive to changes in local conditions and demands. However, this lack of central coordination may sometimes result in suboptimal outcomes (e.g. traffic jams due to mismatched signals).

Data management in decentralized schemes is challenging and is used based on cloud computing and fog in literature reviews. The authors in references [75] [76] designed a model to check traffic conditions by taking advantage of fog infrastructure. Each node can obtain data and process it locally without consulting the centre for decision-making. This type of model is suitable for the dynamic development of transportation systems today. The authors in [77] suggest a central transportation model that would make it easier for the passenger to choose the transportation means that are suitable for their trip based on their preferences and current location. The model is dependent on cloud infrastructure and on applications that provide services such as locating favourites with the help of a geosocial network. The logarithm of the route calculation depends on the various services that the cloud delivers. The main framework of the cloud computing-based carpool services can be as shown as Figure 2.1 in reference [2].

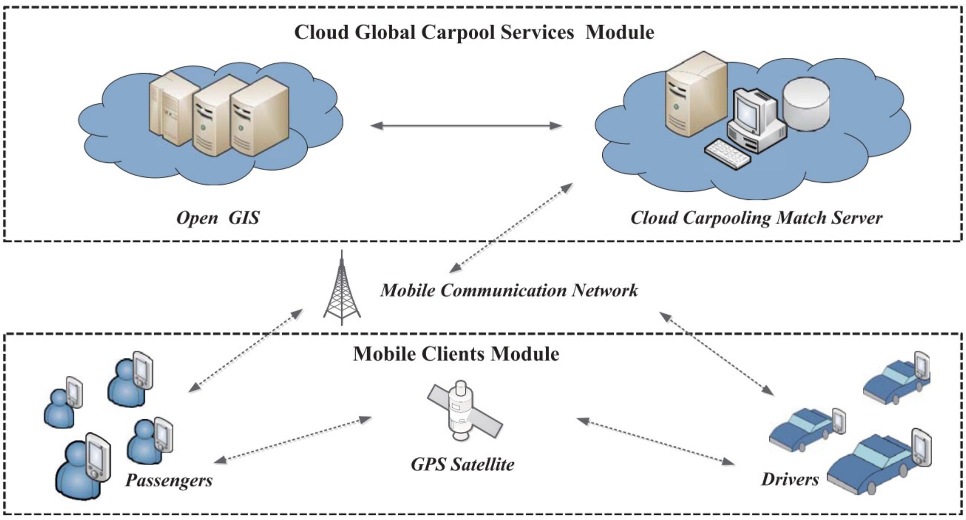

Figure 2.1: The Cloud Computing-based Carpool Services Framework inefficient in [2]

There are many characteristics of the community-based system when compared with the centralized system, as illustrated in the table 2.3.

However, community-based systems have faced some challenges. Sustainability can be difficult without enough ridership or funding available, necessitating careful management [78]. Also, navigating transport regulations designed with larger systems in mind may present additional obstacles. Additionally, limited resources may negatively impact service quality, availability, and frequency.

2.3 A Multi-Mode Transportation Systems

Multi-mode transportation systems are networks that aim to integrate modes of transportation in order to facilitate travel for both passengers and goods. These systems bring together options such as buses, trains, bicycles, ferries and pedestrian pathways to create a travel experience. The main objective is to connect and coordinate these modes of transportation.

There are reasons why multi-mode systems are important and have benefits over single-mode systems, and we can see these in table 2.4.

2.3.1 Challenges

Besides the advantages, there are also some challenges:

- Complexity in Planning and Management: Coordinating between modes of transport and agencies involved can be challenging.
- Cost: Establishing and maintaining modal infrastructure can be financially demanding. Effectively managing data across systems becomes crucial when multiple modes of transportation are involved. This includes passenger information, schedules, and operational data. The integration of this data is vital to providing real-time updates, ticketing options, and seamless travel experiences.
- Integration Issues: Integrating transportation modes comes with challenges. It requires infrastructure like inter-modal terminals. It also necessitates operational coordination, such as aligning schedules and coordinating fare systems.

Table 2.3: Comparison of Community-Based and Centralized Transportation Systems

Aspect [60] [79]	**Community-Based System**	**Centralized System**
Local Focus	Transportation systems are tailored to the specific context, culture, and needs of local communities.	Designed with a broad focus, aiming to serve larger, often diverse populations with uniform services.
User-Centric Systems	Prioritizes convenience, affordability, and preferences of local users over systemic efficiency.	Emphasizes systemic efficiency and operational consistency across the network.
Participatory Governance	Decisions on routes, schedules, and operations include input from local residents or user groups.	Decisions are made by central authorities with limited direct input from local communities.
Flexible and Adaptive	Highly adaptable, quickly responding to changing community needs or preferences.	Less adaptable, with more rigid structures and slower response to changes in local needs or conditions.
Diverse Funding Sources	Funded by a mix of sources, including local government, non-profits, community organizations, and user fees.	Primarily funded by central government budgets and fare revenues.
Integrating with Larger Networks	Can act as feeders into major transit hubs, complementing larger transportation networks.	Functions as the primary provider of transportation services, with less integration with local systems.
Service to Underserved Areas	Often serves communities or populations not well-served by traditional public transit, providing essential access.	Focuses on high-demand routes and areas, potentially neglecting underserved or low-demand areas.
Economic Benefits	Strengthens local economies by employing local residents and meeting their specific needs.	Economic benefits are broader and less targeted, focusing on overall efficiency and cost-effectiveness.

Table 2.4: Comparison of Multi-Mode and Single-Mode Transportation Systems

Aspect [80] [81]	**Multi-Mode Transportation System**	**Single-Mode Transportation System**
Commuter Choices	Offers a range of choices, reducing the reliance on cars.	Limited to one mode, increasing reliance on that mode (e.g., cars).
Efficiency	Improves overall transportation efficiency by optimizing the use of different modes.	Efficiency is limited to the capacity and performance of a single mode.
Sustainability	Contributes to the sustainability of urban transportation by encouraging the use of eco-friendly modes.	Often less sustainable, especially if the single mode is car-based, leading to higher emissions.
Congestion Mitigation	Plays a significant role in mitigating congestion by distributing commuter traffic across multiple modes.	Typically results in higher congestion, as all traffic is concentrated on a single mode.
Emission Reduction	Reduces emissions by promoting the use of public transit, biking, and walking.	Higher emissions, particularly if the primary mode is motor vehicles.
Accessibility	Enhances accessibility in urban areas by providing multiple transport options.	Limited accessibility, especially for non-drivers or those without access to the single mode.

- Additionally, there are policy and regulatory challenges to ensuring that all modes adhere to safety and service standards.
- Cost: Establishing and maintaining modal infrastructure can be financially demanding.

2.4 Time-Based Optimization

2.4.1 Time-Based Optimization in One-Mode Transportation System

Most transportation systems are interested in finding the route by which the passenger reaches their target as quickly as possible compared to other travel options. The algorithm in [82] helps to reduce the extra travel time resulting from a vehicle detour when a new sharing request occurs during the trip.

In addition to the ride-sharing transportation mode, advanced transportation systems are concerned with the development of dynamic public transport and bicycle sharing in smart cities. The model in [83] is based on the idea that the search for the shortest route of public transport should be reduced to a specific area for each car after filtering the requests that would reduce the quality of service provided to the passenger. This system works by creating an efficient path planning greedy algorithm strategy that helps speed up complex calculations. But, this study does not consider the passenger's walk from home to the station. The author in [84] (meta-heuristic) developed an approach called an ant path-oriented carpooling allocation based on the ant colony optimizer to solve the carpooling service time window problem. This model provides the shortest route to avoid congestion compared to an initially planned route.

Some of the most important problems for passengers while using buses are the wait time at the bus stop, the possibility of offering their seat to another passenger, and the cost of the trip. The first study provides a smart application to automatically provide the right route for the passenger with the possibility of obtaining a ticket online. The passenger can use his smartphone or another device to reserve the ticket online, make sure that there is a seat available on the bus with the possibility of getting the seat automatically through the proposed logarithm, and be informed of the expected wait time [85].

One important issue in car sharing is quickly finding a companion to travel with. In [86] , a mixed system was proposed, allowing the driver to drop off and pick up passengers at the same time, thereby enabling the passenger to set the time window. To solve the dynamic ride-sharing problem in smart cities, two proposed optimization techniques were introduced in this system:

symmetry breaking and linearization. To implement these techniques in the system, the authors have created randomly moving vehicles and people depending on the real map information. From the results, they note that the symmetry breaking model is superior to the other two models as it achieved faster access and increased the number of passengers who benefited from the service by 50%. Another aspect of these studies is the establishment of models that help improve the ride-sharing services, such as the trip cost, the time to reach the destination, and users' satisfaction. The authors assume that all drivers between a specific source and destination will follow the same route. Also, the most notable shortcomings in this paper are that the drivers cannot change their route, and the passengers do not have the opportunity to choose their preferences for the trip.

Some works are based on ride-share route planning strategies but with specific limitations, such as common destination and origin. In general, these problems are either single-origin-multi-destination or multi-origin-multi-destination scenarios. In fact, multi-origin-multi-destination trips are considered more complex than single-origin-multi-destination trips [87] [88].The authors in [87] suggested a new technique for calculating the shortest ride-share route to reach the target with optimization constraints. The route would be changed to meet each request, and then the appropriate route change restrictions would be computed. The authors in [88] (modern hybrid) developed a dynamic taxi-sharing model for taxi sharing based on hybrid-simulated annealing (HSA). This new technique requires complex calculations, especially when the number of requests is large. These complex calculations lead to a vast number of repetitions and multiple random disturbances.

Some studies, such as [89], focused on solutions for large-scale real-time ride sharing. In this research, a kinetic tree algorithm is developed that deals with dynamic requests for scheduling. Compared with other models, the kinetic tree has proven to be more efficient and flexible in providing suitable route options with a faster arrival time for passengers. The paper [90] designed a new route query for optimal multi-meeting points in a real-time ride-sharing environment. This route query aims to find the shortest route from the starting point to the destination by minimising the ratio between the distance of the route and the shortest route to the destination.

The authors in [91] used the Lagrangian decomposition approach to create a new ride-sharing model in which the system assigns passengers to taxis while determining the best routes for the journey. In the proposed method, the problem was formulated as a mixed-integer program with the development of two heuristic logarithms. The results were later compared with the use of

CPLEX to solve the same problem. The comparison shows that the model with the Lagrangian decomposition approach is more efficient than CPLEX, especially in relation to computation time.

Researchers in [92] developed a new ride-sharing system that aims to reduce the number of cars, and the distance travelled during the trip. Each trip set consists of the passengers, the vehicle, and certain requirements such as detour distance, number of stops, vehicle capacity, preferred routes, destination, and source. The study proved that reducing the number of cars and distance is difficult if any requirements are not met. Therefore, a new approximate algorithm was developed to achieve optimization when the number of stops is not satisfied, and the other conditions are met. This work limits the number of stops a driver can make to pick up riders.

Reference [93] developed the real-time On-Demand Bus Routing Problem (ODBRP) system as an extension of the static transportation system. To obtain a better quality of service and increase the profits for the on-demand bus company, passengers must submit their requests far in advance. Dynamic requests are added one by one with real-time optimization execution after scheduling static requests. Finally, static optimization is performed to obtain an initial solution suitable for the dynamic part.

The authors in [94] created a dynamic framework aimed at reducing the time taken for the entry process into ride-sharing applications. The model aims to achieve three main goals: minimizing the total travel time of the passengers and the sum/maximum flow time of the requests. By leveraging the Fenwick tree, the results from the experiment in this reference show that the insertion operator is on2 instead of on3. In terms of numbers, the insertion operator can be accelerated by 998.1 times on urban area datasets.

Some studies focus on providing a system that helps to calculate the shortest routes for the driver and the passenger dynamically. For example, [95] is based on the fact that there are common subways for the passenger and the driver between two points, namely the location from which the passenger is picked up and the location where the passenger is dropped off. Here, the authors must recognize that the passenger's route may include some paths where public transport is used before or after the common sub-paths. This model is called the 2 Synchronization Points Shortest Path Problem (2SPSPP). Here, the authors choose the optimal pick-up and drop-off points as well as the concurrent common routes between the driver and the passenger, which achieve the least time to complete the total journey. To guide the logarithm using heuristics, the authors identify possible areas of the pick-up and drop-off.

The reference [96] addresses the one-to-many ride-sharing matching problem with unpredictable trip times between destinations. The objective is to build effective ride-sharing matching systems that minimize the cost of driver detours and the number of matching passengers. The authors frame the ride-sharing matching problem as a Robust Vehicle Routing Problem with Time Window constraints (RVRPTW) to do this. To effectively capture trip time uncertainty, the authors offer a data-driven, deep learning-based method to dynamically predict journey time uncertainty sets. The authors assume that, for each matched driver, no passenger may be dropped off until all passengers have been picked up. The driver cannot pick up additional passengers once the drop-off sequence has begun.

The paper [97] proposes a method for effectively managing the schedules of buses (EBs) operating on routes. The focus here is on optimizing scheduling and charging processes so as to minimize waiting times, variations in usage levels, costs associated with charging, as well as any expenses related to charging. The aim is to ensure that battery degradation occurs evenly while also cutting expenses; for this, the authors employ the branch and price method. Tests show that this approach effectively decreases waiting times, charging costs, as well as variations in usage levels.

2.4.2 Time-Based Optimization in a Multi-Mode Transportation System

When we combine several network graphs into a single graph and then apply the routing algorithm, we obtain a multi-modal route plan. The system in [98] is a piece of multi-modal technology that does not require the passenger to specify restrictions on the use of different modes of transport prior to the start of the trip. For example, the passenger can travel by car and then decide to take the bus. This model differs from [52] which uses the nearest neighbor algorithm since the processing time is faster. A popular method for combining various networks is to solve the nearest neighbour problem (NN), which directly joins spatially-close stops from diverse networks. In addition, the system is implemented so that it can solve any inquiry locally.

In [99], a mobility system was established to combine carpooling and traditional multi-modal transportation in the same trip in real time. The basic idea here is based on the principle of replacing some sub-routes on the traditional multi-modal transport route with carpooling paths. The intention is to reduce the passenger's travel duration. To achieve this, the implementation stages of the system go through three primary steps. First, a group of potential drivers is selected by determining the closest probability between the journey of the passenger and the drivers provided. And then calculate configurations of faster routes for drivers and design a substitution process. In the end, the system chooses the near-optimal solution to achieve the quickest time of arrival of a set of possible and logical solutions. This work is a development based on [100]. Here, the system focuses on the number of stations rather than the number of potential drivers. This concept helps reduce the number of launched Dijkstra algorithms. This work only focuses on reducing trip time, not on reducing both time and cost. Most modern transportation systems create multi-origin-multi-destination strategy transportation models with sophisticated search techniques and lots of information uploaded to the cloud.

Some studies have focused on the many-to-many carpooling problem with multiple cars and various passenger preferences in the real world, such as [101]. In this study, a model is developed to determine the optimal time trip interval for the carpooling problem with the help of prematch information (CPPPMI) and time-space network flow techniques. As a result, this model is very useful for any CPPPMI problems in the smart city and valuable for any researchers designing a plan for carpooling with matching information. The model in this study does not address the random trip time that could be added to the actual scenario. Different from the work presented in [101], the reference [102] uses the Genetic Algorithm (GA) to solve combinatorial optimization problems targeting fairness and passenger security improvement. Many other projects, such as in [103] developed CLASCOON with a software platform for real-time dynamic carpooling services matching algorithm. This software is easily used by the passenger to find a journey companion quickly. Additionally, two essential features are proposed by CLACSOON: first, the dynamic urban partial ride-sharing, which allows the passengers to reach their destination during the journey by walking. The second feature makes it possible to share a ride after a trip has started. The result of the simulation shows that CLACSOON reduced the trip time by 55 % and the CO2 by approximately 10 % compared to others in comparison to current carpooling service models.

Many studies in the last decade have focused on route planning in transportation networks. The authors in [104] outlined the essential techniques used to create a multi-mode transportation system. This study addressed some algorithms for calculating the shortest paths of road networks and performance metrics related to dynamic road networks [105] [106]. In [107], the developers proposed a multi-transit system consisting of public transit and ride-sharing based on the angle-based clustering [AC] algorithm. The proposed model has been compared with the exact solution, as the comparison shows that the AC algorithm performs well in both large and small settings. Through the experience, it is clear that the multi-modal transportation system with ride-sharing achieves higher efficiency in trip time and distance. The developers of this system did not include in their design plans the possibility of a random addition in trip time and wait time as a result of unexpected factors that continuously occur in reality.

Most transportation systems are interested in finding the route by which the passenger reaches their target as quickly as possible compared to other travel options. The reference [108] proposed to integrate dynamic ride-sharing into public transport. The authors suggested a system called Xhare-a-Ride (XAR) to provide the passenger with dynamic movement in the shortest possible time while considering limitations such as route change and walking preference.

Very few current transportation systems include more than one means of transport as an option for the passenger during his journey. Reference [109] targets dynamic multi-network mobility in urban areas. The research suggests a model to find solutions for traffic flow with the participation of carpooling, public buses, and trains within the framework of a single traffic system. The developers in [3] designed a mobile platform that offers suggestions for planning a trip that includes only two modes of transport, carpooling and public buses as shown in Figure 2.2. This solution ensures the shortest route for the passenger while considering delays caused by road accidents and traffic congestion. In [110], the purpose of the system is to combine two modes of transportation (ride-sharing and car-sharing) in one model so that the passenger can operate the multi-trip scheduling as one task. We note the shortcomings in this work. Most modern models focus on only one aspect, for example reducing time without taking into account

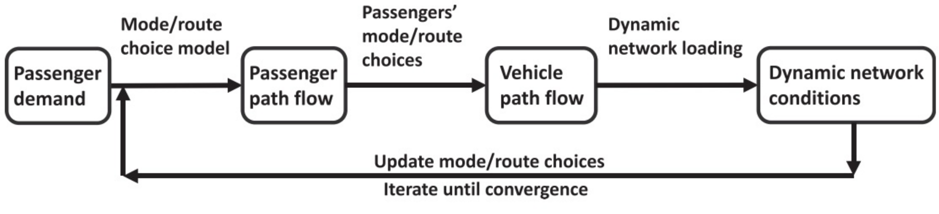

Figure 2.2: The whole process for solving MMDUE in [3]

lowering costs. Also, the paper focuses on one or two modes of the transportation system without considering the diversity in terms of user needs.

Many papers suggest multi-mode transportation systems to find solutions to some transportation problems, without taking into account the trip's time and cost. Tripod's optimization framework in [111] is a new rider order management system that aims to reduce the energy consumed by the transportation system as a whole. The passenger here manages the online request as the model supports multiple means of transport and depends on giving the passenger the best option based on the current and predicate state of the network. However, this model has several limitations. First and most importantly, it does not take into account the passenger's desire to arrive at the lowest cost (time and price). Secondly, the simulation is applied in the morning period only. Reference [112] has proposed a framework for multi-modal transportation networks to find a clearer picture of the different patterns and predictions of urban transport. The model consists of three layers of varying transportation networks as different coverage of these networks is studied for passengers in three different cities. However, mathematical details for this model do not calculate the time and cost required for the trip; as well, their effect on coverage is not studied.

The most current multi-modal routing systems are moving away from the combination of public transportation scheduling with walking and car sharing. This shift is due to the complexity and difficulty imposed by the dynamic movement of transport with the need to update the location of the passenger and the vehicle continually and to update the tracks in the complex systems. In [113], the authors created a multi-modal routing system based on previous information stored in the user's profile with a range of possible travel options. This work aims to reduce complications and computational effort for the multi-modal transportation model through the use of heuristic schemes. Similarly, many studies have developed methods to identify meeting points for car drivers and riders [114] [115]. [114] aimed to build a dynamic, decentralized and integrated system that maintains the privacy of the passenger. The meeting points are calculated in the ride-sharing service so that each user can continue to control their location information. The ride-sharing model in this paper is based on the assumption that there is only one driver and one passenger. In [115], a new site-based system was proposed for suggesting meeting points for passengers and drivers in the real world. In this scenario, it is assumed that the driver drives his car through the city's main streets to be able to pick up one or more passengers from a single meeting site. At the same time, passengers are expected to walk or use public transport to reach the agreed assembly point.

In the study [116], the developers designed a proper system for mobility-on-demand so that the passenger has the freedom to choose the type of transportation he or she prefers while maximizing the utility. In addition, the developers have integrated Bayesian optimization to optimize the supply aspect (fare size and fleet). Finally, taxi data from New York City was used to study the results of applying this model.

To merge ride-sharing with fixed-route transit, the authors in [117] devised a labelling system to match drivers and passengers. The method uses a schedule-based transit shortest path to build ideal itineraries for users, accounting for various transport network complications such as accurate waiting time, in-vehicle time, transfer time, and access time. Then, by using a matching optimization program, riders and drivers are paired for the initial mile of the rider's journey. The task has two distinct objectives: maximizing the total number of matches and maximizing the total reduction in vehicle hours. The authors considered the capacity of vehicles to be limitless. However, overcrowded transit systems frequently face the issue of customers being unable to board due to limited capacity. Also, they assumed that the trip time on both road and transport networks is dependable. However, passenger trip duration is frequently subject to unpredictability. This uncertainty increases with bus transportation services.

2.4.3 Time-Based Optimization in Game Theory One-Mode Transportation System

In [118] , the Bush-Mosteller (B-M) Reinforcement Learning (RL) system is developed to represent the route selection behaviours of travellers in traffic networks who find optimal transit routes that reduce individual trip time. The Nash equilibrium of the congestion game reveals the ideal approach for route selection. The congestion game is applied to the traffic assignment issue by constructing a novel potential function (TAP). The findings of a numerical experiment based on the Nguyen-Dupuis network reveal the efficacy of the theoretical analysis and that the B-M RL-based solution approach performs better than numerous current methodologies.

The authors in [4] explore the analysis and enhancement of traffic flow in road networks using Warsaw as a case study, as shown in Figure 2.3. The primary focus is on optimizing traffic in urban areas. The contributions of this study involve 1) proposing an algorithm to model transportation systems and optimize traffic flow and 2) providing a discussion on the Nash equilibrium and Stackelberg approach to examine the effectiveness of optimization techniques in reducing travel times and increasing traffic flow. However, it is important to acknowledge the limitations of this study, such as its nature and its inability to account for random events. Future research will focus on developing models that incorporate logic for decision-making purposes.

2.5 Fare-Based Optimization

2.5.1 Fare-Based Optimization in One Mode Transportation System

Other studies have only focused on finding solutions to reduce the cost of the trip. The authors of [119] proposed a price model that provides fair pricing to the riders based on each rider's profile so that the service provider makes a good profit without being circumvented. The dynamic ride-sharing pricing system in [120] is based on a double auction-based discounted trade reduction mechanism consisting of a Balanced Budget (BB), Individual Rationality (IR), and Incentive Compatibility (IC).

Researchers in [121] developed a rigorous approximation framework for shared vehicle systems, providing a consistent method for a variety of controls (rebalancing, matching, and cost); objective

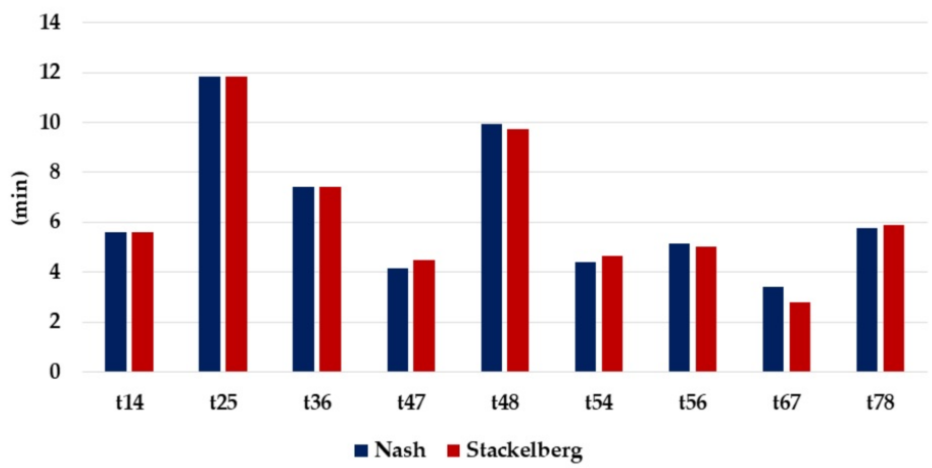

Figure 2.3: Compared travel times through individual links to road network [4]

functions (welfare, revenue, and throughput); and constraints (posted fare, welfare benchmarks, and trip times). Based on the analysis of natural convex relaxations, they provided as special cases existing approximate optimal policies for constrained settings, asymptotic optimality results, and heuristic policies. The results are non asymptotic, and parametric guarantees provide functional insights into the development of real-world systems. The developers assume that consumers' sensitivity to surge pricing is independent of their destination, meaning that all customers arriving at a node have similar value distributions. They demonstrate that, under this assumption, the elevated flow relaxation for the point-pricing problem can be solved with a single eigenvalue computation for throughput/welfare; however, for the issue of maximizing revenue, the elevated flow relaxation must be solved in conjunction with a one-dimensional concave near-optimal solution.

The work in [122] investigates the technique design challenge to encourage rider involvement in on-demand first-mile ride-sharing, considering riders' mobility desires, such as riders' requirements on arrival deadlines, maximum willing-to-pay fees, and detour tolerance. The unexpected, on-demand traveller requests are managed using a rolling horizon strategy. The research assumes that the journey time between two places is deterministic. In fact, the travel time is unpredictable and dependable. Also, this research does not predict the journey time, and it uses one station only.

The reference [123] focuses on the difficulties involved in managing bike-sharing systems based on stations. This research aims to introduce user-centred recommendation strategies that improve both the user experience and system performance. The problem at hand is how to efficiently suggest bicycle stations to users based on factors like distance and the availability of bikes or slots. To address this issue, the authors have implemented the expected resources and expected cost strategies along with expected cost and future impact, which optimize bike distribution while reducing abandonment rates.

The paper [124] investigates how autonomous and electric vehicles (AVs/EVs) may fit into shared transportation systems. A primary goal in vehicle fleet selection should be finding solutions which reduce transportation expenses when taking into account factors like travel patterns, operational expenses, Vehicle Miles Travelled (VMT), travel time, and ownership expenses. This paper's goal is to develop and compare metaheuristic algorithms, including a genetic algorithm and simulated annealing, in order to address complex transportation issues at hand.

2.5.2 Fare-Based Optimization in a Multi-mode Transportation System

The authors at [125] developed a new system and performed data analysis using simulation with questionnaires to study the effect of passengers' preferences, such as the time and cost of the trip, on the maximum profit for Customized Bus (CB) and Ride-Sharing (RS). A competitive game model is designed to achieve maximum profit for each mode of transport by considering passengers' choices which depend mainly on the utility function to fulfil the requirements of the passengers to reach the destination in the shortest possible time. Through the results obtained, the optimal fare for the bus is a non-decreasing function of the fixed factor and linear coefficient of RS when the CB provider knows the RS pricing strategy. The optimal RS price is a quadratic function of the price of CB when the bus price strategy is known to the car provider. Within this multi-transportation model, the passenger can choose only two types of transportation. This is unlike the other model we proposed, which includes three transportation options.

The reference [126] presents a dynamic passive pricing model for cyclists. Passengers may receive a refund if they reach the designated bike area, but the bikes are undersupplied. The proposed model balances the price that the user deserves and his or her path while travelling within the city by bus, for example. It is noteworthy how this design allows the passenger to use one or more modes of transportation during his or her journey. There are several shortcomings in this proposed system compared to our multi-transport system. First, the model mentioned in this research does not include ride-sharing as an option. Second, the number of transfers between types of transportation is limited to only two. Third, there is no mixing in the traffic routes for all kinds of transportation. Finally, the model developers suggest that regardless of preference, if there is no bicycle, the passenger can either walk or take the bus.

The reference [127] is one of the rare studies that have designed a model which allows the passenger to share his or her trip with others while simultaneously calculating the cost of the trip. By using a special application, the passenger can choose the route of his or her trip, including main and secondary roads and type of participation. As a result, through the numerical examples that have been clarified in this reference, toll charges and passenger rewards influence the itineraries and satisfaction of passengers by choosing the appropriate mode of transportation.

The developers of this model did not give the passengers the option to choose their preferred departure time, which affects the ride-sharing dynamic.

During the COVID-19 crisis, the public transport sector was greatly affected. This effect is reflected by the number of passengers and the high operating cost due to the imposition of social distancing in public transportation and regular cleaning. Reference [128] demonstrates the flexibility of using transportation systems that integrate fixed means of transportation, time, and movement, such as public buses, with on-demand transportation service. This study concludes that multi-systems are more flexible than one-way systems, especially in terms of the cost and comfort provided to passengers.

The reference [129] presents a multimodal ride-sharing framework for enabling last-mile connection for public transport passengers. For first or last-mile connections, the integrated framework provides customers with the most suitable mode of travel or a combination of modes. Furthermore, the functional components of the framework and the infrastructure requirements for large-scale deployment are detailed. Specifically, the communication protocols and distributed computing paradigm are elucidated.

The work [5] simulates a multimodal network with ride-sharing services. The developed model re-creates the scenario in which travellers with their cars can choose to be solo drivers, ride-sharing drivers, ride-sharing passengers, or public transit riders. In contrast, travellers without cars can only choose to be ride-sharing passengers or public transit passengers. This paper further develops a doubly dynamical system that investigates the day-to-day dynamics of within-day time-dependent travellers' decisions and traffic conditions. In contrast with our research, where drivers can pick up multiple passengers in one car, this study assumes that each ride-sharing driver can only pick up one passenger in the trip.

The [130] introduces the Tradable Credit Scheme (TCS) to promote people's travel choices in areas with various transportation options and to address traffic congestion and carbon emissions related to transportation systems. The goal is to optimize travel expenses and lower carbon footprints by offering individuals incentives to change their travel decisions.

2.5.3 Fare-Based Optimization in Game Theory Transportation Models

The demand for self-driving electric cars is growing globally, which explains why there is so much research in this regard. The authors in [131] suggested a payment strategy for charging to help developers find optimal places to put the required resources on the roads. This paper, which is based on the multi-agent framework, uses non-cooperative game theory to build its proposed model.

The paper [132] focuses on the use of the coalitional game for designing a new dynamic pricing method which is concerned with basing fare distribution on the distance travelled by each passenger. Equitable distribution of the price encourages people to carpool and reduces the use of private cars, which ultimately reduces the emission of carbon dioxide into the atmosphere. However, the work does not provide a passenger-matching algorithm, and so the optimal utility of all individuals cannot be processed.

In [133] the authors suggest a game theory-based pricing solution to the ride-sharing dilemma of taxi commuters that addresses the near-optimal solution of their travel companion and successfully lowers their cost. This paper proposes two stable matching techniques: Best Time Sharing (BT) and First-Come, First-Served (FCFS). The analysis revealed that the stable matching method lowered the total number of trips while maintaining the dataset's temporal frequency of trips. In addition, the best time pairing and game theory pricing approaches were the most cost-effective in terms of passenger savings. However, the simulations of ride-sharing only considered two passengers and the possibility of three or four people sharing a vehicle was not considered. The evaluation window is barely 30 minutes per day and does not include peak travel times.

Reference [134] suggests a model for surge pricing in ride-hailing platforms from a temporal perspective, highlighting riders' and drivers' strategic behaviour and the fact that drivers react to spike fares far more slowly than commuters. Despite considerable anecdotal evidence, passengers' and drivers' strategic behaviour has not been formally examined in the literature. In this study, the authors adopted and examined a traditional two-period game-theoretical model described in the literature on strategic customer behaviour.

In [135], the authors investigate the coordination of the porcelain supply chain in the realm of e-commerce, specifically focusing on packaging. The study tackles the challenge of aligning the supply chain while considering consumer preferences for packaging as well as its impact on transportation losses and after-sales issues. The main objectives of this study encompass analyzing how packaging influences these aspects by comparing decision-making models, proposing a contract, and emphasizing the role of packaging in reducing losses and enhancing the entire supply chain. To achieve this, our study employs modelling and analysis to compare decentralized decision-making approaches. The outcomes reveal that centralized decision-making leads to profits and better packaging levels. Additionally, the researchers introduce a contract that enhances coordination among stakeholders. However, it is important to acknowledge the limitations of this study such as incorporating the nature of packaging and its environmental impact while also exploring applications in other industries. Ultimately, the result of this study underscores the importance of striking a balance between safeguarding packaged goods and addressing environmental concerns.

The reference [136] utilizes level optimization as a means to choose between rail and road transport for freight transport, taking environmental and social considerations into account as the main goals. This study develops a model in which a rail operator takes charge, and shippers follow the operator's decisions. A programming model was then created that takes into account the objectives of increasing profits while decreasing risk and emissions. To facilitate this work, the authors employ the Karush-Kuhn-Tucker approach to solving this multilevel model. This study presents examples and sensitivity analyses to evaluate the model and understand which factors have an effect on decision-making by both rail operators and shippers. The findings show that investing in infrastructure may help increase market share for rail operators. Enhancing line capacity and train frequency can increase profitability and attract new customers, with shippers often opting for rail transportation if cost is their primary concern. However, it should be noted that this study has its limitations, since it assumes only one rail operator and shipper to represent real world freight transportation systems accurately. Additionally, the research fails to take into account factors like unpredictability in demand or specific time restrictions.

The authors in [137] introduce a pricing technique that focuses on pairings in cities using game theory. The main goal is to lower travel costs for passengers while ensuring drivers still earn an income. The investigators address the issue of commuting and the need to optimize ride-sharing capabilities. For the pricing model, the paper's proposal aims to decrease fares and minimize the number of taxi trips. The authors analyze how intangible factors influence ride-sharing decisions. The authors' findings demonstrate that implementing the game theory-based pricing scheme resulted in a reduction in required taxi trips and a slight increase in travel distance for passengers. Moreover, it had an impact on driver revenue when it came to passenger fares. It is important to note that this pricing model primarily aims to benefit riders through cost reduction. However, it is essential to recognize the limitations of this study, such as relying on urban mobility datasets that may not cover all scenarios and factors affecting ride-sharing decisions. Additionally, the authors did not investigate the effects of this pricing model on stakeholders such as taxi companies or the broader transportation system.

The reference [138] focuses on addressing the challenges of integrating equity considerations, pricing mechanisms, and network expansion decisions into the design of transportation networks. One major issue identified is the lack of models that can effectively incorporate network design, congestion pricing, and equity maximization in their calculations. The primary goal of this paper is to alleviate congestion and ensure the allocation of resources among user groups. To achieve this, a bilevel optimization model is proposed involving three players: an equity maximizer, a network operator, and travellers. This model utilizes an approach of Mathematical Programming with Equilibrium Constraints (MPEC) to formulate and solve a Stackelberg game, which is then transformed into a Mixed Integer Linear Program (MILP). Numerical experiments conducted on a transportation network in Sydney showcase the capabilities of this model. Different objective functions and scenarios are analyzed, with the results revealing tradeoffs between equity, travel time, travel cost, congestion levels, and investment value. This paper underscores the importance of considering equity during transport network design while also suggesting avenues for research; however, its limitations are not explicitly stated.

2.6 Time-Fare-based Optimization in Transportation System

In [139] the author used ant colony optimizing to find the near-optimal solution for the bus school routing problem in Bogotá, Colombia. The result of the experiment, which used real data, shows that the new model leads to a reduction in the time interval of the bus trip and cost in comparison to other models. With the student arriving home at an earlier time, the model improves their quality of life. This work assumes that there are a limited number of buses available to provide the service. In [140] (meta-heuristic), a multi-loading school bus routing model with four meta-heuristics was applied to solve a new complexity in the multi-loading issue. The older version permits students from various schools to ride the same bus at the same time. However, the scheme in this paper enables the bus to take students to or from school simultaneously. This model gives us lower-cost transportation with flexible routing.

Some works are based on ride-share route planning strategies but with specific limitations, such as common destination and origin. In general, these problems are either single-origin-multi-destination or multi-origin-multi-destination scenarios. In [141], a model was developed that calculates the shortest route and best cost for a single-origin-multiple-destination trip.

Some projects have focused on transportation enhancement, targeted cost and trip-sharing. BlaBlaCar provides travellers with the opportunity to share their trip and its cost with others. The company website and application connect drivers with travellers to satisfy their requests [142].

The authors [143] assume that a private company has developed a platform to provide Intercity Ride-Sharing (IRS) services for passengers travelling between two towns. The IRS Vehicle Routing Problem (VRP-IRS) is defined on a directed graph and formulated as a mixed integer linear programming problem in this paper. They propose a variable neighbourhood search algorithm to solve the VRP-IRS, which is NP-hard. Four rider-based local search operators and four trip-based neighbourhood operators are suggested to find more useful neighbourhood-based solutions.

The authors of [144] postulate and demonstrate four features, namely "scheduling preferability," "financial sustainability," "preference-based incentive compatibility," and "preference-based individual rationality," in order to achieve the four incentive objectives, respectively. This online hybrid mechanism consists of a dynamic re-optimization method for re-matching and rerouting and a hybrid real-time pricing mechanism that differentiates between different types of passengers. To obtain large-scale solutions for the online hybrid mechanism, this research adapts the Solution Pooling Technique (SPA) initially suggested in our prior work for a static offline method. The main limitation of this paper is the journey time between two sites is assumed to be constant.

The reference [145] focuses on enhancing the efficiency of routing in transit systems that have fixed time windows. The main objective is to find a way to allocate vehicles so as to meet passenger demand while minimizing costs and time penalties. The goal of this research is to develop a model and an adaptive genetic algorithm that can optimize routing and improve transit systems effectively. The key contributions of this study include proposing an optimization model for routing and utilizing an algorithm. This study has some limitations, such as not accounting for dynamic trip demands, which require investigation for validation purposes. Additionally, this paper provides references related to optimization algorithms used for transportation problems, including tabu search and dynamic programming, among others.

2.7 Objective-Based Optimization in Transportation System

In [146] the School Bus Routing Problem (SBRP) is addressed by designing a particular model using a genetic algorithm to solve the most critical issues in this type of problem, such as route schedules, bus route generation, and stop selection of bus. The result showed the superiority of this algorithm to the previous algorithm, especially since it took into account most of the constraints, such as school arrival time and vehicle and bus capacity.

In some studies, such as [147], the authors propose a shared route percentage (the ratio of the shared route's distance to the driver's total travel distance) model. In this model, the driver shares a vehicle with the rider. For that, the driver first drives to the rider's source, then to the rider's destination, and, finally, to his or her own destination. Here, the vertices are the drivers and the riders; the edges are the driver-rider pair, and edge weights are the driver-rider's SRP. The method is based on the calculation of the upper and lower bounds for each rider-driver pair at a fixed time.

Dial-a-ride service is a bus service that operates in a mode partway between a normally scheduled bus service and a taxi; it is a form of demand-responsive transport. The system typically has a scheduled route, but passengers can call and book a pick-up within an area served by the route, and then the bus route is modified to make the pick-up. Drop-offs anywhere within the area can also be accommodated. The aim is to extend public transport services to the front door of all residences or from any place to any place. Some services operate exclusively for disabled or elderly passengers; other services are open to the general public [148]. Recently, the trend in dynamic ride-sharing research has been to create traffic systems that find the shortest route for the passenger while considering the passenger's preferences, such as trip price. SHAREK in [149] is a model that allows the driver to provide his or her passenger with a sharing service by determining the vehicle's current location and distance from the user. At the same time, the user specifies his current position, the distance to his last destination, and the maximum amount he is willing to pay for his trip. However, since SHAREK deals with only one group of passengers, this limits the use of the system's ride-sharing capabilities. The taxi sharing system in [150] allows each user to determine the payment, the trip's length of time, and the waiting time. The authors in [151] proposed a system to develop the current passenger sharing system to consider the current traffic situation in order to give passengers the preferred options for their journey. One of the drawbacks of this system is that the proposed model does not deal with unexpected road issues in actual situations. The paper [152] designed utility-aware ride-sharing on road networks, a dynamic movement model based on maximising the vehicle-related utility and the rider-related utility without exceeding capacity-constrained and time-constrained riders.

For some existing models, some of their functions are designed to be centralised while others are decentralised. [153] compares a centralised and a decentralised algorithm to find the best match between a rider and a driver while satisfying the lowest cost for dynamic ride sharing. The decentralised algorithm builds on the auction-based design concept. When the authors compare these two algorithms, they note that the decentralisation model achieves greater cost savings than the central model and also reduces the overall trip distance. In the work of Duan et al. [154] developers tackle the problems of self-driving taxis by developing a model that handles passenger hybrid orders. The passenger can request the ride immediately or in advance. The central processing requests short-distance rides while decentralising deals with long-distance requests. In the result of the experiment, it is noted that the computational efficiencies of decentralisation are improved concerning service quality and economic productivity.

The work [155] proposes an efficient and accurate algorithm called PeerMatcher. This algorithm presents reduced complexity in terms of finding a trip plan; however, its communication range is reduced and it cannot cover a high number of transportation forms. Based on its scalability, accuracy and efficiency, PeerMatcher reduces the level of complexity and the amount of time needed to find a solution. The nodes are clustered into groups that have a specific fixed size, which is a drawback if it is used in dynamic communication. The authors in [156] suggest a fully decentralised model so that the agents in the network make decisions individually with the possibility of communicating with the rest of the neighbours and exchanging information without consulting the central agent. Through this model, an ideal profit is achieved at the network level, even if the covered area within this proposed system is extensive. The authors in [157] proposed the creation of distributed software based on the concept of decentralisation upon which this study is based. This proposal focuses on cases in which the passenger receives several offers from drivers, a process which is termed multi-hop ride-matching. The proposal enhances the dynamic mobility that smart cities require while providing convenient options for the passengers. In [158], a virtual payment system is proposed for the driver so that he can choose which passengers he will serve. The decentralised system here is auction-based, where the driver selects the most profitable passenger. Thus, companies benefit from the service fees affiliated with the driver, and the driver earns more income because he has the freedom to choose the user without referring to the centralised company's authority. In the Ridematcher system in [159], the driver–passenger matching process is distributed to fog to achieve a complete decentralisation system. Passenger–driver pairs are chosen depending on the car's route and the time taken for the trip. The model proposed in this work was developed to fit only the ride-matching from the beginning to the end of the route and does not consider if there is partial matching during the trip.

The Vehicle Sharing Program (VSP) in the transportation system includes cars, bicycles, or electric vehicles. The developers in [160] designed a system to determine the best configuration of VSP. The design considers several issues, such as parking location and parking capacity. The algorithm (bi-level, mixed-integer program) allows the passenger to choose the appropriate means of transportation to the stop that is closest to his or her final destination. The results from the simulation show that this system achieves a better balance between the network operators and the passengers' goals in comparison to other systems and thus an increase in the efficiency of the transportation network as a whole [161].

Developers in [162] studied a capacitated Vehicle Routing Problem with Time Windows and simultaneous Pickup and Delivery (VRPTWPD) by developing a Demand Responsive Transit System (DRTS). This model connects to the route to determine the optimal schedule to meet passenger requirements and address the problem of limited empty seats on public buses. This study focuses on rural areas, especially crowded routes where a bus-type taxi service has replaced buses. This model will help route planners to provide the appropriate number of highly efficient means of transportation. Also, this model allows the government to reduce the amount of support it provides to the transportation company.

Ride-sharing platforms have economic and social effects that reflect positively and negatively on society. To study these effects scientifically and realistically, the developers in [163] developed an analytical model based on the information available from the current transportation providers. This study shows that ride-sharing services should be available when their pooling efficiency and quality are high. In general, this service benefits the government and riders, but it may harm the overall income of drivers.

The author in [164] examines the variety of Vehicle Routing Problems (VRP) and the restriction on admitting products that meet a minimum quality standard. This study attempts to develop and optimize a mathematical model that brings the quality issue of a perishable product into the distribution process. The value of a product declines as its quality deteriorates. This problem is described theoretically as a Mixed Integer Nonlinear Programming Problem (MINLP). A genetic algorithm-based heuristic is suggested due to the computational complexity involved in using the model to tackle real-world problems. The proposed method is utilized for numerical case resolution and sensitivity analysis. To make the situation more realistic, traffic congestion should be integrated.

The work [165] defines a new optimization model, Online Matching with Controllable Rewards and Arrival probabilities (OM-CRA), to find the matching strategy and rewards and arrival probabilities dynamically. The authors present a quick 1/2-approximation algorithm for OM-CRA, despite our problem being more difficult than existing ones. By estimating the objective function, the proposed method transforms OM-CRA into a saddle-point problem, which is then solved by the Primal-Dual Hybrid Gradient (PDHG) method accelerated by the problem design.

In [166], the authors consider the development of a system in which users could be asked online to walk to/from close pick-up/drop-off places if doing so increases the overall effectiveness of the ride-sharing system. The authors demonstrate theoretically that the general problem grows increasingly complicated (as it comprises two sub-problems that expand set-cover). The authors analyze the resulting trade-offs and give a general formulation and specific methods for the solution of large cases. In this paper, there are some limitations. First, the authors assume that there is no overcrowding. Second, only one mode of transportation is used. Integrating SMoD and public transportation in building a unified transit network is both challenging and promising, and it would be reasonable to include walking in the on-demand subsystem.

The researchers in [167] investigated the ride-sharing issue with a personalized matching window. Specifically, they proposed an event-driven framework, E-Ride, which maintains request groups via a state graph augmented by a shareability graph. Then, they suggest an effective request cluster enumeration system based on the k-clique in the shareability graph, which improves the request clusters by the effective arrival of subsequent requests.

The reference [168] focuses on the case of a single vehicle that must follow a predetermined request of requests. The authors provided the mathematical programme, demonstrated that it can be solved in polynomial time, and proposed a faster heuristic. They compare the optimal algorithm, the heuristic, and routes that visit the actual request locations, demonstrating that avoiding detours can decrease vehicle costs by more than one-third and overall costs by nearly one-fifth. Furthermore, the heuristic produces competitive outcomes. Simulations over Manhattan's existing road network indicate that the time savings obtained by the heuristic may be necessary for the system to perform in real time.

In [169], a new algorithm called Intelligent Train Operations (ITO) is proposed to optimize the speed trajectory of freight trains. The ITO algorithm involves initialization, calculation, storage, and selection steps for optimizing control sequences through training and updating processes. It outperforms methods like Genetic Algorithm (GA), Fuzzy Predictive Control (FPC), and field test data while also highlighting the need to improve the algorithm's robustness and consider longitudinal coupler force uncertainty. However, there are limitations that require further research on coupler forces to improve the overall robustness of the algorithm.

The reference [170] discusses the challenges faced by logistics service providers with service agreements when they encounter situations in which not all customer orders are known in advance. The main objective of this study is to develop an algorithm that can optimize routing and scheduling plans efficiently when there is uncertainty involved. To achieve this, branch and bound techniques are employed to solve the Vehicle Routing Problem with Stochastic Demands and Customers (VRPSCD). The algorithm focuses on integrating customer planning with service agreements to minimize costs based on computed results. Computational analysis demonstrates that this approach outperformed benchmark methods in terms of cost savings, although computation times were longer, for instance. Nevertheless, the algorithm managed to provide solutions within its designated time limit.

2.8 Objective-Based Optimization in Game Theory Transportation Models

Recently, many smart city developers have considered the adverse effect of energy consumption on the surrounding environment. The author in [103]developed an algorithm (series game variation of the urban game) to reduce the jam on the roads by decreasing the number of cars and therefore decreasing the CO2 emissions. The model was implemented using a spatial data mining model on statistical data collected from the great office hubs in Poland. As a result of this model, the number of cars and the amount of energy consumption were decreased by twelve per cent in comparison to previous models.

Authors in [171] created a new theory with the help of the game theory concept of social networks. This theory is a decentralised system where the utility of the node depends on the local information of the node's neighbours.

In [172] the authors propose a novel approach to ride-sharing that aims to reduce congestion and pollution. The approach involves individuals walking to a central pick-up location and riding together to a single drop-off point before continuing on foot to their final destination. The authors use game-theoretic principles to determine the optimal composition of riders, pick-up and drop-off points, and an equitable method for allocating costs among participants. The cost-allocation method in this paper is based on the principle of proportionality, ensuring that riders who walk more pay less. Unlike the models in this thesis, this study limits the number of stops to one stop only and does not allow for multiple stops along the route.

In [173], the developers attempted to study the routing game on the flow over time by examining the effect of passengers' individual decisions on network efficiency. Specifically, this study establishes a system based on competitiveness within the price of anarchy. Stackelberg's strategy is applied to formulate the boundaries of the total delay price of anarchy and the boundaries of the time price of anarchy.

To create a distinctive design for freight and freight transportation, the developers in [174] developed a model based on the dual-stage game theoretic approach by merging the dual-stage stochastic programming approach with the Stackelberg game. This system facilitates the transfer of decentralized chains, especially for those working under uncertain conditions. In this study, the ticket reservation system for flights that may or may not be delayed is used as a practical example of the designed system. In contrast to studies in which a central unit is used to calculate the cost of congestion in transportation systems, the authors in [175] developed a decentralized system for calculating the cost of congestion. First, the Nash and Stackelberg model was used to study each of the regions under the transportation network. Next, it was found that competition between regions increased the system's efficiency as a whole while giving the best route and least fare congestion cost compared to other systems (local region scheme).

Despite the importance and wide application of the Stackelberg approach in several areas of economics or management, most studies still focus on static bilevel optimization problems. [176] models use discrete-time dynamic Stackelberg games for finding different solutions to problems. In this paper, there are two types of followers in the game: independent followers and dependent followers.

The reference [177] is based on Stackelberg's model to plan an equilibrium bike-sharing network. The designer is the leader, and the passengers are the followers. The final design of the stopping site and all other resources related to the bicycle network must meet the user's requirements in terms of flexible movement within the city and reaching the destination in the least amount of time. Furthermore, the design must fulfil the requirements of the system planner in establishing a network at the lowest possible cost. This balance between the user and the designer will not be achieved without continuous communication throughout the design period. The user changes the route of his journey whenever there is a change in the design to achieve the maximum possible use of the resources available.

Using Stackelberg's model, a real-time pricing scheme was developed in a smart grid with multiple residential users and retailers. It is noted that price competition among the population has been formulated as an evolutionary game and as a non-cooperative game between members of retail management. In addition, two algorithms for the equilibrium solution were developed. Finally, the numerical results of the experiment confirm the efficiency and effectiveness of the scheme presented in this paper in comparison to other previous works [178].

The developers in [179] created a dynamic pricing system to solve the car shortage of ride-sharing service providers. This system is based on a dynamic Stackelberg leader-follower game and is formulated as a multi-period MPEC model. In addition, a non-myopic Approximate Dynamic Programming (ADP) system was developed to transform a dynamic Mathematical Program with Equilibrium Constraints (MPEC) at each decision stage into a solvable Mixed-Integer Quadratic Program (MIQP). Several numerical experiments show that the proposed system significantly increases the ride-sharing system's efficiency in the dynamic settings of intelligent transportation systems.

Conclusion

In summary, the presented models and strategies from existing research have significantly contributed to advancements in transportation optimization. However, our proposed approaches address critical gaps by integrating game theory, optimization algorithms, and multi-mode (ride-sharing, bus, and bicycle) adaptability. This foundation sets the stage for more efficient and flexible urban transportation systems, as demonstrated in our models' superior performance under various simulated conditions.

Chapter 3 Core Elements in Multi-Mode Transportation Models: DMT, GT-MMT, and DGT

3.1 overview

This chapter explores the fundamental techniques and strategic frameworks used in the dynamic mobility traffic (DMT), game theory multi-mode transport (GT-MMT), and decentralised game-theoretic (DGT) models. These models are further explored in more detail in chapters 4, 5, and 6, respectively. Although the GT-MMT and DGT utilize a Stackelberg-genetic model, their approach diverges: the GT-MMT adopts a centralized infrastructure, whereas the DGT emphasizes a community-based framework. Notably, the DGT includes pedestrian walking times, which is absent in the GT-MMT model.

The DMT is a dynamic multi-mode (Ride-sharing and bus) model and primarily uses genetic algorithms (GA) and particle swarm optimization (PSO) to address the complex problem of route optimization. GA effectively explores a wide search space and finds global optima, making it suitable for initial solution generation. PSO demonstrates exceptional performance in optimizing these solutions in real-time, adapting quickly to changing traffic conditions and passenger preferences. The primary benefit of using GA and PSO in the DMT model is their combined ability to handle large, dynamic datasets and provide robust solutions.

The GT-MMT is also a dynamic multi-mode (Ride-sharing, bus, and bicycle-sharing) model that integrates GA and PSO with Stackelberg game theory. In this model, the passenger acts as the leader, and the transportation modes (ride-sharing, bus, and bicycle-sharing) act as followers. The GA and PSO algorithms are used to optimize the routing to give the passenger the best mode combination with minimum travel time and fare. The Stackelberg framework adds a layer of hierarchical decision-making, ensuring that passenger choices influence the strategies of the transportation modes. This combination allows for more efficient system-wide coordination and optimization. The main benefit is the strategic alignment between passenger choices and mode operations, enhancing overall system efficiency. However, the integration complexity and the need for rapid real-time adjustments pose significant challenges.

The DGT model also uses GA and PSO with the Stackelberg framework but with a decentralized control mechanism. Each transportation mode independently collects and processes data, making decentralized decisions. Stackelberg game theory is used here to model the interactions between passengers (leaders) and modes (followers). GA generates initial strategies, while PSO refines these strategies based on local performance metrics and shared data. The decentralized approach improves scalability and robustness to local disruptions. However, achieving global optimization is more challenging, and the coordination between modes without central control can lead to sub-optimal decisions in dynamic environments.

All three models utilize the same simulation tools but are differentiated by varying parameter values tailored to each model's specific requirements. This enables a sophisticated adjustment of the models to various urban mobility scenarios. The GT-MMT and DGT designate the passenger as the leader, with buses, car ride-sharing, and bicycles as followers. The DGT uniquely incorporates walking as an essential component of its strategy, highlighting its community-based infrastructure. In contrast, the DMT, while also a multi-modal transportation system, eschews the complexity of game theory in favour of straightforward GA and PSO approaches and does not incorporate bicycle-sharing.

This analysis highlights each model's distinct contributions to the efficiency and adaptability of urban transit. By mentioning their common features and unique qualities, this chapter aims to thoroughly comprehend how integrated and adaptable transportation models can be customized to effectively address the varied requirements of urban areas.

The remainder of this chapter is organized as follows: section 3.2 (Variables and Parameters for the Three Models); section 3.3 (Topology of Multi-Modal Transportation Models: DMT, GT-MMT, and DGT); section 3.4 (Common Objectives and Constraints for the Three Models); section 3.5 (Common Optimization Algorithms for the Three Models); section 3.6 (Simulating Scenarios for DMT, GT-MMT, and DGT Models).

3.2 Variables and Parameters for the Three Models

In our thesis, the three models (DMT, GT-MMT, and DGT) encompass many common sets and variables, which we explained in the first subsection 3.2.1. The additional common variables between the GT-MMT and DGT in the second subsection 3.2.2. But, the specific parameters and notations for each model are located in chapter 4 for the DMT, in chapter 5 for the GT-MMT, and in chapter 6 for the DGT. The full list of the parameters and notation in the List of symbols part 2.

3.2.1 Common Variables and Parameters for the Three Models

1. Passenger Notations

Each rider's request, designated as p_i within the set R, comes with its own specifics. This includes the source point $S_1^{p_i}$, the last destination $D_{np}^{p_i}$, and intermediate stops $I_r^{p_i}$ for the passenger index p_i. We also have a defined earliest departure time, denoted as Ed_{p_i}, from the source point $S_1^{p_i}$, and the latest arrival time, denoted as La_{p_i}, at the last destination point $D_{np}^{p_i}$. Two constants, AT_{p_i} and BT_{p_i}, are provided to establish each rider's maximum travel time, MT_p. Consider $t(S_1^{p_i}, D_{np}^{p_i})$ as the straight-line travel time from location $S_1^{p_i}$ to $D_{np}^{p_i}$. The calculation for each rider's maximum travel time is $MT_{p_i} = AT_{p_i} + BT_{p_i} * t(S_1^{p_i}, D_{np}^{p_i})$. Besides the maximum travel time, we need to define the maximum fare trip denoted as MF_{p_i}, MW_{p_i}, which is the maximum passenger waiting time, and WT_{p_i}, which is the actual passenger waiting time. For each source location $S_1^{p_i}$ and destination location $D_{np}^{p_i}$, a time frame is established: $[F_{S_1^{p_i}}, E_{S_1^{p_i}}] = [Ed_{p_i}, La_{p_i} - t(S_1^{p_i}, D_{np}^{p_i})]$ which represents the earliest and latest departure times for the passenger, and $[E_{D_{np}^{p_i}}, F_{D_{np}^{p_i}}] = [Ed_{p_i} + t(S_1^{p_i}, D_{np}^{p_i}), La_{p_i}]$ represents the earliest and latest arrival times for the passenger. So, each p_i has many attributes: $p_i = \{S_1^{p_i}, D_{np}^{p_i}, Ed_{p_i}, La_{p_i}, MT_{p_i}, MF_{p_i}, MW_{p_i}, WT_{p_i}\}$.

2. Ride-sharing Notations

In ride-sharing, R is a set of user requests at a given time, $Tr_{p_i}^{c_j}$. Every vehicle, represented by c_j within the set C, comes with specific attributes. These include Ca_{c_j} capacity of the vehicle c_j, a starting location $S_1^{c_j}$, intermediate stops $I_a^{c_j}$ for the ride-sharing index c_j, an end location

$D^{c_j}_{nc}$; the maximum waiting time is MW_{c_j}; the acceptable time detour in the trip is σ_{c_j}; and additionally, there's a demand $Nump(S^{c_j}_1, D^{c_j}_{nc})$ that outlines the total number of passengers to be transported from the pickup point $S^{c_j}_1$ to the point $D^{c_j}_{nc}$. The trip route is expressed as $R = \{c, S^{c_j}_1, D^{c_j}_{nc}, MW_{c_j}, \sigma_{c_j}, Nump(S^{c_j}_1, D^{c_j}_{nc})\}$.

Additionally, there's the earliest departure time, denoted as Ed_{c_j}, from the starting point and the latest arrival time, denoted as La_{c_j}, at the destination. There are also four constants, AT_{c_j}, BT_{c_j}, AD_{c_j}, and BD_{c_j}, utilized to calculate the vehicle's maximum travel time MT_{c_j} and maximum travel distance MD_{c_j}. The computation of time intervals for points $S^{c_j}_1$ and $D^{c_j}_{nc}$ ($[E(S^{c_j}_1), F(S^{c_j}_1)]$ and $[E(D^{c_j}_{nc}), F(D^{c_j}_{nc})]$), which represent the earliest and latest departure times and the earliest and latest arrival times for vehicle c_j respectively, is conducted in the same manner as for the riders' locations.

The computation of the maximum travel distance for a vehicle c_j is MD_{c_j} performed similarly to the riders. If $d(S^{c_j}_1, D^{c_j}_{np})$ is the direct travel distance between two points $S^{c_j}_1$ and $D^{c_j}_{nc}$, then the maximum distance a vehicle can travel is defined as $MD_{c_j} = AD_{c_j} + BD_{c_j} * d(S^{c_j}_1, D^{c_j}_{nc})$. So, each c_j has many attributes: $c_j = \{S^{c_j}_1, D^{c_j}_{nc}, Ca_{c_j}, Ed_{c_j}, La_{c_j}, MT_{c_j}, MD_{c_j}\}$.

3. Public Bus

The public bus schedule determines the route of each vehicle and the time required to reach the destination. The schedule for each bus usually consists of a bus numbering index (b_k), a set of stations (S_{b_k}), a capacity of the bus (Ca_{b_k}), and a set of routes for the bus (R_{b_k}). All the above parameters can be written as $b_k = \{S_{b_k}, Ca_{b_k}, R_{b_k}\}$. A route $r_{b_k} \in R_{b_k}$ can be written as $r_{b_k} = \{S^{b_k}_1, D^{b_k}_{nb}, dt^{b_k}_{s_u}, at^{b_k}_{s_u}, l^{b_k}_{s_u}\}$. Here, a bus is $b_k \in B$; the starting station is $S^{b_k}_1$, the intermediate stop is $I^{b_k}_u$ for the bus index b_k; and the destination station is $D^{b_k}_{nb} \in SB$; the departure time is $dt^{b_k}_{s_u}$ and the arrival time is $at^{b_k}_{s_u} \in TB$, and the number of the passengers inside the bus on each route is $Nump^{b_k}_{e_x}$. The computation of time intervals for points $S^{b_k}_1$ and $D^{b_k}_{nc}$ ($[E(S^{b_k}_1), F(S^{b_k}_1)]$ and $[E(D^{b_k}_{nb}), F(D^{b_k}_{nb})]$), which represent the earliest and latest departure times and the earliest and latest arrival times for vehicle b_k respectively, is conducted in the same manner as for the riders' locations.

4. Trip Notations

Let's assign $O = \{1, ..., np\}$ to represent the collection of pickup locations of the passengers and $D = \{n+1, ...2n\}$ to represent the group of delivery locations. The union of all pickup and delivery points is signified by $E = O \cup D$. Further, when we include the starting and ending locations for vehicles into this set, we get $A = E \cup \{S_1^m, D_{nm}^m\}$ for every m in set M. Consider $Nump_{s_n}^m = l_{s_n}^m$ and $Nump_{s_{n+i}}^m = -l_{s_{n+i}}^m$ to indicate the change in the number of the passengers at locations s_n and s_{n+i}, respectively. Consider $T_{s_n}^m$ as the indicator for when service begins at point s_n^m by vehicle m. It represents the passengers' pickup and delivery times and the vehicles' start and end times (drivers). The time required for service at location s_n^m within the set E (either for pickup or delivery) is represented by $st_{s_n}^m$. We introduce a binary decision variable $x_{s_n}^m$, which is set to 1 if vehicle m provides service at location s_n and then moves directly to location s_{n+i} for service. If this condition is not met, the variable is set to 0.

Besides all the previous common parameters for the three models (DMT, GT-MMT, and DGT), for all three models (DMT, GT-MMT, and DGT), the primary output is the optimal combination of transportation modes for each passenger. This combination, denoted as $\mathcal{C}_{p_i}$ for passenger p_i, is represented as an ordered list of tuples, where each tuple $(m_{nm}, S_1^{p_i}, D_{np}^{p_i}, TT_{p_i}, TF_{p_i})$ contains the transportation mode $m_n m$, the source $S_1^{p_i}$ and destination $D_{np}^{p_i}$ stops, the total trip time TT_{p_i} and the total trip fare TF_{p_i}. The objective is to minimize both the travel time (TT_{p_i}) and the fare (TF_{p_i}), ensuring that passengers can complete their trips in the shortest time possible at the lowest cost. By leveraging advanced optimization algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Stackelberg game theory, the models effectively provide tailored solutions that enhance the overall efficiency and cost-effectiveness of the transportation network. The notations used include T for travel time, C for cost, R for ride-sharing stops, B for bus stops, and K for bike-sharing stations, with subscripts to denote specific passengers (p), vehicles (v), and stops (S). The final outputs for all of the three models can represented as equation 3.1.

$$
\left[\begin{array}{cccc}
\text{Passenger} & \text{Vehicles} & \text{Trip Route} & \text{Total Trip Time and Fare} \\
\hline
1 & \left(m_1, m_2, ...m_{nm}\right) & \begin{pmatrix} \text{Source: } S_1^{p_1} \\ \text{Destination: } S_{np}^{p_1} \end{pmatrix} & \left(TT_{p_1}, TF_{p_1}\right) \\
2 & \left(m_1, m_2, ...m_{nm}\right) & \begin{pmatrix} \text{Source: } S_1^{p_2} \\ \text{Destination: } S_{np}^{p_2} \end{pmatrix} & \left(TT_{p_2}, TF_{p_2}\right) \\
\vdots & \vdots & \vdots & \vdots \\
n & \left(m_1, m_2, ...m_{nm}\right) & \begin{pmatrix} \text{Source: } S_1^{p_i} \\ \text{Destination: } S_{np}^{p_i} \end{pmatrix} & \left(TT_{p_i}, TF_{p_i}\right)
\end{array}\right] \tag{3.1}
$$

3.2.2 Additional Common Variables and Parameters for the GT-MMT and the DGT

For both the GT-MMT and the DGT, we integrate bicycle sharing as a third transportation mode system in addition to ride-sharing and public bus modes. The full list of the parameters and notation in the List of symbols part 2.

In both the GT-MMT and DGT models, bike-sharing is a crucial mode of transportation, with several important parameters and variables to manage and optimize the system effectively. Each bike is denoted as bk_n, where n represents the specific bike instance. The journey of a bike begins at a source stop, represented by $S_1^{bk_n}$, and ends at a destination stop, denoted by $D_{nbk}^{bk_n}$. The total travel time for the bike trip is represented by $TT_{p_i}^{bk_n}$, while the cost incurred for the trip is denoted by $TF_{p_i}^{bk_n}$. Additionally, the bicycle carries only one person, so the capacity of it Ca_{bk} equals 1. The time intervals for points $S_1^{bk_n}$ and $D_{nbk}^{bk_n}$ ($[E(S_1^{bk_n}), F(S_1^{bk_n})]$ and $[E(D_{nbk}^{bk_n})$ $, F(D_{nbk}^{bk_n})]$), which represent the earliest and latest departure times and the earliest and latest arrival times for bike bk_n respectively. The earliest departure time for the bike-sharing mode in the GT-MMT and the DGT can correspond to the service's operational hours. The bike-sharing system has specific times during which bikes are available for rent and return. Also, the bike-sharing mode has a maximum arrival time corresponding to the end of the service's operational hours. This latest arrival time is the time by which bikes need to be returned to a station to avoid additional fees or ensure they are available for maintenance and redistribution. The users in our proposed models have specific closing times for returning the bike.

3.3 Topology of Multi-Mode Transportation Models: DMT, GT-MMT, and DGT

The topology and structure of road routes play a crucial role in the optimization and management of multi-modal transportation systems. The design and analysis of these systems often involve defining nodes, edges, stops, pickup, and delivery points.

The three models integrate multiple modes of transportation, including ride-sharing, buses, and bicycles. The network topology in the three models is represented as a graph, with nodes representing various stops, which can be source stops, intermediate stops, or destination stops. Each node is a point where passengers can board or take off from a vehicle. Source stops are the starting points of a journey, intermediate stops are points where passengers can switch modes or continue their journey, and destination stops are the final points of a journey.

Bus stops and bike-sharing stations serve as intermediate points for passengers to board or alight from buses and bicycles. These stops are crucial for last-mile connectivity and enhancing accessibility. Ride-sharing points facilitate the pickup and drop-off of passengers using ride-sharing services, providing flexibility and convenience.

Edges represent the routes or roads connecting these stops. Each edge Ed_x indicates a direct path between two nodes. However, indirect Routes Ei_x are routes that include one or more intermediate stops between the source and destination nodes. Pickup points are locations where passengers are picked up by ride-sharing or buses, while delivery points are locations where passengers are dropped off. This structure allows the three models to effectively manage and optimize the flow of multiple transportation modes within an integrated network.

Nodes (stops, stations, and ride-sharing points) are connected by edges (routes) to form an interconnected network that allows passengers to travel from one point to another efficiently. The connectivity between nodes and edges ensures that there are multiple paths available for passengers, enhancing the network's robustness and flexibility.

Different modes of transportation, such as buses, bicycles, and ride-sharing, are integrated within the network to provide passengers with a variety of travel options. Integration methods include synchronized schedules, common ticketing systems, and shared information platforms to ensure a seamless travel experience. The three models provide real-time information to passengers about the status of different modes and expected transfer times to ensure smooth transitions between modes.

Integrated transportation networks in smart cities are urban areas with strategically placed multi-modal hubs that facilitate easy transfers between buses, bicycles, and ride-sharing, improving overall connectivity and reducing travel time.

The part of the City of Ottawa, as we can see from Figure 3.1 for the DMT model and from Figure 3.2 for the GT-MMT and DGT models, represents the topology of our work.

3.4 Common Objectives and Constraints for the Three Models

In our thesis, the three models (DMT, GT-MMT, and DGT) have many common objectives and constraints, which we explained in the subsection 3.4.1 and 3.4.2. The additional common objectives and constraints between the GT-MMT and DGT only in the subsection 3.4.3 and 3.4.4. But, the specific objectives and constraints for each model are located in chapter 4 for the

Figure 3.1: Map of Ottawa for the DMT model

Figure 3.2: Map of Ottawa for the GT-MMT and DGT models

DMT, in chapter 5 for the GT-MMT, and in chapter 6 for the DGT.

3.4.1 Common Objective Functions for the Three Models

This subsection mentions all the common objective functions and constraints used by the three models (DMT, GT-MMT, and DGT).

The main objective for all three models is to minimize the total trip time (TT_{p_i}) and the total trip fare (TF_{p_i}) for the passenger (3.2). This is achieved by choosing the optimal combination of transportation modes based on real-time data and personal preferences. The passenger's decision affects the demand dynamics for each transportation mode, guiding their operational strategies.

$$\min\left(\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}\right) \tag{3.2}$$

where α_{p_i} and β_{p_i} are the weighting factors for travel time and cost for passenger p_i, respectively.

Dynamic routing involves adjusting routes in real time based on passenger requests and road conditions:

$$RO(S_1^m, D_{nm}^m) = \text{minimize}\left(\sum_{I_r^{p_i} \in SR} d(S_1^m, I_r^{p_i}) + \sum_{I_r^{p_i} \in SR} d(I_r^{p_i}, D_{nm}^m)\right) \tag{3.3}$$

where $RO(S_1^m, D_{nm}^m)$ defines the optimal route from vehicle source S_1^m to vehicle destination

D^m_{nm}, $d(S^m_1, I^{p_i}_r)$ and $d(I^{p_i}_r, D^m_{nm})$ are distancing from the vehicle source S^m_1 to passenger point $I^{p_i}_r$, and from passenger point $Sr^{p_i}_r$ to the vehicle destination D^m_{nm}, respectively. By optimizing these elements, ride-sharing services can offer more efficient routes that are both faster and potentially cheaper than fixed-route alternatives, making them more attractive to passengers.

In the DMT model, genetic algorithm (GA) and particle swarm optimization (PSO) are used to minimize trip time and fare for passengers. In the GT-MMT and DGT models, GA, PSO, and Stackelberg game theory are combined to achieve these objectives. Passengers act as leaders in making route decisions, while transportation modes adjust strategies accordingly. The trip time and fare equations provided by these algorithms ensure efficient and cost-effective travel for passengers. The details of the equations, algorithms, and operations are illustrated in chapters 4 for the DMT model, 5 for the GT-MMT model, and 6 for the DGT model.

The total trip time (TT_{p_i}) for the passenger in the three models can be seen in the equation 3.5:

$$TT_{p_i} = \beta_{p_i} \sum_{p_i \in R} \sum_{I^{p_i}_r, D^{p_i}_{np} \in SR} T^m_{D^{p_i}_{np}} - st^m_{I^{p_i}_r} - T^m_{I^{p_i}_r} + WT_{p_i} \tag{3.4}$$

Where, $T^m_{D^{p_i}_{np}}$ is the time when the passenger arrived at his destination, $st^m_{I^{p_i}_r}$ is the time required to serve the passenger at point $I^{p_i}_r$ by vehicle m, and $T^m_{I^{p_i}_r}$ is the time of starting service at point $I^{p_i}_r$ by vehicle m

In the DGT, we consider the walking time ($Twalking_{I^{p_i}_r}$), so the equations for the total trip time become:

$$TT_{p_i} = \beta_{p_i} \sum_{p_i \in R} \sum_{I^{p_i}_r, D^{p_i}_{np} \in SR} T^m_{D^{p_i}_{np}} - st^m_{I^{p_i}_r} - T^m_{I^{p_i}_r} + WT_{p_i} + Twalking_{I^{p_i}_r} \tag{3.5}$$

To calculate the total fare TP_{p_i} for the passenger, we use the following equation:

$$TF_{p_i} = \sum_{j=1}^{nc} TF^{c_j}_{p_i} \times x_{c_j} + \sum_{k=1}^{nb} TF^{b_k}_{p_i} \times x_{b_k} + \sum_{n=1}^{nbk} TF^{bk_n}_{p_i} \times x_{bk_n} \tag{3.6}$$

$TF^{c_j}_{p_i}$, $TF^{b_k}_{p_i}$, and $TF^{bk_n}_{p_i}$ are the total fare for the passenger when he rides the car ride-sharing, public bus, and bicycle, respectively. x_{c_j}, x_{b_k}, and x_{bk_n} are binary indicators (0 or 1) denoting whether the mode is used during the trip or not.

Bicycle sharing is not incorporated as a transportation option in the DMT model. So the equation (3.7) becomes:

$$TF_{p_i} = \sum_{j=1}^{nc} TF_{p_i}^{c_j} \times x_{c_j} + \sum_{k=1}^{nb} TF_{p_i}^{b_k} \times x_{b_k} \tag{3.7}$$

The trip fare of the passenger, when he rides the ride-sharing, is calculated by Equation (3.8), where $Fbase_{c_j}$ is a fixed base fare cost applied to all rides, Fd_{c_j} is the fare rate per unit distance (kilometre), $TD_{p_i}^{c_j}$ is the total trip distance of the passenger when he rides c_j, Ft_{c_j} is The rate per unit time (minutes), $TT_{p_i}^{c_j}$ is the total trip time of the passenger when he rides c_j, and $Fadd_{c_j}$ is additional charges that may vary depending on specific conditions such as peak hours, special services, or tolls.

$$TF_{p_i}^{c_j} = Fbase_{c_j} + (Fd_{c_j} \cdot TD_{p_i}^{c_j}) + (Ft_{c_j} \cdot TT_{p_i}^{c_j}) + Fadd_{c_j} \tag{3.8}$$

Equation (3.9) calculates the total price of the bus trip. Here $TT_{p_i}^{b_k}$ is the total trip time for the passenger when he rides the bus, and $TF_{b_k}/hour$ is the bus fare for each passenger per hour.

$$TF_{p_i}^{b_k} = TT_{p_i}^{b_k}\ dollar \times TF_{b_k}/hour \tag{3.9}$$

To calculate the trip cost when using a bicycle in the GT-MMT and the DGT models, you should consider several factors that influence the cost directly and indirectly. Here's a general formulation for estimating the trip cost ($TP_p^{bike_i}$) when a passenger uses a bicycle:

$$TF_{p_i}^{bk_n} = Fbase_{bk_n} + (Fd_{bk_n} \cdot TD_{p_i}^{bk_n}) + (Ft_{bk_n} \cdot TT_{p_i}^{bk_n}) \tag{3.10}$$

Where $Fbase_{bk_n}$ is a fixed cost for a flat booking fee, Fd_{bk_n} denotes the rental cost per kilometre, $TD_{p_i}^{bk_n}$ is the total distance of using a bicycle, Ft_{bk_n} represents the rental cost per minutes of bicycle use, $TT_{p_i}^{bk_n}$ indicates the total time of the bicycle trip in minutes.

3.4.2 Common Constraints for the Three Models

In this subsection, common constraints could involve compliance with regulatory and environmental standards or optimizing route efficiency based on available traffic data.

To ensure balance, constraint (3.11) guarantees that the quantity of vehicles departing from their respective origins matches the number of vehicles arriving at their final destinations.

$$\sum_{s_1^m \in O_m} Num(S_1^m) = \sum_{D_{nm}^m \in D_m} Num(D_{nm}^m) \tag{3.11}$$

Constraint (3.12) guarantees that a rider's request is matched with at least one suitable option.

$$\sum_{Ma \in Match} x(p_i, Ma) \geq 1, \quad \forall p_i \in R \tag{3.12}$$

where R is the set of all rider requests, $Match$ is the set of all matching options available for the riders, and $x(p_i, Ma)$ is a binary variable that is 1 if rider p_i is matched with option Ma and 0 otherwise.

The constraint (3.13) guarantees that the time to move between two adjacent points will not exceed the minimum required time.

$$T_{I_r^{p_i}}^m + st_{I_r^{p_i}}^m + t(I_r^{p_i}, I_{r+d}^{p_i}) \leqslant T_{I_r^{p_i}}^m, \forall v_m \in V, s_i, s_j \in A \tag{3.13}$$

Constraint (3.14) guarantees that the assigned time frames will not be exceeded.

$$F_{S_1^{p_i}} \leqslant T_{I_r^{p_i}}^m \leqslant E_{S_1^{p_i}}, \forall m \in M, S_1^{p_i}, I_r^{p_i} \in A, p_i \in R \tag{3.14}$$

The precedence limitation (3.15) states that a pickup point must be visited before its corresponding drop point.

$$T_{S_1^{p_i}}^m + st_{S_1^{p_i}}^m + t(S_1^{p_i}, D_{np}^{p_i}) \leqslant T_{D_{np}^{p_i}}^m, \forall m \in M, S_1^{p_i} \in O \tag{3.15}$$

Constraint (3.16) guarantees that if a vehicle has a certain load after visiting a specific service point, it must have that load plus the load from the previous service point when it reaches the next one. This ensures that the number of passengers picked up at a location should be equal to the number of passengers dropped off at its corresponding destination.

$$Nump^m_{s_n} - Nump^m_{s_{n+i}} = 0, \forall m \in M, s_n, s_{n+i} \in A \tag{3.16}$$

The purpose of constraint (3.17) is to guarantee that the capacities of the vehicles are not surpassed and that the load of a vehicle is at least equal to $l^m_{s_n}$ after it has serviced a pickup point s_n.

$$l^m_{s_n} \leqslant Nump^m_{s_n} \leqslant Ca_m, \forall m \in M, S_n \in A \tag{3.17}$$

Constraint (3.18) guarantees that vehicles don't have any passengers when they depart and reach their destination.

$$Nump^m_{s_1} = Nump^m_{D_n m} = 0, \forall m \in M \tag{3.18}$$

Constraint (3.19) and (3.20) are implemented to guarantee that each vehicle's distance travelled and duration remain within a predetermined threshold.

$$TD_m \leqslant MD_m, \forall m \in M \tag{3.19}$$

$$TT_m \leqslant MT_m, \forall m \in M \tag{3.20}$$

The constraint 3.21 ensures that each passenger's total travel distance $TD^m_{p_i}$ does not exceed their specified maximum allowed distance $MD^m_{p_i}$.

$$TD^m_{p_i} \leqslant MD^m_{p_i} \tag{3.21}$$

Constraint (3.22) ensures that the time the vehicle takes $TT_{m_1}(y, I^{p_i}_r)$ to reach the next stop must be equal to or less than the time for the other vehicle to travel from its current location to the passenger's pickup station $TT_{m_2}(x, I^{p_i}_r)$.

$$TT_{m_1}(y, I_r^{p_i}) \leqslant TT_{m_2}(x, I_r^{p_i}) \tag{3.22}$$

Constraint (3.23) ensures the time the vehicle takes to move from its current location to the passenger's location is equal to or less than the maximum waiting time MW_{p_i}.

$$TT_m(y, I_r^{p_i}) \leqslant MW_{p_i} \tag{3.23}$$

Constraints (3.24 and 3.25) show that the total time or price for the passenger p_i when he takes any type of transportation must be less than or equal to the threshold value (MT_{p_i} *and* MF_{p_i}), respectively.

$$TT_{p_i}^m \leqslant MT_{p_i} \tag{3.24}$$

$$TF_{p_i}^m \leqslant MF_{p_i} \tag{3.25}$$

Finally, under constraints (3.26 and 3.27), the total trip time TT_{p_i} and the total trip price TF_{p_i} must be less than or equal to the maximum total time MT_{p_i} and the maximum total price MF_{p_i}, respectively.

$$TT_{p_i} \leqslant MT_{p_i} \tag{3.26}$$

$$TF_{p_i} \leqslant MF_{p_i} \tag{3.27}$$

3.4.3 Additional Objectives for GT-MMT and DGT

In this subsection, we illustrate the objective functions for both the GT-MMT and the DGT models, which are based on PSO and genetic algorithms with the Stacklberg game theory approach. The core of these models revolves around the strategic interaction between a selected leader, the passenger, and a range of followers, encompassing diverse transportation modes: buses, ride-sharing services, and bicycles. These models are designed to maximize the efficiency of the transportation network by successfully matching the preferences of passengers with the operational capabilities of each mode of transportation. This involves complex decision-making processes where each mode reacts dynamically to the leader's choices and to the competitive pressures from other modes. We use the terms utilities and strategies here because the GT-MMT and DGT models are based on game theory concepts.

1. Increasing the Number of Passengers in Ride-Sharing

Ride-sharing services in the GT-MMT and DGT aim to maximize the number of riders, as this increases revenue and efficiency. To attract more passengers, they implement dynamic routing strategies. This involves adjusting routes in real time to pick up additional passengers while ensuring minimal disruption to the current occupants' travel times. By dynamically rerouting, ride-sharing services can become more attractive to passengers by offering faster or cheaper routes compared to fixed-route modes like buses. We can represent maximizing the vacant seats in car ride-sharing as equation (3.28)

$$U_{c_j} = \max(1 - \frac{Nump_{e_x}^{cj}}{Ca_c}) \tag{3.28}$$

where $Nump_{e_x}^{cj}$ is the number of passengers in the car in route e_x, and Ca_c is the maximum capacity of the car.

2. Increasing the Number of Passengers in Buses

Buses aim to maximize the number of passengers while following a fixed route with predefined stops. To increase ridership, buses can focus on reliability and frequency. Buses can become a more reliable choice for passengers by improving schedule adherence and decreasing wait times. They can also coordinate with information from the GT-MMT model to adjust frequencies during peak hours or on high-demand routes, aligning more closely with real-time passenger needs. We can represent maximizing the vacant seats in the public bus as equation (3.29)

$$U_{b_k} = \max(1 - \frac{Nump_{e_x}^{b_k}}{Ca_b}) \tag{3.29}$$

where $Nump_{e_x}^{b_k}$ is the number of passengers in the bus b_k, and Ca_b is the maximum capacity of the bus.

To handle variations in demand, particularly during peak hours, buses can adjust their frequency:

$$f_{s_u} = \text{adjust\_frequency}(D_{s_u}^{b_k}, D_{\text{avg}}^{b_k}, \Delta T_{b_k}) \tag{3.30}$$

where $D_{s_u}^{b_k}$ is the demand at stop s_u, $D_{\text{avg}}^{b_k}$ is the average demand, and ΔT_{b_k} is the time interval for peak hours. Buses utilize data from the GT-MMT or the DGT model to dynamically adjust frequencies and enhance coordination:

$$\text{new\_frequency}_{s_u} = \text{base\_frequency} \times \left(1 + \frac{D_{s_u}^{b_k}}{D_{\text{avg}}^{b_k}}\right) \tag{3.31}$$

This formula ensures that the bus frequencies are adjusted based on real-time demand, improving the system's responsiveness and efficiency.

3. Increasing the Use of Bicycles

The primary goal is to maximize the usage rate of bicycles across the network, ensuring efficient utilization and high availability. This objective (3.32) can be formulated as follows:

$$U_{bk_n} = \max\left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)}\right) \tag{3.32}$$

where U_{bk_n} is the utility function for the usage rate of bicycles, $Numbk_{use}(I_l)$ denotes the number of bicycles in use at location I_l, $Numbk_{av}(I_l)$ represents the total number of bicycles available at location I_l, and nbk is the number of locations or docking stations.

3.4.4 Additional Constraints for GT-MMT and DGT

Both models incorporate similar constraints that govern the operations of the transport modes to achieve the objectives of the GT-MMT and DGT. These constraints typically include operational limits such as capacity, frequency, and service availability, which ensure that the solutions are viable and sustainable within the practical confines of urban transport systems. However, there are other additional constraints for the DGT, which are mentioned in chapter 6.

Constraint (3.33) ensures that the bus adheres to the scheduled times at each stop to maintain reliability. where $at_{s_u}^{b_k}$ is the actual arrival time at stop s_u, $scht_{s_u}^{b_k}$ is the scheduled arrival time, and $T_{\text{tolerance}}$ is the allowable deviation from the schedule.

$$|at_{s_u}^{b_k} - scht_{s_u}^{b_k}| \leq T_{\text{tolerance}} \tag{3.33}$$

Many constraints are used to optimize the bicycle-sharing system while ensuring service quality.

Constraint(3.34) ensures that the number of bicycles at any given station does not fall below a demand-induced threshold.

$$Nbike_{\text{av}}^{s_{bk}} - Nbike_{\text{use}}^{s_{bk}} \geq Nbike_{\text{min}}^{s_{bk}}, \quad \forall s_{bk} \tag{3.34}$$

$Nbike_{\text{min}}^{s_{bk}}$ is a constant representing the minimum number of bicycles that must be available at each docking station s_{bk} to ensure that the station can meet demand. It is a safety stock level that prevents the station from running out of bicycles. The variable $Nbike_{\text{av}}^{s_{bk}}$ represents the total number of bicycles at docking station s_{bk} at a given time, while $Nbike_{\text{use}}^{s_{bk}}$ represents the number of bicycles currently in use or reserved at docking station s_{bk}.

$$Ma_{s_{bk}} + Re_{s_{bk}} \leq Opt_{max}^{s_{bk}}, \quad \forall s_{bk} \tag{3.35}$$

where $Ma_{s_{bk}}$ and $Re_{s_{bk}}$ are maintenance and redistribution activities at location s_{bk}, respectively, and $Opt_{max}^{s_{bk}}$ is the maximum allowed operations per station.

$$Fma_{s_{bk}} + Fre_{s_{bk}} \leq Tcost_{max}^{s_{bk}}, \quad \forall s_{bk} \tag{3.36}$$

where $Fma_{s_{bk}}$ and $Fre_{s_{bk}}$ are the costs associated with maintenance and redistribution at location s_{bk}, and $Tcost_{max}^{s_{bk}}$ is the budget for operational costs.

Constraint (3.37) shows that the bicycle must be available at its station for the passenger to be able to use it on his journey. If it is available, the av_{bk_n} equals 1; otherwise, the value is zero.

$$av_{bk_n} = \begin{cases} 1, & \text{if } \textit{the bicycle is available} \\ 0, & \text{otherwise} \end{cases} \tag{3.37}$$

Dynamic re-balancing to redistribute bicycles according to varying demand patterns, which can be modeled as a constraint (3.38) Where $Numbk_{re}(s_j, s_{bk})$ is the number of bicycles relocated from station s_j to station s_{bk}, and $Nbike_{\min}^{s_{bk}\,\prime}$ is the adjusted demand at station s_{bk} after redistribution.

$$Numbk_{av}^{s_{bk}} + \sum_{s_j=1}^{n} Numbk_{re}(s_j, s_{bk}) - \sum_{s_k=1}^{n} Numbk_{re}(s_{bk}, s_k) = Nbike_{\min}^{s_{bk}\,\prime} \quad \forall s_{bk} \tag{3.38}$$

Constraint (3.39) shows that the total re-balancing movements should not exceed a certain threshold to control operational costs, where $Numbk_{re}$ is the maximum allowable re-balancing effort.

$$\sum_{s_j=1}^{n} Numbk_{re}(s_j, s_{bk}) - \sum_{s_k=1}^{n} Numbk_{re}(s_{bk}, s_k) \leq Numbk_{re} \tag{3.39}$$

3.5 Common Optimization Algorithms for the Three Models

To optimize the dynamic multi-modal transportation (DMT) model, we use particle swarm optimization (PSO) and genetic algorithms (GA). For the Game Theory-based Multi-Modal Transportation (GT-MMT) and Decentralized Game Theory (DGT) models, we integrate PSO, GA, and Stackelberg. In this section, we outline the steps of each optimization technique for these models, while detailed explanations of the algorithmic processes and mathematical equations will be presented in chapter 4 for DMT, Chapter 5 for GT-MMT, and Chapter 6 for DGT.

3.5.1 Optimization Terms in the Three Models

In the GT-MMT, DGT, and DMT models, we employ several optimization terms to describe the various processes, criteria, and evaluations involved in achieving optimal transportation strategies.

1. Near-optimal solution

A near-optimal solution refers to a solution that is close to the best possible (optimal) solution, but not necessarily the absolute best. In optimization problems, it is often difficult or computationally expensive to find the exact optimal solution, especially in complex or large-scale problems. A near-optimal solution is considered acceptable when it achieves a result that is sufficiently close to the optimal value within a specified margin or tolerance level, providing a balance between solution quality and computational efficiency.

For example, in the GT-MMT, DGT, and DMT models, optimization techniques such as Particle Swarm Optimization (PSO) or Genetic Algorithms (GA) may produce near-optimal solutions that are practical and efficient for real-time applications.

2. Objective Function and Fitness Function

Objective Function refers to the mathematical expression that defines the goal of the optimization process. In all three models (GT-MMT, DGT, and DMT), the objective function is used to minimize trip time and fare for passengers.

Fitness Function evaluates how well a solution (or particle) performs based on the objective function. In the context of the GT-MMT and DGT models, the fitness function incorporates various utility functions for both passengers and transportation modes. The DMT model uses the fitness function to evaluate passenger satisfaction and transportation stress.

3. Convergence and Convergence Criteria

Convergence refers to the process by which the optimization algorithms (such as Particle Swarm Optimization or Genetic Algorithms) approach a near-optimal solution. The point at which the solution stabilizes, indicating that further iterations do not produce significant improvements, is known as convergence.

Convergence Criteria define the stopping conditions for the optimization algorithms. These criteria include limits on the number of iterations or a threshold for changes in the fitness function. For each of the three models, the convergence criteria are tailored to their unique optimization problems.

4. Local and Global Optimization

Local Optimization refers to the optimization that occurs within individual communities in the decentralized structure of the DGT model. In this model, each community optimizes its transportation modes based on local demand and available resources. Local optimization is a core aspect of the DGT model, allowing each community to adjust its strategies without a centralized controller. This ensures that local transportation systems are optimized based on real-time, community-specific data, which improves flexibility and scalability. The local optimization process is coordinated between different communities to achieve global optimization through collaboration.

Global Optimality refers to the solution that minimizes or maximizes the objective function across all possible solutions. It is the ideal outcome that the optimization algorithms aim to achieve.

3.5.2 Genetic Algorithm Integration

Genetic algorithms (GA) are used to explore a vast search space and find global optima. They use a series of generations to evolve solutions to optimization problems, drawing inspiration from the process of natural selection. Integration of objectives and constraints for the three models within a genetic algorithm framework involves GA key components, as we see in table 3.1 and the primary GA process, as we can see in table 3.2. In the table, we write some terms used when evaluating the algorithm of the three models. The process of the GA can be formulated as:

$$PO^{(Gt+1)} = \text{Mutation}(\text{Crossover}(\text{Selection}(PO^{(Gt)}))) \quad (3.40)$$

where $PO^{(Gt)}$ is the population at generation Gt.

Table 3.1: Key components of GA

Component of GA	**Definition**
Population (PO)	A set of candidate solutions
Chromosomes (CH)	Each candidate solution in the population
Genes (Ge)	Elements of a chromosome representing decision variables
Fitness Function (F_{ga})	Evaluates the quality of each solution in GA

Table 3.2: Process of GA

Step of GA Process	**Explanation**
Initialization	Generating initial populations with varied strategic configurations for bicycle availability and redistribution
Fitness Evaluation	Assessing the effectiveness of each strategy based on the objective function and adherence to constraints
Selection	Select the near-optimal solutions based on their fitness values
Crossover	Combine pairs of solutions to produce new offspring
Mutation	Introduce random changes to the offspring to maintain diversity
Evaluation	Evaluate the fitness of the new solutions
Optimization	Iteratively refining strategies until satisfactory optimization levels are achieved, marked by convergence or a predefined performance threshold

Table 3.3: Some terms in GA

Component of GA	**Definition**
Population Size (PS_{ga})	Number of chromosomes (potential solutions) in the population.
Number of Generations (Gt)	Total number of times the genetic algorithm iterates over the population. Each iteration or generation produces a new set of chromosomes
Iterations	This term is often used interchangeably with generations

3.5.3 Particle Swarm Optimization (PSO)

Particle swarm Optimization (PSO) excels in fine-tuning solutions in real time, adapting quickly to changing conditions. The social behaviour of fish schooling or flocking in birds serves as inspiration for PSO. The main outlines of the Key components of the PSO and the PSO process are shown in table 3.4 and table 3.5, respectively.

In PSO, particles are candidate solutions in the search space. Each particle represents a set of strategies for the transportation modes (e.g., routes, schedules, pricing).

The position of a particle in the search space corresponds to a specific strategy configuration for the transportation system. Each position vector x_p includes parameters such as route choices, departure times, and pricing strategies for different transportation modes.

The velocity of a particle determines how the particle's position changes in each iteration. It reflects the direction and magnitude of the change in the strategy configuration. The velocity is influenced by the particle's personal best position and the global best position. The personal best position is the best position (strategy configuration) a particle has achieved so far, based on the fitness evaluation. It represents the most effective strategy found by that specific particle up to the current iteration. The Global best position is the best position found by any particle in the entire swarm. It represents the most effective strategy configuration found by the collective swarm up to the current iteration.

Table 3.4: Key Components of PSO

Parameters	**Definition**
Particles (Pa_i)	Each particle represents different strategy configurations for the players in the model
Velocity (v_i)	The rate of change of the particle's position, which determines how the strategies are updated in each iteration
Position ($Xpso_i$)	The current solution of the particle, which are the specific strategy configurations being evaluated
Personal Best ($pbest_i$)	The best strategy each particle has found
Global Best ($gbest$)	The best strategy found by the entire swarm

Table 3.5: PSO Process

Steps of PSO Process	**Explination**
Initialization	Initialize the particles' positions and velocities randomly
Update Velocity	Update each particle's velocity based on its personal best and the global best
Update Position	Move each particle to a new position based on its updated velocity.
Evaluation	Evaluate the fitness of each particle's new position
Iteration	Repeat the update and evaluation steps to find the optimal configuration

3.5.4 GT-MMT and DGT Stackelberg Optimization

Stackelberg Game Theory is applied in both the GT-MMT and DGT models, where passengers act as leaders and transportation modes act as followers. The main key concepts of Stackelberg game theory can be summarized in the table 3.6. This subsection summarized in short the main outlines of the Stacklberg equilibrium, but the details of the Stacklberg operations with the details of the algorithms and operation are in chapters 5 and 6.

Table 3.6: Key components of Stackelberg game theory

Parameters	Definition
Leaders (p_i)	The decision-makers who move first (passengers)
Followers (m)	The decision-makers who move after observing the leaders' decisions (transportation modes)
Leader's Strategy (A_{p_i})	The optimal strategy chosen by the leader
Follower's Strategy (A_m)	The optimal strategy chosen by the follower in response to the leader's strategy
Payoff Function (U)	The utility or payoff for each player (leader and follower)
Stackelberg Equilibrium	The Stackelberg equilibrium is reached when the leader has chosen the optimal strategy, taking into account the followers' best responses

In the Stackelberg game theory model, the interaction between the leader and the followers is characterized by a hierarchical decision-making process. The leader makes a decision first, and the followers respond to this decision by optimizing their own strategies. This process can be mathematically represented using the following equations:

$$\max_{A_L} U_L(A_L, A_F^*(A_L)) \tag{3.41}$$

where A_L is the strategy the leader chooses, and U_L is the leader's payoff function, which depends on both the leader's strategy (A_L) and the followers' ideal response ($A_F * (A_L)$).

In equation 3.41, the leader anticipates the followers' optimal response $A_F^*(A_L)$ to any strategy A_L they might choose. The leader then selects the strategy A_L that maximizes their own payoff U_L, taking into account the followers' best response.

The followers determine their optimal strategy $A_F^*(A_L)$ in response to the leader's strategy A_L by maximizing their own payoff U_F:

$$A_F^*(A_L) = \arg\max_{A_F} U_F(A_L, A_F) \tag{3.42}$$

where A_F is the strategy chosen by the followers and U_F is the payoff function of the followers, which depends on both the leader's strategy A_L and the followers' own strategy A_F. $\arg\max_{A_F} U_F(A_L, A_F)$ represents the strategy A_F that maximizes the followers' payoff U_F given the leader's strategy A_L. In equation 3.42, the followers observe the leader's strategy A_L. Then, each follower chooses their strategy A_F to maximize their own payoff U_F, considering the leader's choice A_L.

This hierarchical decision-making process captures the essence of the Stackelberg game, where the leader has the advantage of making the first move, and the followers react optimally to this move. This framework is widely used in economics, business strategy, and multi-modal transportation systems to model competitive and cooperative interactions between different agents.

For example, if the passenger (leader) shows a preference for quicker, more cost-effective routes, ride-sharing services may adjust their pricing or routing to match these expectations. Similarly, if a significant number of passengers begin to prefer bicycles for short distances, the bicycle-sharing system might need to increase the number of bikes in high-demand areas to maintain service quality. The bus system might not be as flexible in routing as ride-sharing but can compete by offering lower fares and increasing bus frequency based on the predictive analytics provided by the GT-MMT model. The bicycle-sharing system's integration into the broader GT-MMT model involves coordination with other transportation modes under the Stackelberg leadership framework. This coordination is typically achieved through a genetic algorithm that optimizes the routes and schedules of buses and ride-sharing vehicles and the distribution and availability of bicycles. The genetic algorithm iteratively adjusts the transportation strategies, including bicycle re-balancing plans, to improve overall system efficiency and user satisfaction.

Using a Stackelberg-genetic approach, the model can change over time using a PSO-genetic algorithm that improves based on constant feedback from system performance. This lets the model adapt to new situations and make better decisions for future operations. This leads to an iteratively optimized system where both leaders and followers adjust their strategies to achieve a balanced, efficient transportation network. Continuous data flow ensures the system adapts in real time, adjusting strategies based on current conditions and feedback.

3.6 Simulating Scenarios for DMT, GT-MMT, and DGT Models

In the practical part, the route journey in the three transportation models (DMT, GT-MMT, and DGT) is built using simulation of urban mobility (SUMO) and OMNET++ programs. In addition, we use OpenStreetMap (OSM) to extract a part of a map of the City of Ottawa as a case study. The three models leverage advanced simulation and mapping tools to create a realistic urban mobility scenario. In the following, we will highlight the main stimulation tools we use in the thesis. The details of the simulation setting, analysis, and results for the DMT in section 4.4, the GT-MMT model in section 5.5, and the DGT in section 6.5.

3.6.1 Simulation Tools

The main simulation tools in our work are:

1. OpenStreetMap

A collaborative project to create a free, editable map of the world. It is used to generate realistic network topologies in simulations by providing accurate geographic data. The OpenStreetMap (OSM) provides detailed and up-to-date geographic information and offers accurate road networks, public transit routes, and geographical features essential for the urban area.

2. SUMO (Simulation of Urban Mobility)

An open-source traffic simulation package designed to handle large road networks. SUMO can simulate vehicular movement, public transportation, and pedestrian traffic, offering extensive tools for traffic modelling and analysis.

3. OMNET++

The other tool is OMNET++, which serves as a network simulation framework to model the communication network within the three models. We use OMNET++ to build the proposed model in this thesis. It allows the modelling of complex interactions within a network and can be extended with various frameworks to simulate specific scenarios. In addition, it facilitates the simulation of data exchange between regional managers, vehicles, and passengers, promoting efficient information flow for real-time decision-making.

4. INET

: The *Inet* framework is used to build a Wi-Fi interface. It is a framework for OMNET++ that provides models for internet protocols and communication networks. It includes modules for simulating different types of network devices and protocols, enabling detailed network simulations.

5. Veins

An open-source vehicular network simulation framework based on OMNET++ and SUMO. We connect Veins with SUMO to create a realistic mobility scenario. Veins facilitate a simulation of vehicle-to-vehicle and vehicle-to-infrastructure communication within a traffic simulation environment.

6. Python

It is a programming language used for scripting and automating simulation tasks, data analysis, and integration. Python's extensive libraries and frameworks make it ideal for managing simulation workflows and analyzing simulation results.

3.6.2 Steps to Stimulate the Three Models

To simulate real-time scenarios for the three models (DMT, GT-MMT, and DGT) using OMNET++, INET, Veins, OpenStreetMap, SUMO, and Python, follow these detailed steps:

1. Generating Synthetic Data

In the data collection stage, we simulate the data collection process by generating synthetic data that mimics real-world conditions. This involves creating structured data that represents various aspects of the transportation system, such as vehicle locations, passenger demands, traffic conditions, and more. Here's how we simulate this stage:

- **Define the Parameters**: Determine the parameters that you need to collect data for, such as the number of vehicles, number of passengers, geographical area, time intervals, and simulation time.
- **Generate Data for Each Parameter**: Create functions to generate synthetic data for vehicles, passengers, and other relevant entities.
- **Combine Data into a Single Structure**: Merge the generated data into a single, structured dataset that can be easily processed and analyzed.
- **Save and Manage the Data**: Store the dataset in CSV file format.

2. Setting Up the Simulation Environment

Set up the simulation environment using OMNET++, INET, Veins, and SUMO. OpenStreetMap can be used to create a realistic map.

1. **OpenStreetMap:**
    - Extract the map of the area you want to simulate.
    - Convert the map data into a format that SUMO can use.
2. **SUMO:**
    - Import the map data into SUMO.
    - Define the routes, traffic lights, and vehicle types.
    - Use Python to generate traffic flows and vehicle trips.
3. **Veins:**
    - Integrate SUMO with OMNET++ using Veins.
    - Define communication between vehicles and infrastructure (V2I and V2V communication).
4. **OMNET++ and INET:**

- Set up the network simulation in OMNET++ using INET.
- Define the communication protocols and simulation parameters.

3. Real-Time Information and Updates

Use Python scripts to simulate real-time updates and information dissemination to passengers. Create a Python script to periodically update traffic conditions, vehicle locations, and passenger demands and implement the algorithms used in the models to satisfy the objectives of the thesis.

4. Running the Simulation

Run the integrated simulation to observe how the three models perform in real-time scenarios.

1. **Initialization:**
   - Initialize the simulation environment with the generated synthetic data.
2. **Execution:**
   - Execute the simulation, allowing OMNET++, INET, Veins, and SUMO to interact in real time.
   - Use Python scripts to monitor and adjust the simulation parameters dynamically.

First, to simulate the mobile application in OMNeT++, define the application logic using C++ classes derived from the `cSimpleModule` class. Next, utilize INET modules, `TcpApp` and `UdpApp`, to handle network communications within the simulation. Implement message passing using subclasses of the `cMessage` class for data exchange between different application parts. Finally, simulate user interactions through scripts, as OMNeT++ does not provide support for actual user interface development.

5. Evaluation and Analysis

Evaluate the performance of the three models based on various metrics such as travel time, cost, and passenger satisfaction. Here, we just summarized the main parameters for the three models in table 3.7. The details of the simulation setting, analysis, and results for the DMT in section 4.4, the GT-MMT model in section 5.5, and the DGT in section 6.5.

By carefully setting these parameters, we can tailor the simulation to reflect realistic scenarios and effectively evaluate the performance and outcomes of the DMT, GT-MMT, and DGT models. This enables us to assess how well the model meets its objectives, such as maximizing rider satisfaction, ensuring efficient real-time data exchange, and providing efficient travel options.

Table 3.7: Simulation parameters

SUMO Parameters	
Network File	Defines roads, lanes, and junctions.
Road Network	NETCONVERT
Demand File	Details about trips and vehicle flows.
Vehicle Types	Specifications for different vehicle types
Traffic Lights	Definitions and timings for traffic signals.
Simulation Time	Duration for the traffic simulation(6 a.m-11:59 p.m)
Length of Time Step	1 minute
Dimension of Simulation Area	20km x 20km
The Integration Method	Euler Update
INET Framework Parameters	
Network Topology	Structure of nodes and links.
Node Configuration	Details for each network node.
Routing Protocol	Protocol for routing data packets
Application Layer	Configuration of data generation.
Veins Parameters	
Mobility Models	Defines node movement patterns.
Communication Parameters	Settings for wireless communication
Scenario Configuration	Describes the integrated simulation scenario
OpenStreetMap (OSM) Parameters	
Map Data	Geographic information from OSM.
Extraction Tools	Tools used to convert OSM data for SUMO.

OMNeT++ General Parameters	
Network Canvas	Visual representation of the network.
Simulation Parameters	General settings for the simulation.
Statistic Collection	Configuration for data recording.
Simulation Period	6 a.m-11:59 p.m
Network Parameters	
Station Locations	Coordinates of all transportation points
Road Network	Graph representing the city's road layout
Traffic Data	Real-time coordinates of each vehicle

Chapter 4 A Dynamic Mobility Traffic (DMT) Model

4.1 Model Overview

In response to the growing challenges in urban transportation systems, this chapter introduces a novel dynamic mobility traffic (DMT) scheme strategically designed to integrate public buses with car ride-sharing services. The primary goal of this innovative approach is to significantly enhance transportation efficiency and passenger satisfaction through the optimized use of mixed-modal resources.

The DMT scheme's inception is predicated on the principle that combining various modes of transportation can result in a more flexible, responsive, and efficient urban transit system. By leveraging real-time data exchange, the scheme facilitates dynamic decision-making processes that consider the real-time operational status and demands across different transportation modes.

Real-time data exchange serves as the backbone of the DMT scheme, enabling an unprecedented level of coordination between the regional manager of the transportation modes, the operators of public buses, and providers of car ride-sharing services. This integration allows for adjustments to be made in real time, enhancing the ability to respond swiftly to fluctuating demands and operational challenges.

Figure 4.1 shows the main components of the DMT model, consisting of car ride-sharing, public buses, and passengers. For a comprehensive understanding of the full topology associated with the DMT model, refer to subsection 3.3 in chapter 3.

In this model, costs differ depending on the distance travelled by the passengers to reach their destination. We assumed several relevant hypotheses that must be mentioned. First, we did not consider the person's walking time between the stations on his trip. Second, while the passenger's origin is the first station of the journey, the last station is the destination. Finally, security and trust have been assumed in this model.

The remainder of this chapter is organized as follows: section 4.2 (Objectives function and

Problem Formulation); section 4.3 (DMT Optimization Algorithm); section 4.4 (Simulation and Results).

4.2 Objectives function and Problem Formulation

The main objectives of the dynamic mobility traffic (DMT) model are to minimize each passenger's total travel time and fare while optimizing the overall system efficiency:

$$\min\left(\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}\right) \tag{4.1}$$

This model aims to provide passengers with the best combination of transportation modes to achieve the lowest travel time and cost for their trips. For a comprehensive understanding of the full notations, objectives, and constraints associated with the DMT model, please refer to subsections 3.2.1, 3.4.1, and 3.4.2. These subsections provide detailed explanations of the parameters, mathematical formulations, and constraints that guide the optimization processes within the DMT framework.

The scheduling process for cars and public buses defines their routes and destination times. We assume that the passenger (p) periodically receives information describing the current car

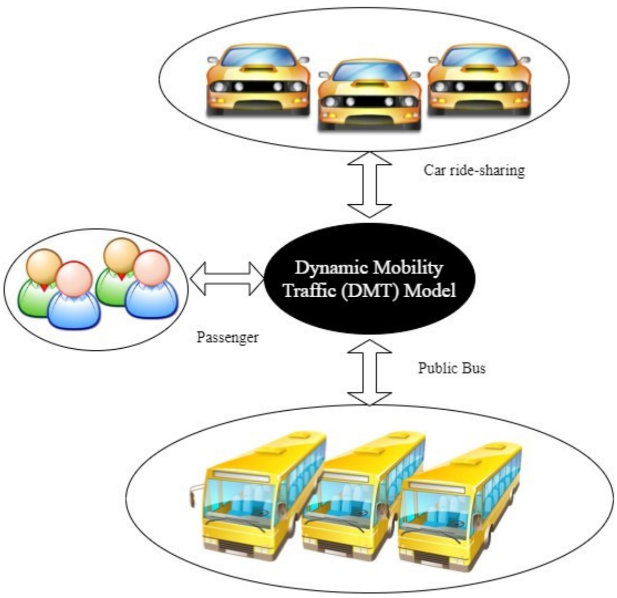

Figure 4.1: DMT model overview

ride-sharing position and/or the bus position and their seating availability. This information, which is collected based on a passenger's preference (pr_{p_i}) for a bus, car, or both, is used to calculate the total price (TF_{p_i}) and the total travel time (TT_{p_i}).

We assume that the DMT manager receives updated information from all participants every 5 minutes. As we can see, Figure 4.2 shows the main interaction between the players in the model. Each element has a set of parameters that contribute to implementing the proposed model's algorithm. In this figure, the regional manager uses the DMT to manage the interconnection between the car ride-sharing, the public bus, and the passenger.

In general, the operation steps can be concluded as follows: Start with data collection by gathering real-time data using OpenStreetMap and OMNET++ to simulate a realistic urban environment. Then, establish a network for data exchange between regional managers, public buses, car ride-sharing services, and passengers. The system needs to continuously update schedules and routes based on real-time traffic data and passenger demands and inform passengers at each station about the capacity and current position of the upcoming buses and cars. Also, adjusting routes for the ride-sharing dynamically to address changing traffic conditions and passenger needs is very important to the DMT model.

We assume that all public buses and car ride-sharing are connected to a mobile application

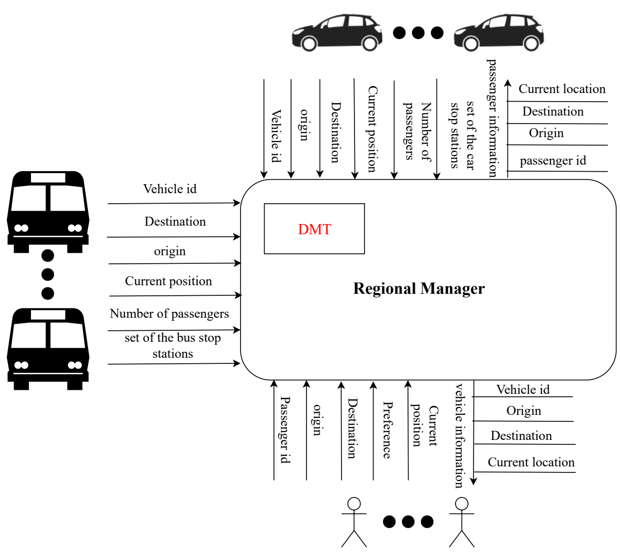

Figure 4.2: The data exchanged between the regional manager, the public buses, the ride-sharing cars and the riders

,

to track all participants in real time. We propose the DMT algorithm to manage the demand for the transport service. Upon receiving periodical information about the availability of public buses and cars (available seats, time to reach the stations), the DMT calculates the time required by each available vehicle (having empty seats) to arrive at the final destination from the current passenger's position. It schedules the results in descending order. The result is compared to the passenger's maximum arrival time, and the first public bus with an equal or lower value will be selected. In order for the passenger to determine whether to continue to travel on the same public bus or transfer to another bus, the time required to reach the destination on the recent bus is compared to the result of the previous process. The passenger chooses a public bus in the shortest possible time. The same process occurs if car ride-sharing is selected as the mode of transportation for the whole trip. On the other hand, the passenger can choose to use a combination of the public bus and car ride-sharing on his trip. The model gives him the optimal combination with the fastest arrival time and the least cost.

The Dynamic Mobility Traffic (DMT) model integrates various components to optimize urban transportation. The transportation modes, denoted as m, include car ride-sharing and the public bus, providing a range of options for city travellers. This is represented by the notation $m = \{c_j, b_k\}$

Passenger data, represented as P_{data}, includes a collection of passenger information, with each passenger p having specified origin $S_1^{p_i}$, destination $D_{np}^{p_i}$, and preferred mode of transportation m. This data structure is crucial for personalizing travel solutions and is expressed as $P_{data} = \{R \mid i = 1, 2, \ldots, Numbp\}$ and $p = \{S_1^{p_i}, D_{np}^{p_i}, np\}$.

Furthermore, vehicle data, denoted as m_{data}, plays a critical role in the model. This includes details of each vehicle m in the system, such as capacity Ca_m, current location and estimated arrival time $LA_{v_{v_{m+k}}}$. The notation for vehicle data is given by $V_{data} = \{V_m \mid j = 1, 2, \ldots, m\}$ and $V_m = \{C^{v_m}, LA_{v_{v_{m+k}}}\}$.

The core of the DMT model lies in formulating an optimization problem that aims to minimize the total travel time and cost for all passengers while maximizing vehicle utilization. Figure 4.3 shows the interactions between a user and the DMT model.

4.3 DMT Optimization Algorithm

Combining genetic algorithms with particle swarm optimization provides a robust and adaptable solution for dynamically optimizing the DMT model, ensuring efficient and responsive transportation routes. This method combines genetic algorithms with particle swarm optimization to make a strong and flexible way to improve the DMT model, making sure that transportation routes are quick and responsive. The GA phase provides a diverse set of potential solutions, while the PSO phase refines these solutions to adapt to real-time changes and new requests, resulting in optimized routes and schedules that minimize total trip time and fare for passengers. The main step of the process is seen in algorithm 1 and followed by the details of the process step.

Step 1: Data Collection (Initialization Phase)

The objective of this phase is to collect real-time and static data needed to initialize and continually update the transportation model.

Real-time traffic data, denoted as RT, includes traffic speeds, congestion levels, and accidents. This data is represented as a matrix where each element RT_{ed_x} denotes the traffic information on the edge ed_x. Vehicle data, denoted as M, is a set of vehicles $\{m_1, m_2, \ldots, m_n\}$. For each vehicle $m \in M$, the location L_m can be represented as (x_m, y_m). Passenger data, denoted as P, is a set of passengers $\{p_1, p_2, \ldots, p_i\}$. For each passenger $p_i \in R$, the origin $S_1^{p_i}$ can be represented as $(x(S_1^{p_i}), y(S_1^{p_i}))$, the destination $D_{np}^{p_i}$ can be represented as $(x(D_{np}^{p_i}), y(D_{np}^{p_i}))$, the latest and earliest departure time, and the maximum fare MF_{p_i} that the passenger is willing to pay. See the subsection 3.2.1 in chapter 3 to read the full data with all variables and notations.

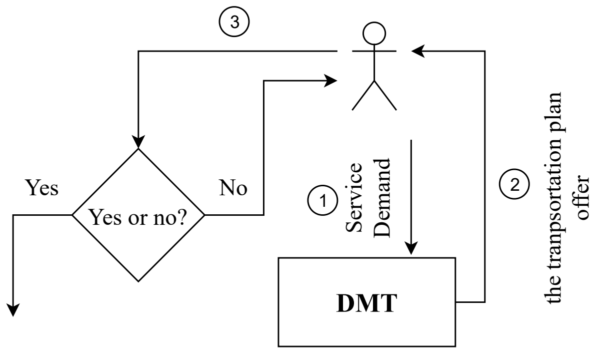

Figure 4.3: The interactions between a user and the DMT model

Algorithm 1 DMT Optimization Model Algorithm

Input: Number of particles/individuals, number of iterations I, initial population PO_o, weight factors α, β

Output: Optimal combination of modes for each passenger with the lowest fare and shortest trip time

Initialization: Initialize the population of particles (for PSO) and individuals (for GA) with random positions and velocities. Define the objective function for each passenger p_i:

$$\min\left(\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}\right) \tag{4.2}$$

for $t = 1$ ***to*** I **do**

- **Step 1: Evaluate Fitness**
  for *each particle/individual* **do**
  - Calculate fitness $Fitness(p_i)$ based on the objective function in Equation 4.9
- **Step 2: Genetic Algorithm Operations**
  Select individuals based on fitness values.
  Perform crossover to generate offspring.
  Apply mutation to offspring.
  Replace less fit individuals with new offspring.
- **Step 3: PSO Operations**
  for *each particle* **do**
  - Update velocity using the equation:

    $$v_{pa_i}^{t+1} = \omega v_{pa_i}^{t} + c_1 r_1 (pbest_{pa_i} - Xpso_{pa_i}^{t}) + c_2 r_2 ((gbest_{pa} - Xpso_{pa_i}^{t}) \tag{4.3}$$

    Update position using the equation:

    $$Xpso_{pa_i}^{t+1} = Xpso_{pa_i}^{t} + v_{pa_i}^{t+1} \tag{4.4}$$

    Update personal best $pbest_{pa_i}$
- Update global best $gbest_{pa}$
- **Step 4: Scheduling and Selection**
  for *each route* **do**
  - Calculate the time required by each available vehicle to arrive at the final destination.
    Schedule results in descending order.
    Compare results to the passenger's maximum arrival time.
    Select the first public bus or ride-sharing option with equal or lower value.
- **Step 5: Route Optimization**
  Determine if the passenger should transfer or continue in the same vehicle.
  Select the near-optimal combination of public bus and ride-sharing.

return Optimal combination of modes for each passenger

The real-time traffic data RT can be mathematically represented as:

$$T = \begin{bmatrix} RT_{11} & RT_{12} & \cdots & RT_{1n} \\ RT_{21} & RT_{22} & \cdots & RT_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ RT_{n1} & RT_{n2} & \cdots & RT_{nn} \end{bmatrix}$$

Step 2:Initialization:

- Initialize the population of chromosomes and particles randomly.

  Each chromosome represents a potential solution. Specifically, a chromosome represents the combination of transportation modes and routes that a passenger can take from their source to their destination, aiming to minimize travel time and fare. Let CH_i denote the i-th chromosome in the population. Each chromosome CH_i is composed of routes $RO_{i,m}$ for each vehicle m in the system. Let M be the set of vehicles, where $M = \{m_1, m_2, \ldots, m_n\}$. Let R be the set of passengers, where $R = \{p_1, p_2, \ldots, p_i\}$.

  Initial solutions generated by the genetic algorithm to generate initial routes and vehicle combinations that are likely to be more efficient. The population size is denoted by PS_{ga}, which is the number of chromosomes in the population. A larger PS_{ga} increases the diversity of potential solutions but also increases computational complexity.

  A chromosome CH_i can be represented as $CH_i = \{Ro_{i,1}, Ro_{i,2}, \ldots, Ro_{i,m}\}$, where $RO_{i,m}$ represents the route for vehicle m in chromosome CH_i. Each route $RO_{i,m}$ is a sequence of stops that vehicle m will visit: $RO_{i,m} = \{S_{i,j,1}, S_{i,j,2}, \ldots, S_{i,j,k}\}$, where $S_{i,j,k}$ denotes the k-th stop in the route of vehicle v_j in chromosome C_i. Each stop $S_{i,j,k}$ is associated with a specific passenger pickup or drop-off point. For each passenger $p \in P$, the origin O_p and destination D_p are included in the route.

  The constraints include the vehicle capacity constraint and the time window constraint. To see the full constraint, read the subsection 3.4.2. The vehicle capacity constraint is represented as:

$$l_{s_n}^m \leqslant Nump_{s_n}^m \leqslant Ca_m, \forall m \in M, S_n \in A \tag{4.5}$$

The time window constraint is represented as:

$$F_{S_1^{p_i}} \leqslant T_{I_r^{p_i}}^m \leqslant E_{S_1^{p_i}}, \forall m \in M, S_1^{p_i}, I_r^{p_i} \in A, p_i \in R \tag{4.6}$$

The initial population for GA is denoted by:

$$\text{Population}_{\text{GA}} = \{CH_1, CH_2, \ldots, CH_{N_{PS_{ga}}}\}$$

- Initialize the population of particles.

  Each particle in the swarm represents a potential solution, encoding routes and schedules for all vehicles. Let NP_{pso} denote the number of particles in the swarm. The position of particle i, represented as $\mathbf{x}_i$, indicates a potential solution. The velocity of particle i, denoted by $\mathbf{v}_i$, determines how the position of the particle will change over time. The personal best position of particle i is represented as $\mathbf{p}_i$, and the global best position found by the swarm is denoted by $\mathbf{g}$. The inertia weight w controls the influence of the previous velocity on the current velocity. The cognitive (personal) learning factor is represented by c_1, and the social (global) learning factor is denoted by c_2. Additionally, r_1 and r_2 are random numbers uniformly distributed in the interval $[0, 1]$.

  Particle Representation: Each particle in the swarm, denoted as Pa_i, represents a potential solution for the routing problem. The position of particle i, $Xpso_i$, consists of routes for all vehicles: $Xpso_i = \{RO_{i,1}, RO_{i,2}, \ldots, RO_{i,j}\}$. Here, $RO_{i,j}$ represents the route for vehicle m in particle Pa_i.

  Route Representation: Each route $RO_{i,j}$ consists of a sequence of stops that vehicle v_j will visit: $RO_{i,j} = \{S_{i,j,1}, S_{i,j,2}, \ldots, S_{i,j,k}\}$, where $S_{i,j,k}$ denotes the k-th stop in the route of vehicle m_j in particle Pa_i.

  Initialization Process: The initialization of the swarm involves generating initial solutions either randomly or using heuristic methods. Each initial solution ensures that all passengers are assigned to a vehicle without violating capacity constraints. The initial swarm for PSO is represented as: $\text{Swarm}_{\text{PSO}} = \{Pa_1, Pa_2, \ldots, Pa_{N_{\text{PSO}}}\}$

Step 3:Fitness Evaluation:

By integrating the fitness evaluation process with GA and PSO, the algorithm leverages the strengths of both methods to find near-optimal solutions for minimizing total trip time and fare for passengers while satisfying constraints. The main steps of the fitness evaluation can be concluded as follows:

- Evaluate the fitness of each chromosome and particle.
- Update personal best positions for particles.
- Update the global best position based on the fitness values of particles.
- Evaluate the fitness of each chromosome and particle using the objective function.

  The goal of the fitness evaluation process is to assess how well each chromosome (in GA) and each particle (in PSO) meets the optimization objectives, specifically minimizing the total trip time and fare for passengers while satisfying constraints.

For Genetic Algorithm (GA):

1. **Evaluate Fitness of Each Chromosome:**

$$F(CH) = \frac{1}{\alpha \sum_{p\in R} TT_{p_i}(CH) + \beta \sum_{p\in R} TF_{p_i}(CH) + \text{Penalties}(CH)} \tag{4.7}$$

Here, $TT_{p_i}(CH)$ is the travel time for passenger p_i in chromosome CH, $TF_{p_i}(CH)$ is the fare for passenger p_i in chromosome CH, and Penalties(CH) accounts for any violations of constraints such as vehicle capacity and time windows.

The objective function DMT optimization problem is designed to minimize both the trip time and the fare for passengers. The objective function is given by the equation:

$$Fitness(p_i) = (\alpha_{p_i} TT_{p_i} + \beta_{p_i} TF_{p_i}) \tag{4.8}$$

where α_{p_i} and β_{p_i} are the weighting factors for travel time and cost for passenger p_i, respectively. For the full explanation of the objectives for the DMT, read 3.4.1 in chapter

3. The optimization is subject to several constraints. All constraints with the explanation details in subsection 3.4.2 in chapter 3.

For Particle Swarm Optimization (PSO):

1. **Evaluate Fitness of Each Particle:**

In the context of dynamic mobility traffic (DMT) optimization, the fitness of each particle is evaluated based on an objective function that seeks to minimize both the total trip time and the fare for passengers. This fitness evaluation process is crucial for determining how well a particular routing and scheduling solution (represented by a particle) meets the optimization goals. The objective function of the DMT problem is designed to balance the trade-offs between travel time and cost. It is defined as follows:

$$\min\left(\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}\right) \tag{4.9}$$

where t_p is the travel time for passenger p, f_p is the fare for passenger p, and α and β are weights that reflect the relative importance of minimizing time and cost, respectively. These weights can be adjusted based on specific goals or preferences. The components of the objective function include travel time (t_p), which represents the total time taken for a passenger p to travel from their origin to their destination, including waiting time, in-transit time, and any transfer times if multiple modes of transport are used. The fare (f_p) represents the cost incurred by passenger p for their trip, including base fares, distance-based charges, and any additional costs due to dynamic pricing mechanisms. The weights (α and β) are used to adjust the relative importance of minimizing travel time versus minimizing fare. For instance, if the primary goal is to reduce travel time, α would be set higher relative to β.

The fitness function is used to evaluate how well a solution (particle) performs according to the objective function. It is defined as:

$$\text{Fitness}(X) = \frac{1}{\alpha \sum_{p\in P} t_p(X) + \beta \sum_{p\in P} f_p(X) + \text{Penalties}(X)} \tag{4.10}$$

where X represents a particular solution (chromosome or particle) in the search space,

$t_p(X)$ is the travel time for passenger p in solution X, $f_p(X)$ is the fare for passenger p in solution X, and Penalties(X) includes penalties for any violations of constraints such as vehicle overcapacity or time window violations. The fitness function, based on the objective function, takes into account the travel time component ($t_p(X)$), which sums the travel times of all passengers for a given solution X, ensuring that the overall travel time for all passengers is as low as possible. It also considers the fare component ($f_p(X)$), which sums the fares paid by all passengers for a given solution X, ensuring that the total cost to passengers is kept low. The penalty term (Penalties(X)) adds penalties for any constraint violations in the solution X, ensuring that solutions violating these constraints are less favourable. The inverse of the sum is taken, meaning that lower values of the objective function (indicating better solutions) correspond to higher fitness values; thus, the goal is to maximize fitness, which aligns with minimizing the objective function.

2. **Update personal and global best positions based on fitness values.**

Update the personal best position for each particle based on its fitness. If the fitness of any particle is better than the current global best, update the global best position.

Each particle in the Particle Swarm Optimization (PSO) algorithm maintains a memory of its best position encountered so far, referred to as the personal best ($\mathbf{p}_i$). The algorithm also keeps track of the best position encountered by any particle in the swarm, known as the global best ($\mathbf{g}$).

a) Update Personal Best

The personal best position of a particle is updated whenever the current position of the particle has a better fitness value than the fitness value of the personal best position. This ensures that the particle retains the memory of the near-optimal solution it has found so far. Mathematically, this update process can be represented as:

$$\text{If Fitness}(Xpso_i(t+1)) > \text{Fitness}(pbest_i) \text{ then } pbest_i = Xpso_i(t+1) \tag{4.11}$$

Here, $\mathbf{x}_i(t+1)$ represents the new position of particle i at time $t+1$, and $pbest_i$ is the personal best position of particle i. The fitness function, based on the objective function, evaluates how well a given solution meets the optimization objectives, minimizing total trip time and fare.

b) Update Global Best

The global best position is updated whenever any particle's personal best position has a better fitness value than the fitness value of the current global best position. This ensures that the swarm collectively remembers the near-optimal solution found by any particle. Mathematically, this update process can be represented as:

$$\text{If Fitness}(pbest_i) > \text{Fitness}(gbest) \text{ then } gbest = pbest_i \tag{4.12}$$

Here, $gbest$ is the global best position, and $pbest_i$ is the personal best position of particle i. The fitness function, based on the objective function, evaluates the fitness of the personal best position of particle i and compares it to the fitness of the current global best position.

Step4: GA Operations:

Perform selection, crossover, and mutation on the chromosome population. Then, ensure the feasibility of the new offspring by repairing any constraint violations.

1. Selection Process

The goal of the selection process in the genetic algorithm is to choose parent chromosomes from the population based on their fitness scores. The selected parents will then undergo crossover and mutation operations to produce the next generation of solutions.

In Tournament Selection, a subset of chromosomes is chosen randomly, and the best chromosome (with the highest fitness score) is selected as a parent. Let $PO = \{CH_1, CH_2, \ldots, CH_N\}$ be the set of chromosomes, and let $F_{ga}(CH_i)$ be the fitness of chromosome CH_i. Let k be the tournament size, and let SelectedParents be the set of chosen parents for crossover.

For each parent selection, a subset $Subset_g a \subset \text{PO}$ of size k is randomly chosen. The chromosome CH_{best} with the highest fitness in the subset is identified using the equation:

$$CH_{\text{best}} = \arg \max_{CH \in Subset_{ga}} F_{ga}(CH) \tag{4.13}$$

The chromosome C_{best} is then added to SelectedParents. This process is repeated until the desired number of parents is selected.

Suppose the population consists of 5 chromosomes with fitness values: $F(CH_1) = 10$, $F(CH_2) = 20$, $F(CH_3) = 15$, $F(CH_4) = 5$, and $F(CH_5) = 25$. If the tournament size $k = 3$, a subset $\{CH_2, CH_4, CH_5\}$ is randomly chosen. The best chromosome in the subset is CH_5 with a fitness of 25, and CH_5 is added to SelectedParents.

2. crossover Process

The crossover operation aims to combine the genetic material of two parent chromosomes to produce offspring that inherit characteristics from both parents. The process begins by selecting a crossover point c randomly. This point determines where the routes from the parent chromosomes will be split. For instance, if the route length L is 10 and the crossover point c is 4, the first four stops will come from one parent and the remaining stops will come from the other parent.

Let P_1 and P_2 be two parent chromosomes, and let O_1 and O_2 be the offspring chromosomes. The routes for vehicle v_j in parent chromosomes P_1 and P_2 are denoted as $R_{P_1,j}$ and $R_{P_2,j}$, respectively. Similarly, the routes for vehicle v_j in offspring chromosomes O_1 and O_2 are denoted as $R_{O_1,j}$ and $R_{O_2,j}$, respectively. The crossover point is represented by c, and L denotes the length of the routes in the parent chromosomes.

The crossover operation can be described in the following steps:

The first step is the selection of a crossover point. Here, select a crossover point c randomly along the length of the routes in the parent chromosomes. The crossover point determines where the routes will be split and combined. The crossover point c is chosen such that $1 \leq c \leq L - 1$.

The second step is splitting and combining routes. Here, split the routes $R_{P_1,j}$ and $R_{P_2,j}$ at the crossover point c into two segments. Combine the first segment of $R_{P_1,j}$ with the second segment of $R_{P_2,j}$ to form the route $R_{O_1,j}$ for offspring O_1. Similarly, combine the first segment of $R_{P_2,j}$ with the second segment of $R_{P_1,j}$ to form the route $R_{O_2,j}$ for offspring O_2.

The last step is ensuring feasibility. To ensure feasibility, the combined routes must be checked against vehicle capacity constraints and time window constraints. The vehicle capacity constraint ensures that the number of passengers assigned to a vehicle does not exceed its capacity. Mathematically, this can be represented as:

$$\sum_{p \in P_{i,j,k}} x_{p,v_j} \leq C_{v_j} \quad \forall v_j \in V \tag{4.14}$$

where $P_{i,j,k}$ is the set of passengers associated with stop $S_{i,j,k}$, x_{p,v_j} is 1 if passenger p is assigned to vehicle v_j, otherwise 0, and C_{v_j} is the capacity of vehicle v_j.

The time window constraint ensures that each passenger is picked up and dropped off within their specified time windows. This can be represented as:

$$T_{i,j,k}^{\text{pickup}} \leq T_{i,j,k} \leq T_{i,j,k}^{\text{dropoff}} \quad \forall S_{i,j,k} \in R_{i,j} \tag{4.15}$$

where $T_{i,j,k}$ is the time at which vehicle v_j visits stop $S_{i,j,k}$, and $T_{i,j,k}^{\text{pickup}}$ and $T_{i,j,k}^{\text{dropoff}}$ are the earliest and latest allowable times for pickup and drop-off.

Consider the routes $R_{P_1,j}$ and $R_{P_2,j}$ for a vehicle v_j in the parent chromosomes P_1 and P_2, respectively:

$$R_{P_1,j} = \{S_{P_1,j,1}, S_{P_1,j,2}, \ldots, S_{P_1,j,c}, S_{P_1,j,c+1}, \ldots, S_{P_1,j,L}\}$$
$$R_{P_2,j} = \{S_{P_2,j,1}, S_{P_2,j,2}, \ldots, S_{P_2,j,c}, S_{P_2,j,c+1}, \ldots, S_{P_2,j,L}\}$$

After the crossover operation, the routes $R_{O_1,j}$ and $R_{O_2,j}$ for the offspring chromosomes O_1 and O_2 are:

$$R_{O_1,j} = \{S_{P_1,j,1}, S_{P_1,j,2}, \ldots, S_{P_1,j,c}, S_{P_2,j,c+1}, \ldots, S_{P_2,j,L}\}$$
$$R_{O_2,j} = \{S_{P_2,j,1}, S_{P_2,j,2}, \ldots, S_{P_2,j,c}, S_{P_1,j,c+1}, \ldots, S_{P_1,j,L}\}$$

3. Mutation Process

The mutation operation aims to introduce small, random changes to chromosomes to explore new potential solutions and maintain genetic diversity within the population. Let C be a chromosome, and let $R_{C,j}$ be the route of vehicle j in chromosome C.

The mutation operation involves two main types of mutations: swap mutation and time adjustment mutations.

In the swap mutation, two stops S_{i,j,k_1} and S_{i,j,k_2} are randomly selected within a route $R_{C,j}$. These stops are then swapped, transforming the sequence from $\{\ldots, S_{i,j,k_1}, \ldots,$ $S_{i,j,k_2}, \ldots\}$ to $\{\ldots, S_{i,j,k_2}, \ldots, S_{i,j,k_1}, \ldots\}$.

In the time adjustment mutation, a stop $S_{i,j,k}$ is randomly selected within a route $R_{C,j}$. The departure time $T_{i,j,k}$ at this stop is then adjusted slightly by ΔT. The new time is given by $T_{i,j,k} \to T_{i,j,k} + \Delta T$, ensuring that the new departure time does not violate any constraints, specifically: $T_{i,j,k}^{\text{pickup}} \leq T_{i,j,k} + \Delta T \leq T_{i,j,k}^{\text{dropoff}}$.

Example: Suppose the route $R_{C,j}$ is $\{S_1, S_2, S_3, S_4\}$. In a swap mutation, if stops S_2 and S_4 are swapped, the new route becomes $R_{C,j} = \{S_1, S_4, S_3, S_2\}$.

Constraints: To ensure the feasibility of the mutated routes, two main constraints must be respected. First, no time window violations should occur, which means the adjusted departure times must satisfy $T_{i,j,k}^{\text{pickup}} \leq T_{i,j,k} + \Delta T \leq T_{i,j,k}^{\text{dropoff}}$. Second, vehicle capacity constraints must be maintained, ensuring that the total number of passengers assigned to any vehicle does not exceed its capacity: $\sum_{p \in P_{i,j,k}} x_{p,v_j} \leq C_{v_j}$.

Step 5: PSO Operations:

Update the velocity and position of each particle. In the DMT optimization context, the position $Xpso_i$ of a particle represents a specific routing and scheduling solution for the vehicles. The velocity v_i represents the direction and magnitude of changes to this solution. By iterative updating the velocity and position of each particle, the PSO algorithm explores different routing and scheduling configurations, aiming to minimize the total trip time and fare for passengers. The velocity update equation governs how each particle adjusts its velocity based on three components: inertia, cognitive learning, and social learning. The equation is given by:

$$v_i(t+1) = w \cdot v_i(t) + co_1 \cdot st_1 \cdot (pbest_i - Xpso_i) - x_i(t)) + co_2 \cdot st_2 \cdot (g_{best} - Xpso_i(t)) \quad (4.16)$$

The inertia component, represented by $w \cdot v_i(t)$, involves the inertia weight w, which controls the influence of the previous velocity of the particle. A higher inertia weight encourages global exploration, while a lower inertia weight encourages local exploitation. Here, $v_i(t)$ denotes the current velocity of particle i at time t.

The cognitive (personal) component, given by $co_1 \cdot st_1 \cdot (pbest_i - Xpso_i)$, represents the particle's tendency to return to its personal best position. The term co_1 is the cognitive learning factor, which determines the weight of this component. The term st_1 is a random number uniformly distributed in the range [0, 1], introducing stochastic behaviour. In this context, $pbest_i$ is the personal best position of particle i, and $Xpso_i(t)$ is the current position of particle i at time t.

The social (global) component, expressed as $co_2 \cdot st_2 \cdot (g_{best} - Xpso_i(t))$, represents the particle's tendency to move towards the global best position found by the swarm. The term co_2 is the social learning factor, which determines the weight of this component. The term st_2 is another random number uniformly distributed in the range [0, 1], introducing additional stochastic behaviour. Here, g_{best} is the global best position found by the swarm, and $Xpso_i(t)$ is the current position of particle i at time t.

The velocity update equation combines these three components to adjust the particle's velocity, balancing exploration (searching new areas) and exploitation (refining known good solutions).

After updating the velocity, the particle's position is updated using the following equation:

$$Xpso_i(t+1) = Xpso_i(t) + v_i(t+1) \quad (4.17)$$

$Xpso_i(t+1)$ is the new position of particle i at time $t+1$. $Xpso_i(t)$ is the current position of particle i at time t. $\mathbf{v}_i(t+1)$ is the updated velocity of particle i at time $t+1$. This equation simply adds the updated velocity to the current position, moving the particle to a new position in the search space.

Step 6: Iteration:

Repeat the fitness evaluation, GA operations, and PSO operations until a termination criterion (maximum number of iterations and convergence threshold) is met. The PSO algorithm iterates through the position and velocity updates, fitness evaluations, and best position updates until a termination criterion is met. Through this process, the PSO algorithm effectively balances the exploration of new solutions with the exploitation of known good solutions, optimizing the DMT system's performance.

Maximum Number of Iterations: The algorithm stops after a predefined number of iterations, ensuring that the optimization process does not run indefinitely.

Convergence: The algorithm stops when the improvement in fitness values between iterations falls below a certain threshold, indicating that the particles have converged to a stable solution.

By iterating through these steps, the PSO algorithm explores the search space, converging towards a near-optimal solution that minimizes the total trip time and fares for passengers while respecting constraints.

4.4 Simulation and Results

Figure 3.1 shows the partial map of the City of Ottawa with two forms of transportation (public buses and car ride-sharing). In section 4.4.2, we evaluate our model (DMT), and then we compare the performance of our DMT in section 4.4.3 with a multi-load model presented in [140].

4.4.1 Simulation Setup

Simulation setups are presented in Table 4.1. To obtain part of the Ottawa map, we use Simulation of Urban Mobility (SUMO), and to set the route, we use OMNET++. SUMO randomly sets up the mobility traffic scenario. This simulation measures the movement of car ride-sharing and public buses in the City of Ottawa from 12:00 a.m. until 11:59 p.m. For more detailed information on the simulation settings and parameters used in the DMT model, please refer to the simulation section 3.6 in 3 which provides comprehensive insights into the initialization processes, network configuration, transportation mode parameters, and the integration of real-time data to ensure accurate and realistic simulations.

From 6 am to 11:59 pm, the simulation was run by adjusting some parameters of the public bus and the car on the trip, such as the number of public buses to 30, the maximum capacity of the public bus to 50 passengers, the number of the bus station to 30, the number of cars to 10, and the maximum car ride-sharing capacity to 4 passengers. In addition, we set the number of passengers to 1000 and assigned the route for each passenger from the departure station to the destination. The passenger chooses his/her preferred mode of transportation (a public bus, car ride-sharing, or a combination of a bus and car). In this work, the maximum arrival time and the maximum price are determined randomly by the simulation.

4.4.2 Experiment Evaluations and Results Discussion

1. The average user satisfaction

In the DMT, user satisfaction is the number of passengers who accept the trip plan (source, destination, price, trip time, bus and car ride-sharing scheduling) as suggested by the DMT system after sending the demand divided by the total number of passengers in the simulation.

Table 4.1: Simulation parameters for the DMT model

Vehicles types	Car ride-sharing, Public bus and bicycle
Simulation time	12:00 a.m.-11:59 p.m.
Time set length	1 minute
Number of the ride-sharing	10
Number of public buses	30
Number of passengers	10000
number of the public bus stations	10
number of the ride-sharing stations	8
Maximum capacity of the bus	50 passengers
Maximum capacity of ride-sharing	4 passenger
simulation period	12 a.m-11:59 p.m

It can be seen from Figure 4.4 that the satisfaction rate of passengers using any mode of transportation decreases with an increase in the density of users. The average user satisfaction when using only buses or cars is less than the average user satisfaction when using a combination of car ride-sharing and public buses. The savings rate for the public bus is 36.8%, and for car ride-sharing, when the number of users is equal to 1500, it is 68.42%, as shown in Tables 4.2 and 4.3. This ratio remains almost unchanged in the case of 7000 users but differs when the number of passengers is equal to 10,000. In the case of 10,000 passengers, the savings rate is almost 62.5% for riding the public bus only, and 75% for car ride-sharing only.

Conclusion

As shown in Figure 4.4, it is evident that the user satisfaction level using only car ride-sharing to travel is lower than the user satisfaction level using only a public bus. This result can be illustrated as follows. The percentage of users who express satisfaction when using a bus is higher, especially since buses have more seats than cars. The explanation for this result is that the DMT scheme concept is based on how the users reach their destination at the lowest possible price and in the shortest possible time, regardless of other luxury preferences or congestion factors while riding.

2. The stress level metric

At the same time, the stress level is the number of transportation modes (car and bus) used by the passengers divided by the total number of vehicles in the simulation. Figure 4.5 compares the stress level of passengers when using various modes of transportation from 6 am to 11:59 pm. The figure shows that the pressure level when using only the public bus or only the car ride-sharing is very similar. The stress level increased significantly from 6 am to 8 am. It then decreased and increased during the day until 6 pm. At 6 pm, the stress level was 95%, after which the percentage began to drop to 10%.

If we compare the savings rate (the rate of conservation) when using the car ride-sharing or public bus only with that using the car and bus together, we find that the price savings (the rate of conservation) increase from 6 am until it reaches 45% at 10 am. Next, the conservation rate decreased very significantly to reach around 15% at 11 am. From 11:01 am, the same ratio will

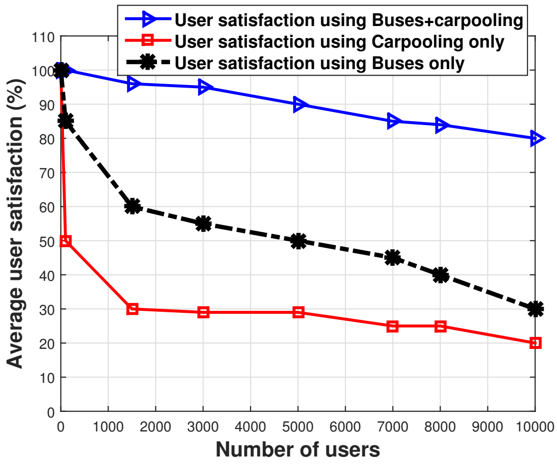

Figure 4.4: The average user satisfaction

Table 4.2: Saving rate of average user satisfaction in the case of using public bus only

Passenger Number	**Public bus and Car ride-sharing (%)**	**Public bus Only(%)**	**Saving Rate (%)**
[1-1500]	95	60	36.8
[1501-7000]	85	45	47.06
[7001-10000]	80	30	62.5

continue to rise to 60% at 4 pm before the rate begins to decline significantly after 6 pm to 0 at 11 pm.

From Table 4.4, we find that the savings rate between 6:00 am and 11:00 am is 24.6% when using the public bus or car ride-sharing. This ratio increases until 3:00 pm, at which time the savings rate is 50% for riding the car ride-sharing or the public bus. However, it is equal to 0 at 11:59 pm.

Table 4.3: Saving rate of average user satisfaction in the case of using a car ride-sharing only

Passenger Number	**Public bus and Car ride-sharing (%)**	**Car ride-sharing Only(%)**	**Saving Rate (%)**
[1-1500]	95	30	68.42
[1501-7000]	85	25	70.6
[7001-10000]	80	20	75

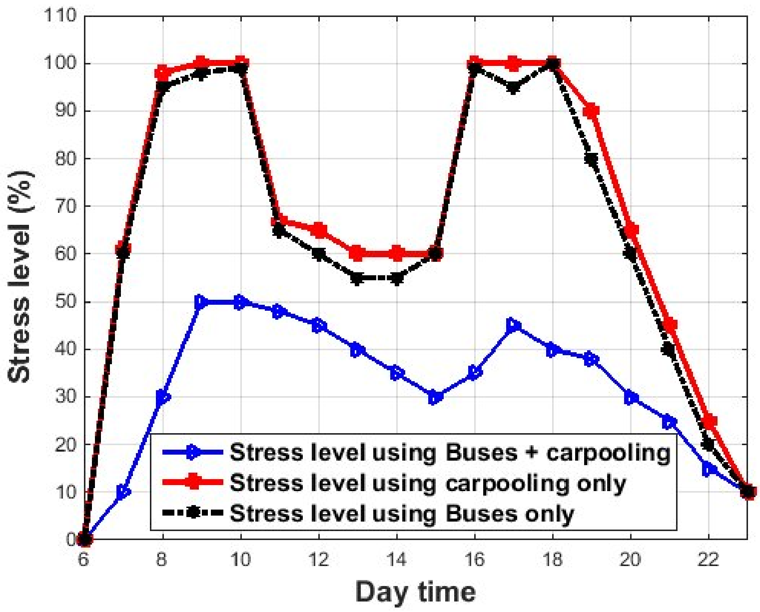

Figure 4.5: The average transportation stress level.

Conclusion

It can be seen from Figure 4.5 that during rush hours, the ratio of the stress level ascends, illustrating the rate of high stress when using any transport mode between 7 and 10 am and then between 3 and 6 pm. In addition, we noticed that the stress level is lower when using two transportation modes together compared to the stress level when using only one transportation mode. By providing this feature, DMT not only reduces the time and fare for the user to arrive at its destination, but also helps to encourage commuters to use car ride-sharing, reduce the amount of manufacturing, and put pressure on governments to reduce the cost of running a vast number of buses. Thus, this can ultimately minimize the environmental damage caused by the use of a huge number of public buses with their correlated GHG emissions and reduce the government's transportation budget.

Note: The error bars in the plots are exceptionally small, reflecting minimal variability within the data and a high degree of precision in the results. This indicates that the measured values are closely grouped around the average, resulting in a narrow confidence interval. Consequently, the error bars may not be visually prominent in the graphs. Given their negligible impact on the visualization, they were not included in the thesis plots, underscoring the consistency and reliability of the findings.

Table 4.4: Saving rate of average stress level in the case of using a public bus or car ride sharing

Day Time	**Public Bus and Car ride-sharing (%)**	**Car ride-sharing or Public Bus(%)**	**Saving Rate (%)**
[6:00 AM-11:00 AM]	49	65	24.6
[11:01 AM-3:00 PM]	30	60	50
[3:01 PM-11:59 PM]	10	10	0

4.4.3 Comparison Between DMT and a Multi-load Model

To prove the effectiveness of our proposal, we compare our DMT performance with the heuristic multi-load model, which is presented in [140].

4.4.3.1 A multi-load System Overview

The authors in [140] developed a new mechanism to deal with the multi-load school bus routing problem. This mechanism is the advanced version of the mixed load model. The mixed loading theory solves the bus routing problem in which students from different schools use the same bus at the same time. On the other hand, the multi-load model allows students from different schools and in different periods (morning, afternoon, evening, and full-time) to ride the same bus, regardless of whether the students are being taken to school or brought back home.

The main objective in [140] is to build a model with a reduced daily cost of routing in the city. According to statistical data from the Brazilian government, the charge is approximately 36 million U.S. dollars annually when using a mixed-load model and 34 million U.S. dollars annually when using a multi-load model. Four heuristics, each created using a different Iterated Local Search, have been developed to determine which mechanism will yield the best result. The first heuristic (H1) is called classic ILS with a normal local search.

On the other hand, the second heuristic (H2) adds a new local search called SmallVRPTW to the set of neighbourhood structures. In the third heuristic (H3), the authors used an elite set with a diversification strategy to avoid the local optima problem. Finally, the fourth heuristic

(H4) contains more smart criteria to decide the following: 1) which solution to use in the VND loop, 2) which solution to remove from the elite set, and 3) when to add a solution to the elite set.

For perturbation, 1) Classic Perturbation is used in H1, and 2) Trip Perturbation is used in H1, H2, H3, and H4.

A simulation was implemented to evaluate and compare these four heuristics, using real data from school bus routing in Espírito Santo, Brazil, within 12 hours, regardless of whether students were going to school or returning home. Based on the results, the authors conclude that H4 is superior to the other three heuristics in terms of the least cost and the shortest travel time for transporting passengers since it saves approximately two US $2 million.

4.4.3.2 Main Differences Between DMT and A multi-load Model

Although the work in [140] and our work in this paper both focus on finding a more cost-effective and less time-consuming model for travellers within cities, there are fundamental differences between them. First of all, H4 is based only on the school bus, while DMT is based on the bus and car ride-sharing, thereby providing passengers with more options to reach their destinations. Secondly, H4 deals with constant data for bus movement and student mobility. It is different from the DMT model, which depends on the dynamic information about the presence of public buses and car ride-sharing. As mentioned earlier, the passenger receives instantaneous data about the movement and place of the vehicle during the trip. Thirdly, the output of H4 is calculated based on the average values of time and cost for the bus trip within 12 hours. At the same time, the DMT relies on the model's dynamic information within 24 hours to calculate the satisfaction of each passenger immediately after the trip ends. Figures 4.6 and 4.7 show the inputs and outputs of the multi-load model and the DMT, respectively.

In this section, we re-run the multi-loading algorithm in [140] using MATLAB from 6 a.m. to 11:00 p.m. and compare the results with those obtained from our work (DMT).

1. Average user satisfaction

In multi-loading, user satisfaction is the number of students who arrived at their destinations without delay divided by the total number of students.

Figures 4.8 and 4.9 compare the average user satisfaction for the DMT and a multi-loading algorithm for 40,000 and 100,000 passengers, respectively. We define user satisfaction as the number of passengers since the trip's start divided by users who responded positively. From Figure 4.8, we notice that the satisfaction rate of passengers using DMT or a multi-loading model decreases with increasing passenger density. The average user satisfaction when using DMT is higher than the average user satisfaction when using a multi-loading system. It can be seen from Table 4.5 that the average user satisfaction when using DMT is higher than the average user satisfaction when using multi-loading. Similarly, the average satisfaction level does not change significantly as the number of passengers increases compared to that of the multi-load.

2. The stress level of passengers

At the same time, the stress level is the number of school buses used by the students divided by the total number of buses in the simulation.

Figure 4.10 compares the stress level of passengers when using DMT and the multi-loading algorithm. For DMT and multi-load models, the proportion increased significantly in the morning period from 6 am to 9 am, reaching approximately 50%. For DMT, the stress level decreases

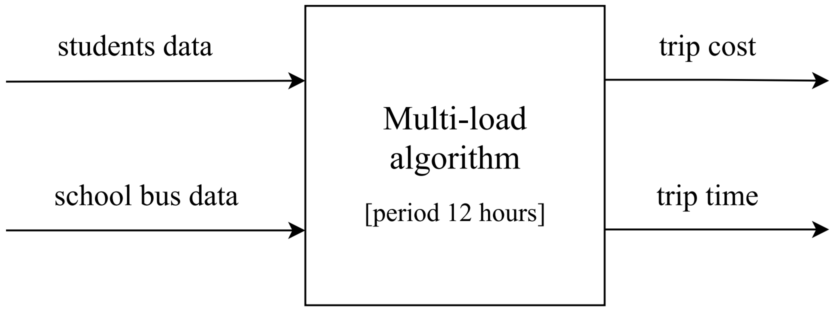

Figure 4.6: *The inputs and the outputs of the multiload*

Table 4.5: Average user satisfaction savings rate in the case of using the DMT or the multi-load model

Passenger Number Interval	**DMT (%)**	**Multi-load Model(%)**	**Saving Rate (%)**
[1-40000]	72	59	18.06
[40001-60000]	70	45	35.71
[60001-100000]	65	20	69.23

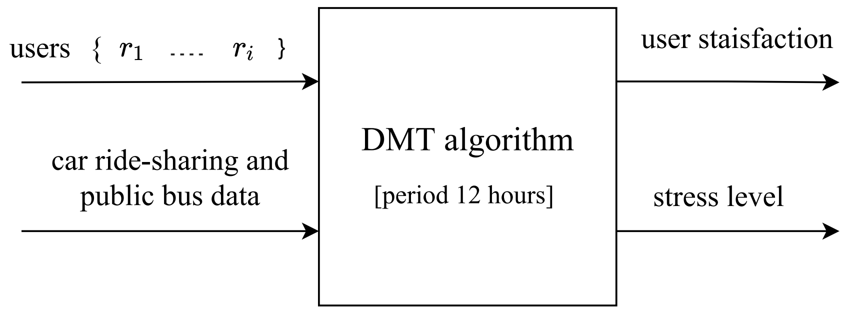

Figure 4.7: The inputs and the outputs of the DMT model

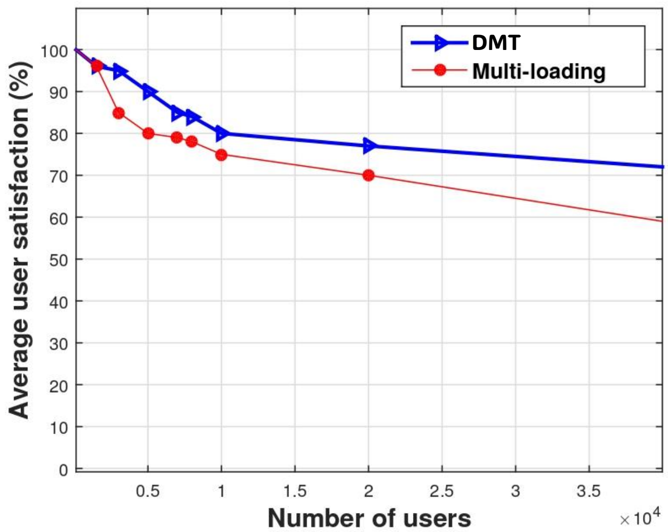

Figure 4.8: Average user satisfaction for the DTM and the multi-load model when the number of users is 40,000

and increases during the day until 5 pm. At 5 pm, the stress level was 45%, after which the percentage decreased until it reached 10% at 11:00 pm. In contrast, for the multi-loading model, the level increases smoothly and does not change significantly throughout the day.

From Table 4.6, we find that the savings rate of the average stress level between 6:00 am and 9:00 am is approximately 9.09% when using a multi-loading system. This ratio increases when the time reaches 3:00 pm. At 3:00 p.m., the savings rate is around 28.57%. However, the maximum saving rate was reached at 11:59 pm, with 84.61%.

We can conclude that DMT provides superior performance than the multi-loading algorithm

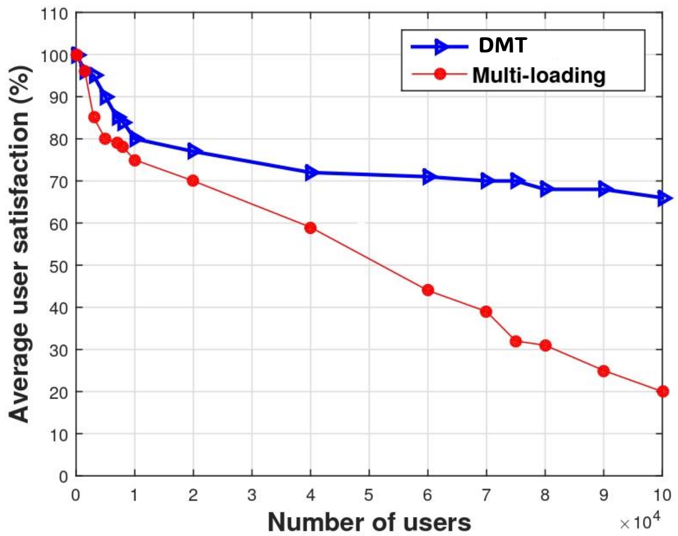

Figure 4.9: Average user satisfaction for the DMT and the multi-load model when the number of users is 100,000

because it is based on measuring the satisfaction of individual passengers. Unlike the work of [140], DMT can dynamically and instantly update information about public buses, ride-sharing, and passengers.

Table 4.6: Average stress level saving rate in the case of using the DMT or the multi-load model

Day Time	**DMT (%)**	**Multi-load Model(%)**	**Saving Rate (%)**
[6:00 AM-9:00 AM]	50	55	9.09
[9:01 AM-3:00 PM]	30	62	51.61
[3:01 PM-5:00 PM]	45	63	28.57
[5:01 PM-12:00 PM]	10	65	84.61

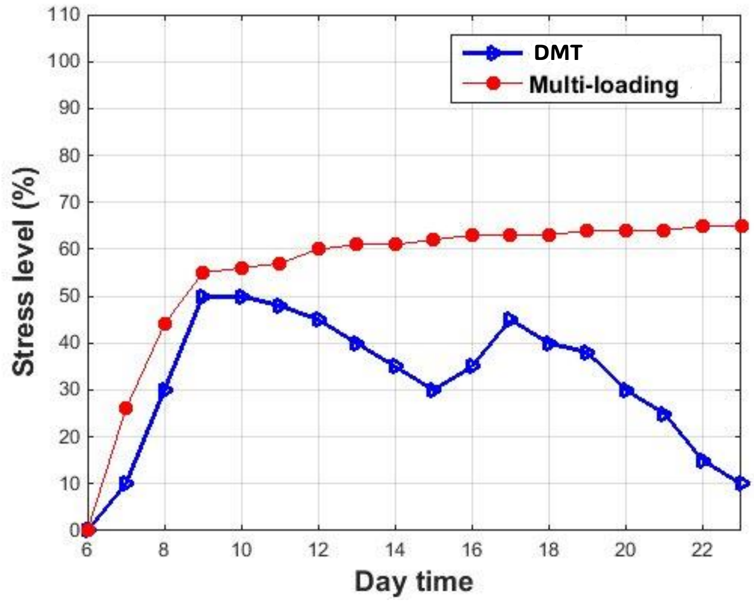

Figure 4.10: Average stress level for the DMT and the multi-load model when the number of users is 24,000

Chapter 5 Game Theory Multi Mode Transport (GT-MMT) Optimization Model

5.1 Model Overview

The Game Theory Multi-Mode Transportation (GT-MMT) model is a sophisticated and centralized approach to optimizing urban transportation systems by integrating multiple modes of transportation, including buses, ride-sharing services, and bicycles. The core idea behind GT-MMT is to utilize game theory principles, particularly Stackelberg game theory, to model and optimize the interactions between different transportation modes and passengers. This model aims to enhance the overall efficiency of the transportation network, minimize travel time and costs for passengers, and increase the utilization of various transportation modes.

In the GT-MMT model, passengers are considered leaders who make decisions based on their preferences, minimizing travel time and costs. The transportation modes (buses, ride-sharing, and bicycles) act as followers, adjusting their operations (routing and scheduling) in response to the passengers' decisions. This hierarchical decision-making structure, characteristic of Stackelberg game theory, ensures that the strategies of the transportation modes are aligned with the passengers' preferences, leading to a more efficient and user-centric transportation system.

The centralized control unit in the GT-MMT scheme plays a crucial role in coordinating the operations of different transportation modes. This central unit collects real-time data from various sources, processes this information, and uses it to make informed decisions about routing, scheduling, and resource allocation. By centralizing these operations, the GT-MMT model can effectively manage the entire transportation network, ensuring that all modes work together harmoniously.

To achieve these goals, the GT-MMT model employs advanced optimization algorithms, including Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), in conjunction with Stackelberg game theory. Genetic Algorithms are used to explore a wide range of potential solutions, while Particle Swarm Optimization excels in fine-tuning these solutions in real time. The integration of Stackelberg game theory ensures that the optimization process accounts for the hierarchical relationship between passengers and transportation modes, optimizing the system based on these interactions.

The integration of bicycles into the multi-mode model adds another layer of complexity and flexibility, allowing for a more diverse and resilient transportation network. The GT-MMT model also leverages real-time data and advanced simulation tools to continuously monitor and adjust the transportation system, ensuring that it remains efficient and responsive to the needs of the passengers.

In summary, the GT-MMT model represents a comprehensive and innovative approach to urban transportation optimization. It utilizes Stackelberg game theory and advanced optimization algorithms within a centralized framework to create a more efficient, cost-effective, and user-friendly transportation system. The following sections will delve into the details of the GT-MMT model, including its objectives, constraints, variable notations, and the specific roles of GA, PSO, and Stackelberg game theory in achieving its goals.

5.2 Main components of the GT-MMT

The main components of the GT-MMT model include the following:

First: Transportation Modes

Fig 5.1 shows the main transportation modes of the GT-MMT, car ride-sharing, public buses, bicycles, and passengers.

Second: Central Control Unit (CCU)

The Central Control Unit (CCU) is a pivotal component of the GT-MMT model, ensuring efficient management and optimization of the multi-modal transportation system. It consists of two primary functions: Central Processing and Data Aggregation.

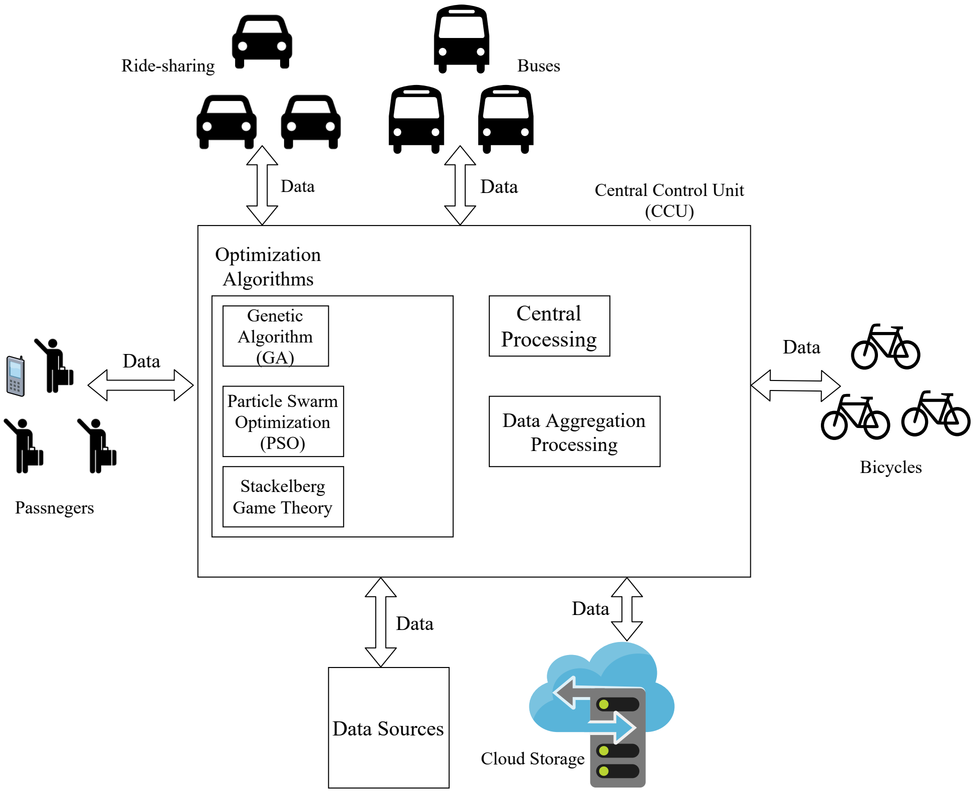

Figure 5.1: GT-MMT model overview

1. Central Processing

Central Processing is the core component responsible for:

- **Data Processing**: It handles the extensive data simulated through various tools, ensuring it is processed efficiently and accurately. This involves:
  - **Data Cleaning**: Removing any errors or inconsistencies in the data.
  - **Data Transformation**: Converting data into a usable format for analysis.
  - **Data Analysis**: Applying statistical and computational methods to extract meaningful insights from the data.
- **Decision-Making**: Based on the processed data, the Central Processing unit makes critical decisions regarding:
  - **Route Optimization**: Determining the most efficient routes for different transportation modes (buses, ride-sharing vehicles, bicycles).

  - **Scheduling**: Coordinating the schedules of various transportation modes to minimize waiting times and ensure timely transfers.
  - **Resource Allocation**: Distributing resources such as vehicles and drivers optimally to meet passenger demand.

- **Coordination**: It ensures seamless interaction and coordination between different components of the transportation system, such as:
  - **Communication with Vehicles**: Sending updated routes, schedules, and other instructions to vehicles.
  - **Passenger Information**: Providing passengers real-time travel plan updates.
  - **Emergency Management**: Responding to emergencies and unexpected situations promptly to minimize disruptions.

2. Data Aggregation Processing

Data Aggregation involves collecting and consolidating data from multiple sources to provide a comprehensive view of the transportation system. Given that we do not have an existing dataset, we use simulation tools to simulate real scenarios. This includes:

- **Data Collection**: Gathering data from various simulated sources such as:
  - **Vehicles**: Information on vehicle locations, speeds, available seats, and status (idle, moving, waiting).
  - **Passengers**: Data on passenger demands, trip requests, and preferences.
  - **Sensors**: Traffic conditions, environmental factors, and other relevant metrics collected from simulated roadside sensors, GPS devices, and other IoT devices.

- **Data Aggregation**: Combining and organizing the collected data into a unified format, which involves:
  - **Integration**: Merging data from disparate sources to create a single, cohesive dataset.
  - **Storage**: Storing the aggregated data in cloud storage for easy access and retrieval.

- **Real-Time Updates**: Continuously updating the data to reflect real-time conditions and changes in the transportation system.

- **Data Utilization**: Leveraging the aggregated data for various purposes, such as:
  - **Monitoring**: Keeping track of system performance, traffic conditions, and passenger demands.
  - **Optimization**: Using the data to optimize routes, schedules, and resource allocation.
  - **Prediction**: Analyzing historical and real-time data to predict future trends and demands, allowing for proactive management and decision-making.

3. Optimization Algorithms

- **Genetic Algorithms (GA)**: Used for exploring a wide search space to find near-optimal solutions for routing, scheduling, and resource allocation.
- **Particle Swarm Optimization (PSO)**: Excels in fine-tuning solutions in real-time, adapting to dynamic conditions.
- **Stackelberg Game Theory**: Models the hierarchical interaction between passengers (leaders) and transportation modes (followers), optimizing strategies based on these interactions.

Third: Data Collection and Sensing Infrastructure

The data collection and sensing infrastructure is crucial for gathering real-time information on various aspects of the transportation system. Given that we do not have an existing dataset, we use simulation tools to simulate real scenarios and to simulate the GPS, sensors, and mobile applications. This infrastructure includes:

- **Sensors**: These are deployed throughout the transportation network to monitor and collect data on traffic conditions, vehicle speeds, and environmental factors. Sensors can be placed at strategic locations such as intersections, bus stops, and along major routes.
- **GPS Devices**: Installed in vehicles, GPS devices provide real-time location data, enabling accurate tracking of vehicle movements. GPS data helps in determining the exact position

of buses, ride-sharing vehicles, and bicycles, which is essential for route optimization and scheduling.

- **Mobile application**: Mobile applications play a significant role in the data collection and user interaction process. Simulation mobile applications provide a platform for passengers to input their travel requests and receive real-time updates.

The collected data from sensors and GPS devices is used to monitor traffic flow, identify congestion points, and adjust routes dynamically to ensure efficient transportation. Real-time data enables the system to respond promptly to changing conditions, such as traffic jams or accidents, thereby minimizing delays and improving overall service reliability.

The key features of mobile applications include:

- **Travel Requests**: Passengers can enter their travel details, such as source and destination points, preferred departure times, and any specific mode preferences (e.g., ride-sharing, bus, bicycle).
- **Real-Time Updates**: The application provides passengers with real-time information on the status of their travel requests, including estimated arrival times of vehicles, current traffic conditions, and any delays or changes in schedules.
- **Notifications**: Passengers receive notifications about their trip status, such as vehicle arrival alerts, route changes, and any disruptions in service. This ensures that passengers are well-informed and can make timely decisions about their travel plans.

The integration of mobile applications with the transportation system enhances passenger experience by providing seamless access to transportation services, improving convenience, and ensuring timely and efficient travel. The data collected from mobile applications also feeds into the central system, helping to refine demand predictions and optimize resource allocation.

Fourth: Communications Infrastructure

1. Vehicle-to-Vehicle (V2V) Communications

Vehicle-to-vehicle (V2V) communication is a crucial component of the communication infrastructure in modern transportation systems. It enables direct vehicle communication, allowing them to share information and coordinate actions without relying on intermediary systems. Here are the key details and benefits of V2V communication:

- **Information Sharing:** V2V communication allows vehicles to share important information such as their current speed, direction, location, and braking status. This real-time exchange of data helps vehicles anticipate each other's movements and make informed decisions to enhance safety and efficiency.
- **Collision Avoidance:** One of the primary benefits of V2V communication is the ability to prevent accidents. By continuously sharing their position and speed data, vehicles can detect potential collisions and take proactive measures to avoid them. For example, if a vehicle suddenly brakes, it can immediately notify nearby vehicles to slow down or stop.
- **Platooning:** V2V communication enables the formation of vehicle platoons, where multiple vehicles travel closely together at high speeds with synchronized movements. This reduces air resistance and improves fuel efficiency. Vehicles within a platoon can communicate with each other to maintain near-optimal distances and speeds, enhancing overall traffic flow.
- **Cooperative Maneuvering:** Vehicles can coordinate complex manoeuvres such as lane changes, merging, and overtaking through V2V communication. This coordination improves traffic flow and reduces the likelihood of congestion or accidents caused by abrupt or unexpected movements.
- **Traffic Management:** V2V communication contributes to better traffic management by enabling vehicles to share information about traffic conditions, road hazards, and construction zones. This information helps drivers and automated systems make informed route choices, reducing travel time and congestion.

2. Vehicle-to-Infrastructure (V2I) Communications

Vehicle-to-infrastructure (V2I) communication facilitates communication between vehicles and the central control unit or other infrastructure components, such as traffic lights, road signs, and traffic management systems. V2I communication plays a vital role in optimizing transportation systems. Here are the key details and benefits of V2I communication:

- **Traffic Signal Coordination:** V2I communication allows vehicles to interact with traffic signals, enabling adaptive signal control based on real-time traffic conditions. Traffic signals can adjust their timings to optimize traffic flow, reduce waiting times, and minimize fuel consumption and emissions.
- **Dynamic Route Guidance:** Vehicles can receive real-time route guidance from the central control unit based on current traffic conditions, road closures, and incidents. This dynamic route guidance helps drivers avoid congested areas and find the fastest routes to their destinations.
- **Parking Management:** V2I communication can assist in efficient parking management by guiding vehicles to available parking spaces. Parking infrastructure can communicate occupancy status to vehicles, reducing the time spent searching for parking and alleviating congestion in parking areas.
- **Emergency Vehicle Priority:** V2I communication allows emergency vehicles to communicate with traffic signals and request priority passage. Traffic signals can adjust their timings to create a clear path for emergency vehicles, reducing response times and improving public safety.
- **Infrastructure Monitoring:** V2I communication enables continuous monitoring of infrastructure components such as bridges, tunnels, and roadways. Sensors embedded in the infrastructure can detect structural issues, wear and tear, and environmental conditions. Vehicles can communicate with these sensors to receive alerts and warnings about potential hazards.
- **Data Collection and Analysis:** V2I communication facilitates the collection of valuable data from vehicles, which can be used for traffic analysis, infrastructure planning, and

policy-making. The central control unit can analyze this data to identify traffic patterns, optimize transportation strategies, and improve overall system performance.

By implementing robust V2V and V2I communication systems, the transportation network can achieve higher levels of safety, efficiency, and reliability. These communication technologies enable real-time information exchange, proactive decision-making, and coordinated actions, ultimately enhancing the overall performance of the transportation system and improving the travel experience for passengers.

5.3 Objectives function and Problem Formulation

The GT-MMT model is designed to optimize the transportation network by incorporating a hierarchical decision-making process using the Stackelberg game theory. In this model, passengers act as leaders, and transportation modes (ride-sharing, buses, bicycles) act as followers. The primary objectives of the model are to minimize travel time and cost, maximize the number of passengers in ride-sharing and buses, and ensure bike availability.

Note: in the GT-MMT scheme, we can use the objective function term or the utility function instead of each other to represent the main goals of the entire model or the main goal for each player in the model. The total fitness function, based on the objective function, for the GT-MMT model, equals the utility function of the passenger (U_{p_i}) and the utility function of each mode (ride-sharing(U_{c_j}), bus (U_{b_k}), and bike-sharing (U_{bk_n})).

To calculate the objectives for each player in a Stackelberg-genetic model involving riders, ride-sharing, we need to define the mathematical formulations for the utility functions of both the leaders (riders) and the followers (ride-sharing). These calculations involve a series of mathematical expressions that consider various factors like travel time, distance, cost, capacity, and number of passengers.

$$Fitness(A) = \alpha_{p_i} \cdot U_{p_i}(A_{p_i}) + \beta_{p_i} \cdot U_{c_j}(A_{c_j}) + U_{b_k}(A_{b_k}) + U_{bk_n}(A_{bk_n}) \tag{5.1}$$

where $Fitness(A)$ is the total fitness for all the players in the models and A is the set of the strategies of the players.

For the passenger, the utility function is to minimize the total trip time (TT_{p_i}) and the total trip fare (TF_{p_i}) (5.2).

$$U_{p_i}(A_{p_i}) = \min \sum_{p_i=1}^{Nump} (\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}) \tag{5.2}$$

where α_{p_i} and β_{p_i} are the weighting factors for travel time and cost for passenger p_i, respectively. The trip time and fare equation in section core chapter...

For ride-sharing, the objective function is maximizing the vacant seats on it, as we can see in equation (5.3).

$$U_{c_j}(A_{c_j}) = \max(1 - \frac{Nump_{e_x}^{c_j}}{Ca_c}) \tag{5.3}$$

where $Nump_{e_x}^{c_j}$ is the number of passengers in the car in route e_x, and Ca_c is the maximum capacity of the car.

Buses aim to maximize the number of passengers while following a fixed route with predefined stops. We can represent maximizing the vacant seats in the public bus as equation (5.4)

$$U_{b_k}(A_{b_k}) = \max(1 - \frac{Nump_{e_x}^{b_k}}{Ca_b}) \tag{5.4}$$

where $Nump_{e_x}^{b_k}$ is the number of passengers in the bus b_k, and Ca_b is the maximum capacity of the bus.

For the bicycle, the primary goal is to maximize the usage rate of bicycles across the network, ensuring efficient utilization and high availability. This objective (5.7) can be formulated as follows:

$$U_{bk_n}(A_{bk_n}) = \max\left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)}\right) \tag{5.5}$$

where U_{bk_n} is the utility function for the usage rate of bicycles, $Numbk_{use}(I_l)$ denotes the number of bicycles in use at location I_l, $Numbk_{av}(I_l)$ represents the total number of bicycles available at location I_l, and nbk is the number of locations or docking stations.

So, the equation 5.6 becomes:

$$Fitness(A) = \min \sum_{p_i=1}^{Nump} (\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}) + \max(1 - \frac{Nump_{ex}^{c_j}}{Ca_c}) + \max(1 - \frac{Nump_{ex}^{b_k}}{Ca_b}) + \max\left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)}\right) \quad (5.6)$$

The constraints in the GT-MMT model ensure the feasibility and efficiency of the transportation system. Some key constraints include balancing vehicle departures and arrivals, matching rider requests with suitable transportation options, and adhering to maximum travel times and time windows. For instance, the constraint to ensure that the total distance of a passenger's trip is within the maximum allowable distance can be written as:

$$\sum_{i=1}^{n} D_{s_i,s_{i+1}} \leq MD_{p_i}^{m} \quad (5.7)$$

where $D_{s_i,s_{i+1}}$ is the distance between stops s_i and s_{i+1}, and $MD_{p_i}^{m}$ is the maximum allowable distance for the passenger.

In the core chapter, the full objective functions, constraints, and variables are elaborated in detail. These include mathematical formulations for the Stackelberg game model, the fitness evaluation in Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), and the iterative process for real-time dynamic adjustments. The variables encompass the set of transportation modes (M), set of passengers (P), source points ($S_1^{p_i}$), destination points ($D_{np}^{p_i}$), travel times (TT_{p_i}), costs (TF_{p_i}), and other relevant parameters required for optimizing the transportation network. The comprehensive explanation provides a thorough understanding of how the GT-MMT model integrates these elements to achieve its optimization goals.

The following algorithms show the operation of implementing the GT-MMT model in many steps. The final goal of the model can be implemented by evaluating these algorithms with all math equations, objective functions, and constraints related to the GT-MMT. For a comprehensive understanding of the full notations, objectives, and constraints associated with the GT-MMT model, please refer to subsections 3.2.1, 3.2.2, 3.4.1, 3.4.2, 3.4.3, and 3.4.4. These subsections provide detailed explanations of the parameters, mathematical formulations, and constraints that guide the optimization processes within the GT-MMT framework. The main inputs and outputs parameters in the GT-MMT model we can see in the Figure 5.2.

To make the problem easier to solve, we assume several relevant hypotheses. First, we did not consider the person's walking time between the stations on his or her trip. Second, the stations for the different vehicles are located in the same position. Third, while the passenger's origin is the first station of the journey, the last station is the destination. Finally, security and trust are assumed in this model.

By aligning the strategies of different transportation modes under a unified Stackelberg leadership framework, the GT-MMT facilitates a more coordinated and efficient urban mobility system. This thesis explores these alignments and the corresponding mathematical formulations that enable such strategic optimizations within these complex transportation networks.

5.4 GT-MMT Optimization Algorithm

5.4.1 Overview and Main Components of the Algorithm

In this thesis, we use the Stackelberg equilibrium game theory model to apply the competition play between the players [180]. The Stackelberg game theory, PSO, and GA provide a hierarchical structure for the GT-MMT model, which is a centralized model that uses them to satisfy the

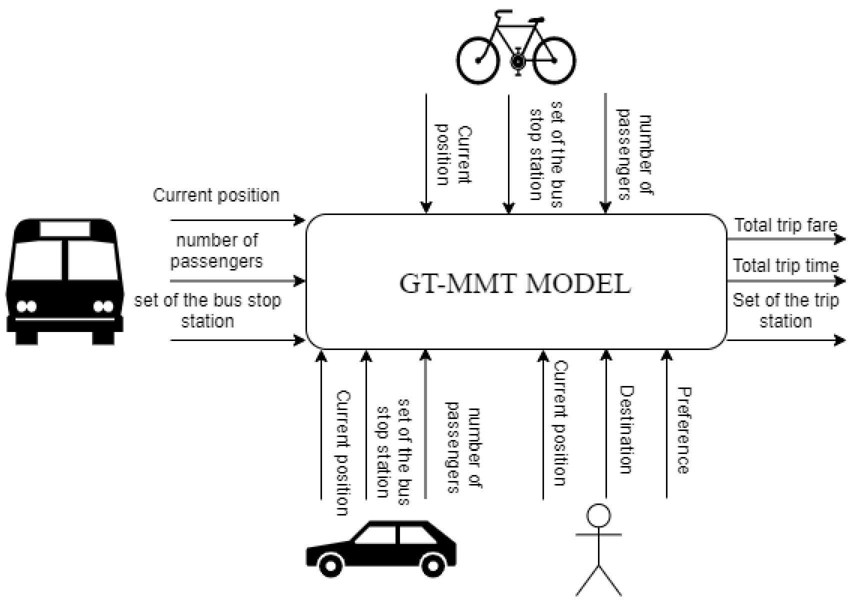

Figure 5.2: The main inputs and outputs parameters in the GT-MMT model

optimization. For a comprehensive understanding of the basic components and definitions of GA, PSO, and Stackelberg used in the GT-MMT scheme, please refer to section 3.5. But if you want to read the basic elements for each algorithm, please read subsection 3.5.2 for the GA, subsection 3.5.3 for the PSO, and subsection 3.5.4 for the Stackelberg game theory. These subsections provide summary explanations of these algorithms that guide the optimization processes within the GT-MMT framework. The central unit coordinates and optimizes the interactions between different transportation modes. The central unit collects data from all transportation modes and passengers, processes this data, and sends optimized instructions back to the transportation modes. This ensures a coordinated and efficient system-wide operation. For a comprehensive understanding of the full notations, objectives, and constraints associated with the GT-MMT model, please refer to subsections 3.2.1, 3.2.2, 3.4.1, 3.4.2, 3.4.3, and 3.4.4. These subsections provide detailed explanations of the parameters, mathematical formulations, and constraints that guide the optimization processes within the GT-MMT framework.

The following are outlines of the operation for each component in the algorithm. Stackelberg's game theory involves a leader-follower structure where passengers act as leaders by making decisions about their preferred travel routes and modes based on their optimization criteria, minimizing travel time and cost. The various transportation modes, including ride-sharing, buses, and bicycles, act as followers. These modes adjust their operational strategies, such as routing and scheduling, in response to the passengers' choices. The particle swarm optimization (PSO) and genetic algorithms (GA) process begins with the initialization phase, where GA generates an initial population of possible routing, scheduling, and resource allocation solutions. GA then evaluates these solutions based on fitness criteria, such as travel time and cost, and combines the near-optimal solutions to create new generations through selection and crossover. GA introduces random changes through mutation to explore new areas of the solution space. Finally, PSO is used to refine these solutions in real time, quickly adapting to changing conditions and further optimizing the strategies.

In the following subsection, we discuss the game model used in the GT-MMT by illustrating the steps of the optimization algorithm used to calculate the utility function (objective functions) for all players.

5.4.2 Details Operation of the Algorithm

In this subsection, we will present the main algorithm and sub-algorithms for the GT-MMT model. Each algorithm will be explained in detail, including the steps, mathematical notations, and equations involved. These algorithms encompass the genetic algorithm (GA) for initial strategy generation, particle swarm optimization (PSO) for real-time dynamic adjustments, and Stackelberg game theory for hierarchical decision-making. The integration of these techniques ensures that the GT-MMT model achieves near-optimal performance in managing multi-mode transpor- tation systems. The GT-MMT model consists of a main algorithm, the "GT-MMT Optimization Algorithm" 2, which outlines the overall optimization process and several sub-algorithms, each dedicated to specific steps within the main algorithm. These include "Initial Strategy Generation with Genetic Algorithms (GA)" 3, "Real-Time Dynamic Adjustments with PSO and Stacklberg" 4.

Step 1: Problem Definition and Initialization

Define Problem Space: The main inputs are needed for the GT-MMT algorithm in table 5.1. These inputs include the passenger data, the transportation modes data, network and traffic information, optimization parameters, objective functions, and their main constraints. For a comprehensive understanding of the full inputs, notations, objectives, and constraints associated with the GT-MMT model, please refer to subsections 3.2.1, 3.2.2, 3.4.1, 3.4.2, 3.4.3, and 3.4.4. These subsections provide detailed explanations of the parameters, mathematical formulations, and constraints that guide the optimization processes within the GT-MMT framework. To read more about this thesis's topology and map, read section 3.3.

The variables in the problem space include the set of transportation modes V (e.g., ride-sharing, bus, bicycle), the set of passengers P, source points for each passenger S_p, destination points for each passenger D_p, travel time T, and cost C. The objective functions are formulated to minimize travel time T and cost C, to increase the number of passengers in ride-sharing and buses, and to ensure bike availability. To read the full objectives function, see section 5.3.

Data Collection: In our work, we simulate the real-world scenario using various simulation tools and programs. This involves gathering simulated real-time data on traffic conditions, passenger preferences, and the operational status of various transportation modes. By leveraging these tools, we create a comprehensive and dynamic dataset that mimics the complexities and variations of actual transportation systems.

Table 5.1: Inputs parameters for the GT-MMT algorithm

Passenger Information (P)	
$Numbp$	Number of passengers needing transport
P_{pref}	Preferences regarding cost, time, and mode of transport
Transportation Modes (M)	
M	Details of available transportation modes (buses, ride-sharing, bike-sharing)
Ca_m	Capacity of each transportation mode
MF_m	Pricing model for each transportation mode
MT_m	Threshold travel time for each mode of transportation
Network and Traffic Information ($P_{\textbf{Veins}}$)	
Network Topology	Map of the transportation network, including routes and connectivity
Service Schedules	Timetables for public transportation modes
Optimization Parameters ($P_{\textbf{OSM}}$)	
PS_{ga}	Number of the chromosomes in the population for the GA
NP_{pso}	Number of Particles for PSO
I	The number of iterations the algorithm will run
PO_o	Initial population for the GA
Weight Factors (α, β, γ)	Weight factors for the objective function
Objective Functions	
Leader's Objective Function	Objective function for the passenger (minimize trip time, cost, maximize comfort)
Followers' Objective Functions	Objective functions for transportation modes (maximize utilization, minimize operational cost, maximize profit)
Constraints	

Operational Constraints	Constraints related to the operational aspects of each transportation mode and passenger

Algorithm 2 GT-MMT Optimization Algorithm

Input: Number of chromosome (PS_{ga}), number of particles(NP_{pso}), number of iterations I, weight factors α, β, γ (for the full inputs,read 3.2.1, 3.2.2, 3.4.1, 3.4.2, 3.4.3, and 3.4.4

Output: the near-optimal combination of modes for the passengers and the near-optimal strategies

for $alg = 1$ ***to*** I **do**

Step 1: Problem Definition and Initialization
Define the problem space, variables, and objective functions. Gather simulated real-time data on traffic conditions, passenger preferences, and operational status of various transportation modes.

Step 2: Initial Strategy Generation with Genetic Algorithms (GA)
Generate initial population for GA, evaluate initial population, and perform selection, crossover, and mutation. This includes creating random initial strategies for routing, scheduling, and pricing for all transportation modes. (Refer to Algorithm 3)

Step 3: Real-Time Dynamic Adjustments with Particle Swarm Optimization (PSO)
Initialize PSO, evaluate fitness dynamically based on real-time data, update personal best and global best, update velocity and position, adjust routes, schedules, and strategies, and iterate until convergence criteria are met. Then, integrate the outputs into the Stackelberg game theory model.(Refer to Algorithm 4)

if $|Fitness_{current} - Fitness_{previous}| < \epsilon$ **then**

Exit loop

return the near-optimal combination of modes for each passenger and the near-optimal strategies

Step 2: Initial Strategy Generation with Genetic Algorithms (GA)

Process: Create initial strategies or the population (PO) for the passenger and for each transportation mode (ride-sharing, bus, bicycle). These strategies (A) are encoded as chromosomes, representing the potential solutions for the optimization problem. This involves evaluating the fitness of each strategy, selecting the best-performing ones, and applying crossover and mutation to create new strategies. The fitness function combines multiple objectives: minimizing travel time and fare for the passenger trip, increasing the number of passengers in ride-sharing and buses, and increasing

Algorithm 3 Initial Strategy Generation with Genetic Algorithms (GAs)

Input: Initial population PO_0

Output: New population PO_{new} (Optimized initial strategies)

Generate Initial Population: Create a diverse set of initial strategies for routing, scheduling, and pricing for all transportation modes. Each strategy is represented as a chromosome in the genetic algorithm.

for $g = 1$ ***to*** Gt **do**

Evaluate Initial Population: Evaluate the fitness of each strategy based on predefined performance metrics.

$$Fitness(CH) = U_{p_i} + U_{c_j} + U_{b_k} + U_{bk_n} \tag{5.8}$$

Selection, Crossover, and Mutation:

- **Selection:** Select the best-performing strategies using a roulette wheel selection method.
$$P_{selected} = \text{Select}(P_0, \text{Fitness(CH)})$$
- **Crossover:** Perform crossover to generate new offspring by combining parts of two parent solutions.
$$\text{Offspring} = \text{Crossover}(PO_{selected})$$
- **Mutation:** Apply mutations to the offspring to introduce variations.
$$\text{Mutated Offspring} = \text{Mutation(Offspring)}$$
- **Replacement:** Replace less fit individuals in the population with new offspring.
$$PO_{new} = \text{Replace}(PO_0, \text{Mutated Offspring})$$

if $|Fitness(CH)^{(g)}_{best} - Fitness(CH)^{(g-1)}_{best}| < \epsilon$ *for a predefined number of iterations* **then**

Exit loop

Integrate outputs from GA into the PSO initialization.

return New population PO_{new} (optimized initial strategies).

the usability of the bike. The process of this step can be divided into five operations, as follows:

1. Create initial strategies (population)

The population is represented by multiple chromosomes. A chromosome (CH) is a vector that encodes a sequence of passenger, ride-sharing, bus, and bike-sharing strategies in one combined structure. Randomly assign passengers to chromosomes while ensuring diversity:

$$\text{Passenger Group}_i = \{p_1, p_2, \dots, p_n\} \quad \text{for} \quad i = 1, 2, \dots, C$$

where n is the number of passengers assigned to each chromosome.

Here's how to represent a population of size n:

$$\mathbf{PO} = \begin{bmatrix} \mathbf{CH}_1 \\ \mathbf{CH}_2 \\ \vdots \\ \mathbf{CH}_n \end{bmatrix} \tag{5.9}$$

Where each $\mathbf{CH}_i$ is a chromosome representing the initial strategies for all players in the GT-MMT model. To read the details of the parameters and structure fare trip, see the section 3.4.1.

The combined chromosome for the GA in the GT-MMT model is a concatenation of these vectors:

$$\mathbf{CH} = \left[\mathbf{CH}_{\text{passenger}}, \mathbf{CH}_{\text{ride-sharing}}, \mathbf{CH}_{\text{bus}}, \mathbf{CH}_{\text{bike}} \right] \tag{5.10}$$

The passenger's strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{passenger}} = \begin{bmatrix} M_{\text{passenger}} & R_{\text{passenger}} & T_{\text{dep}} & F_{\text{passenger}} & M_{\text{transfer}} & U_{\text{passenger}} \end{bmatrix} \tag{5.11}$$

The component $M_{\text{passenger}}$ represents the mode of transportation selected by the passenger, such as a bus, ride-sharing, or bike-sharing. The $R_{\text{passenger}}$ component denotes the route chosen by the passenger for the journey. The departure time, indicated by T_{dep}, specifies when the passenger starts the journey. The $P_{\text{passenger}}$ component refers to the pricing strategy or cost consideration for the trip. The M_{transfer} component represents the strategy for transferring between modes during the journey. Finally, the $U_{\text{passenger}}$ component reflects the comfort and utility preferences of the passenger, which influence their satisfaction with the trip.

The bus's strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{bus}} = \begin{bmatrix} R_{\text{bus}} & Scht_{\text{bus}} & Ca_{\text{bus}} & F_{\text{bus}} \end{bmatrix} \tag{5.12}$$

The component R_{bus} represents the route selection strategy for the bus. The scheduling strategy, denoted by $Scht_{\text{bus}}$, determines the departure and arrival times as well as the service frequency. The capacity management strategy, indicated by Ca_{bus}, is focused on optimizing the use of the bus's seating and standing capacity. The pricing strategy, F_{bus}, involves setting the fare structure for bus services.

The ride-sharing strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{ride-sharing}} = \begin{bmatrix} R_{\text{ride-sharing}} & S_{\text{ride-sharing}} & C_{\text{ride-sharing}} & P_{\text{ride-sharing}} \end{bmatrix} \tag{5.13}$$

The component $R_{\text{ride-sharing}}$ represents the route selection strategy for the ride-sharing vehicle. The pickup and drop-off scheduling strategy, denoted by $S_{\text{ride-sharing}}$, determines the timing and sequence of passenger pickups and drop-offs. The capacity management strategy, indicated by $C_{\text{ride-sharing}}$, optimizes the utilization of the ride-sharing vehicle's seating capacity. Lastly, the pricing strategy, $P_{\text{ride-sharing}}$, involves setting the fare structure for ride-sharing services.

The bike-sharing strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{bike-sharing}} = \begin{bmatrix} S_{\text{bike-sharing}} & A_{\text{bike-sharing}} & M_{\text{bike-sharing}} & P_{\text{bike-sharing}} \end{bmatrix} \tag{5.14}$$

The component $S_{\text{bike-sharing}}$ represents the station placement and distribution strategy for the bike-sharing system. The bike availability management strategy, denoted by $A_{\text{bike-sharing}}$, ensures that bikes are available where and when needed. The maintenance and re-balancing strategy, indicated by $M_{\text{bike-sharing}}$, keeps bikes in good condition and properly distributed. Finally, the pricing strategy, $P_{\text{bike-sharing}}$, involves setting the fare structure for bike-sharing services.

2. Evaluate Initial Population

Fitness Evaluation: Fitness is calculated for each chromosome, generally combining the utilities for passengers and transportation modes. So the equation 5.15 can be written as:

$$Fitness(CH) = U_{p_i}(CH_{p_i}) + U_{c_j}(CH_{c_j}) + U_{b_k}(CH_{b_k}) + U_{bk_n}(CH_{bk_n}) \tag{5.15}$$

where $Fitness(CH)$ is the total fitness for all the players in the models, and CH is the set of chromosomes. The fitness function, which is based on the objective function, evaluates each chromosome based on the collective performance metrics, including total trip time, total fare, and capacity utilization. The goal is to find solutions that optimize these metrics for the entire group of passengers.

$$\text{Fitness}(\mathbf{CH}_i) = \alpha \cdot \left(\frac{1}{|P_i|} \sum_{j=1}^{|P_i|} TT_{p_i} \right) + \beta \cdot \left(\frac{1}{|P_i|} \sum_{j=1}^{|P_i|} TF_{p_i} \right) + \gamma \cdot \left(\frac{1}{M} \sum_{k=1}^{M} (1 - \frac{Nump_{e_x}^{c_j}}{Ca_c}) + \right) + \left(\frac{1}{M} \sum_{k=1}^{M} (1 - \frac{Nump_{e_x}^{b_k}}{Ca_b}) + \right) + \left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)} \right) \tag{5.16}$$

Where α, β, γ are weight factors that balance the importance of travel time, fare, and capacity utilization.

The parameter α_p is the weight factor for travel time for passenger p. The parameter β_p is the weight factor for cost for passenger p. The parameters γ_c, γ_b, and γ_{bk} are the weight factors for the number of passengers for the ride-sharing, bus, and bicycle, respectively. Additionally, $Numberc$, $Numberb$, and $Numberbk$ represent the number of ride-sharing, buses, and bicycles, respectively.

The total trip time and total fare for the group of passengers represented by a chromosome in the GT-MMT model are calculated by aggregating the individual travel times and fares for each passenger. Each passenger has their own origin and destination, so the calculations take into account these individual routes.

Calculate the travel time for each passenger based on their origin and destination, then sum these travel times to get the total trip time for the group. For a chromosome $\mathbf{CH}_i$ representing a group of passengers, the total trip time is the sum of the individual travel times:

$$TT(CH_i) = \sum_{j=1}^{|PCH_i|} TT_{p_j} \tag{5.17}$$

Where $TT(CH_i)$ represents the total trip time for chromosome $\mathbf{CH}_i$. The notation $|PCH_i|$ indicates the number of passengers assigned to chromosome $\mathbf{CH}_i$. Finally, TT_{p_j} refers to the travel time for passenger TT_{p_j}.

Calculate the fare for each passenger based on their route and mode of transport. Sum these fares to get the total fare for the group. Similarly, the total fare for a chromosome $\mathbf{C}_i$ is the sum of the individual fares:

$$TF(CH_i) = \sum_{j=1}^{|PCH_i|} TF_{p_j} \tag{5.18}$$

Where $TF(CH_i)$ represents the total trip fare for chromosome $\mathbf{CH}_i$. The notation $|PCH_i|$ indicates the number of passengers assigned to chromosome $\mathbf{CH}_i$. Finally, TF_{p_j} refers to the travel fare for passenger TF_{p_j}.

The GT-MMT genetic algorithm's selection process effectively guides the evolutionary search toward better transportation strategies by focusing on high-fitness chromosomes. This shows how preferences and operational efficiencies interact in a multi-mode transportation system.

3. Selection

Selection: Select the best-performing strategies from the initial population using a roulette wheel selection method.

Equations:

$$PO_{\text{selected}} = \text{Select}(PO_o, \text{Fitness}) \tag{5.19}$$

The selection mechanism ensures that chromosomes leading to higher fitness are more likely to reproduce. This means that successful strategies will be passed on to the next generation, improving the quality of the solution as a whole. The primary goal is to select chromosomes representing effective transportation strategies, ensuring that beneficial traits are passed on to subsequent generations.

The probability of selection is proportional to the chromosome's fitness computed as:

$$P(CH_i) = \frac{Fitness(CH_i)}{\sum_{j=1}^{N} Fitness(CH_j)} \tag{5.20}$$

Where CH_i represents the i-th chromosome in the population. The notation $F(CH_i)$ denotes the fitness of the i-th chromosome, quantifying how well it meets the GT-MMT model's objectives. Finally, $P(CH_i)$ refers to the selection probability of the i-th chromosome.

4. Crossover

Crossover: Combine parts of two parent solutions to generate new offspring. This process is called crossover.

Equations:

$$\text{Offspring} = \text{Crossover}(PO_{\text{selected}}) \tag{5.21}$$

Use genetic operations to generate new chromosomes, ensuring a mix of passengers:

$$\mathbf{CH}_{\text{new}} = \text{Crossover}(\mathbf{CH}_{\text{parent1}}, \mathbf{CH}_{\text{parent2}}) \tag{5.22}$$

Offspring: Offspring are new strategies generated from the crossover and mutation of selected parent strategies. They introduce diversity into the population and help explore new potential solutions.

In this thesis, the uniform crossover is used. Uniform crossover is a pivotal operation in genetic algorithms used to combine the genetic information of two parent chromosomes to produce offspring. In the GT-MMT model, uniform crossover allows for the independent consideration of each strategy component for recombination for optimizing transportation strategies. This facilitates the exploration of innovative solutions that efficiently balance the objectives of passengers and transportation providers. The ability of uniform crossover to generate diversified solutions is crucial for avoiding local optima and finding effective strategies for complex urban transportation systems. This operation is essential because it introduces new strategy combinations and enhances genetic diversity within the population, which is crucial for exploring a vast solution space and avoiding premature convergence.

In the GT-MMT framework, the uniform crossover operation is particularly important as it allows for combining different strategies from two-parent solutions. In this context, the leader is the passenger, and the followers are the ride-sharing, bus, and bike transportation modes. The uniform crossover operation can lead to offspring that inherit various advantageous traits from their parents, potentially improving the overall system performance.

Let $\mathbf{CH}_i$ and $\mathbf{CH}_j$ be two parent chromosomes. Each chromosome consists of strategies for the passenger and the transportation modes. The two parents can be represented as:

$$\mathbf{CH_i} = \left[\mathbf{CH}_{\text{passenger}}, \mathbf{CH}_{\text{ride-sharing}}, \mathbf{CH}_{\text{bus}}, \mathbf{CH}_{\text{bike}}\right] \tag{5.23}$$

$$\mathbf{CH_j} = \left[\mathbf{CH}_{\text{passenger}}, \mathbf{CH}_{\text{ride-sharing}}, \mathbf{CH}_{\text{bus}}, \mathbf{CH}_{\text{bike}}\right] \tag{5.24}$$

During the uniform crossover operation, an offspring chromosome $\mathbf{CH}_k$ can inherit some strategies from parent 1 (CH_i) and other strategies from parent 2(CH_j). Mathematically, this can be represented as:

The passenger's strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}^{new}_{\text{passenger}} = \begin{bmatrix} M^i_{\text{passenger}} & R^i_{\text{passenger}} & T^j_{\text{dep}} & F^i_{\text{passenger}} & M^j_{\text{transfer}} & U^j_{\text{passenger}} \end{bmatrix} \tag{5.25}$$

the mode of transportation M^i_p from parent $\mathbf{C}_i$ and the departure time D_p from parent $\mathbf{C}_j$.

The bus's strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}^{new}_{\text{bus}} = \begin{bmatrix} R^i_{\text{bus}} & Scht^j_{\text{bus}} & Ca^j_{\text{bus}} & F^i_{\text{bus}} \end{bmatrix} \tag{5.26}$$

The ride-sharing strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}^{new}_{\text{ride-sharing}} = \begin{bmatrix} R^j_{\text{ride-sharing}} & S^i_{\text{ride-sharing}} & C^i_{\text{ride-sharing}} & P^j_{\text{ride-sharing}} \end{bmatrix} \tag{5.27}$$

The bike-sharing strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}^{new}_{\text{bike-sharing}} = \begin{bmatrix} S^i_{\text{bike-sharing}} & A^i_{\text{bike-sharing}} & M^j_{\text{bike-sharing}} & P^i_{\text{bike-sharing}} \end{bmatrix} \tag{5.28}$$

So, the new strategies (chromosomes) mathematically can be represented as:

$$CH_{new} = \left[\mathbf{CH}^{new}_{\text{passenger}}, \mathbf{CH}^{new}_{\text{ride-sharing}}, \mathbf{CH}^{new}_{\text{bus}}, \mathbf{CH}^{new}_{\text{bike}}\right] \tag{5.29}$$

Where CH_{new} is derived from the respective parent strategies using the binary uniform crossover mask operation:

1. **Select Parents** Identify two parent chromosomes, $\mathbf{CH}_i$ and $\mathbf{CH}_j$, which will be used for crossover.

2. **Generate Binary Crossover Mask** Create a binary mask of the same length as the chromosomes. Each position in the mask is a binary value (0 or 1) chosen randomly.

   Example: If the length of the chromosome is 5, a possible binary mask could be $\mathbf{M} = [1, 0, 1, 0, 1]$.

3. **Apply Binary Mask to Create Offspring** The binary mask determines which genes (positions) to take from each parent. For each gene position k:

   - If the mask value $\mathbf{M}_k$ is 1, take the gene from parent $\mathbf{CH}_i$.
   - If the mask value $\mathbf{M}_k$ is 0, take the gene from parent $\mathbf{CH}_j$.

4. **Construct Offspring Chromosome** Combine the selected genes to form the new offspring chromosome $\mathbf{CH}_k$.

5. Mutation

Mutation: Apply random changes to the offspring to introduce variations. The mutation alters one or more genes in the chromosome to explore new solutions.

Equations:

$$\mathbf{CH}_{\text{mutated}} = \text{Mutation}(\mathbf{CH}_{\text{new}}) \tag{5.30}$$

The mutation process in the GT-MMT model's genetic algorithm introduces random changes to offspring chromosomes, facilitating the exploration of new strategic possibilities and enhancing genetic diversity. In the GT-MMT, the mutation operation within the genetic algorithm plays a pivotal role in introducing random variations to the offspring chromosomes. This mechanism is instrumental in exploring new strategic possibilities and preserving genetic diversity, which is vital for avoiding local optima and ensuring a thorough exploration of the solution space.

We integrate mutation probability and mutation functions with the steps of applying the five mutation operators by following these steps:

1. **Define Mutation Probability:** Each gene in the chromosome has a mutation probability μ.

2. **Mutation Functions:**

   - Define the mutation functions for each operator.

   - Apply the mutation functions based on the mutation probability.

3. **Apply the Five Operators:** Sequentially apply the mutation operators, but only if the random number generated is less than the mutation probability.

We define the following notations: $\mathbf{CH}$ is the chromosome, $\mathbf{A}_{\text{passenger}}$ is the passenger strategy vector, $\mathbf{A}_{\text{bus}}$ is the bus strategy vector, $\mathbf{A}_{\text{ride-sharing}}$ is the ride-sharing strategy vector, $\mathbf{A}_{\text{bike-sharing}}$ is the bike-sharing strategy vector, μ is the mutation probability, $\mathbb{P}$ is a random number between 0 and 1, and $\text{mutate}(x, \mu)$ is the mutation function for gene x with probability μ. The operation of the mutation can be as follows:

1. **Bit Flip Mutation**:

   Suitable for binary decisions in the strategy, such as whether a specific route is active or whether a certain transportation mode is available. It flips bits (0 to 1 or 1 to 0) in the chromosome to explore different binary configurations.

$$\text{mutate}_{\text{bit-flip}}(x, \mu) = \begin{cases} 1 - x & \text{if } \mathbb{P} < \mu \\ x & \text{otherwise} \end{cases} \tag{5.31}$$

2. **Swap Mutation**:

   Effective for sequences where the order is important, such as the sequence of stops in a bus route or the order of passenger pickups in ride-sharing. It randomly selects two positions in the chromosome and swaps their values, maintaining the permutation's integrity while allowing exploration of different sequence orders.

$$\text{mutate}_{\text{swap}}(x, \mu) = \begin{cases} \text{swap } (x_i, x_j) & \text{if } \mathbb{P} < \mu \\ x & \text{otherwise} \end{cases} \tag{5.32}$$

   Here, x_i and x_j are randomly chosen positions.

3. **Scramble Mutation**: Useful when a portion of the strategy (like pricing or scheduling) needs to be shuffled without altering the overall structure too much. It randomly selects a subset of the chromosome and scrambles the genes within that subset to introduce localized variation.

$$\text{mutate}_{\text{scramble}}(x, \mu) = \begin{cases} \text{scramble}(x[i:j]) & \text{if } \mathbb{P} < \mu \\ x & \text{otherwise} \end{cases} \tag{5.33}$$

   Here, $[i : j]$ is a randomly chosen subset.

4. **Inversion Mutation**: Reverses the order of a subset of genes within a chromosome, which is useful for strategies involving sequences, such as reversing the order of stops on a route. This mutation helps discover more efficient routing or scheduling configurations by exploring the reversed sequences.

$$\text{mutate}_{\text{inversion}}(x, \mu) = \begin{cases} \text{invert}(x[i:j]) & \text{if } \mathbb{P} < \mu \\ x & \text{otherwise} \end{cases} \tag{5.34}$$

   Here, $[i : j]$ is a randomly chosen subset.

5. **Gaussian Mutation**: This mutation is ideal for strategies that involve continuous variables, such as adjusting pricing, capacity allocations, or departure times. Gaussian mutation adds a small random value drawn from a Gaussian (normal) distribution to each gene in the chromosome. This helps fine-tune strategies by making small, incremental changes, which is particularly useful in fine-grained optimization problems.

$$\text{mutate}_{\text{gaussian}}(x, \mu, \sigma) = \begin{cases} x + \mathcal{N}(0, \sigma) & \text{if } \mathbb{P} < \mu \\ x & \text{otherwise} \end{cases} \tag{5.35}$$

Here, $\mathcal{N}(0, \sigma)$ is a Gaussian random variable with mean 0 and standard deviation σ.

The impact of the mutation on the GT-MMT model can be summaries as follows:

- **Exploration of New Strategies:** Mutation enables the investigation of unexplored strategies, varying in aspects like route preferences, departure times, or logistical operations.
- **Genetic Diversity:** Introducing random changes through mutation sustains diversity within the population, a critical factor in preventing premature convergence and ensuring a comprehensive search of the solution space.

Step 3: Real-Time Dynamic Adjustments with PSO and Stacklberg

The algorithm integrates particle swarm optimization (PSO) and Stackelberg Game theory to find the near-optimal routes and strategies for multiple transportation modes. The process iteratively updates and evaluates the strategies using real-time data. This operation of the algorithm can be divided into five steps:

1. Initialization

A swarm is a collection of particles, each representing a potential solution. Let $Swarm = \{\mathbf{P}a_1, \mathbf{P}a_2, \ldots, \mathbf{P}a_{NP}\}$ be the swarm, where NP is the number of particles. Each particle $\mathbf{P}a_i$ represents a set of strategies. Each particle has a position $\mathbf{X}pso_i(t)$ (strategy configuration) and a velocity $\mathbf{v}_i(t)$ (the rate of change in the particle's position (strategy configuration) over time).

Algorithm 4 Real-Time Dynamic Adjustments with PSO and Stacklberg

Input: Swarm of particles, initial velocities, optimized initial strategies, real-time data

Output: Optimized particle positions and velocities (near-Optimal strategies)

Initialize PSO: Initialize a swarm of particles, each representing a different set of routes and schedules. Assign initial velocities to each particle.

for $t = 1$ ***to*** I_{max} **do**

  1. Velocity and Position Update: Update the velocity v_{pa_i} for each particle pa_i

$$v_{pa_i}^{t+1} = \omega v_{pa_i}^{t} + c_1 r_1 (pbest_{pa_i} - Xpso_{pa_i}^{t}) + c_2 r_2 ((gbest_{pa} - Xpso_{pa_i}^{t}) \tag{5.36}$$

  Update the position $Xpso_{pa_i}$ for each particle pa_i using the equation:

$$Xpso_{pa_i}^{t+1} = Xpso_{pa_i}^{t} + v_{pa_i}^{t+1} \tag{5.37}$$

  2. Fitness Evaluation: Evaluate the fitness of each particle based on the combined objectives and updated strategies by using the Stackelberg equilibrium with the PSO

$$Fitness(Xpso_{pa}) = z \cdot U_{p_i}(Xpso_{p_i}) + \beta \cdot U_{c_j}(Xpso_{c_j}) + U_{b_k}(Xpso_{b_k}) + U_{bk_n}(Xpso_{bk_n}) \tag{5.38}$$

  - Define the leader's objective function to minimize travel time and cost:

$$\text{Minimize } U_p = \alpha TT + \beta TF \tag{5.39}$$

  - Define the followers' objective functions to maximize the number of passengers in the ride-sharing and the bus and increase the usability of the bike:

$$\min_{A_L} U_L(A_L, A_F^*(A_L)) \tag{5.40}$$

$$A_F^*(A_L) = \arg\max_{A_F} U_F(A_L, A_F) \tag{5.41}$$

  3. Update Personal Best and Global Best:
  - Update the personal best position $pbest_{pa_i}$:

$$pbest_{pa_i}^{t+1} = \begin{cases} Xpso_{pa_i}^{t+1} & \text{if Fitness}(Xpso_{pa_i}^{t+1}) < \text{Fitness}(pbest_{pa_i}^{t}) \\ pbest_{pa_i}^{t} & \text{otherwise} \end{cases} \tag{5.42}$$

  - Update the global best position $gbest_{pa}$

$$gbest_{pa}^{t+1} = \begin{cases} Xpso_{pa_i}^{t+1} & \text{if Fitness}(Xpso_{pa_i}^{t+1}) < \text{Fitness}(gbest_{pa}^{t}) \\ gbest_{pa}^{t} & \text{otherwise} \end{cases} \tag{5.43}$$

  if $|Fitness(gbest_{pa}^{t+1}) - Fitness(gbest_{pa}^{t})| < \epsilon$ *for a predefined number of iterations* **then**

    Exit loop

return near-Optimal strategies for leaders and followers

A swarm of particles is initialized to begin the PSO process. Here, the initial positions of particles in the swarm are the output strategies of the GA operation (the previous step in the GT-MMT algorithm). So, $Xpso_i = CH$, where $Xpso_i$ is the initial strategy and CH is the near-optimal strategy for the GA operation. Initialize the velocity $\mathbf{v}_i(t)$ randomly. Each particle is assigned a personal best position, denoted as $\mathbf{pbest}_i = \mathbf{X}pso_i$. The global best position **gbest** is determined as the position of the particle that has achieved the highest fitness value among all particles in the swarm.

$$gbest = \arg \min_{pa \in \text{Swarm}} \big(\text{Fitness}(pbest^t_{pa_i})\big) \tag{5.44}$$

2. Velocity and Position Update

Adjust the velocity of each particle based on its personal best position and the global best position.

$$v^{t+1}_{pa_i} = \omega v^t_{pa_i} + c_1 r_1 (pbest_{pa_i} - Xpso^t_{pa_i}) + c_2 r_2 ((gbest_{pa} - Xpso^t_{pa_i}) \tag{5.45}$$

where w is the inertia weight, c_1 and c_2 are acceleration coefficients, and r_1 and r_2 are random numbers between 0 and 1, $pbest_p$ is the personal best position of particle p, and $gbest$ is the global best position.

Update the position of each particle based on its updated velocity.

$$Xpso^{t+1}_{pa_i} = Xpso^t_{pa_i} + v^{t+1}_{pa_i} \tag{5.46}$$

3. Fitness Evaluation

The fitness function combines multiple objectives: minimizing travel time and minimizing fare for the passenger, increasing the number of passengers in ride-sharing and buses, and ensuring bike availability. In the PSO process, the strategies are represented as particles Pa_i. Fitness is calculated for each particle, generally combining the utilities for passengers and transportation modes. So the equation 5.15 can be written as:

$$Fitness(Xpso_{pa}) = z \cdot U_{p_i}(Xpso_{p_i}) + \beta \cdot U_{c_j}(Xpso_{c_j}) + U_{b_k}(Xpso_{b_k}) + U_{bk_n}(Xpso_{bk_n}) \quad (5.47)$$

where $Fitness(Xpso_{pa})$ is the total fitness for all the players in the models, and $Xpso$ is the set of strategies. This fitness is evaluated using the Stackelberg equilibrium to determine near-optimal strategies for leaders and followers.

In the context of the GT-MMT algorithm, the Stackelberg equilibrium involves the interaction between the leader (passenger) and the followers (transportation modes: bus, ride-sharing, and bike-sharing). The leader makes decisions first, and the followers react to those decisions. This hierarchical decision-making process is modelled using the Stackelberg game theory.

The leader (passenger) aims to minimize their travel time and fare. The objective function of the leader is defined as:

$$\text{Minimize } U_{\text{leader}} = \alpha TT_{leader} + \beta TF_{leader} \quad (5.48)$$

where TT_{leader} is the total trip time, TF_{leader} is the total fare, and α and β are weight factors that balance the importance of travel time and fare.

The leader (passenger) selects a strategy $Xpso_{\text{leader}}$ that minimizes their objective function:

$$Xpso^*_{\text{leader}} = \arg \min_{U_{\text{leader}}} U_{\text{leader}}(Xpso_{\text{leader}}, Xpso_{\text{follower}}) \quad (5.49)$$

This involves choosing the optimal routes, departure times, and modes of transportation.

The leader solves the following optimization problem:

$$\min_{x_{\text{leader}}} (\alpha TT_{\text{total}}(x_{\text{leader}}, x_{\text{follower}}) + \beta TF_{\text{total}}(x_{\text{leader}}, x_{\text{follower}})) \quad (5.50)$$

The followers (transportation modes) aim to maximize their capacity utilization. The objective function for the followers is defined as:

$$\text{Maximize } U_{\text{follower}} = \gamma \left(\frac{\text{Passenger Count}}{\text{Capacity}} \right) + \delta \left(\frac{\text{Bike Usability}}{\text{Total Bike Capacity}} \right) \quad (5.51)$$

where Passenger Count is the number of passengers using the mode, Capacity is the total capacity of the mode, Bike Usability is the utilization of bike-sharing, and γ and δ are weight factors that balance the importance of passenger count and bike usability.

Given the leader's strategy A^*_{leader}, each follower (transportation mode) selects a strategy A_{follower} that maximizes their objective function:

$$Xpso^*_{\text{follower}} = \arg\max_{Xpso_{\text{follower}}} U_{\text{follower}}(Xpso^*_{\text{leader}}, Xpso_{\text{follower}}) \tag{5.52}$$

This involves adjusting the operational strategies such as route adjustments, departure time changes, and capacity allocation.

Each follower solves the following optimization problem:

$$\max_{Xpso_{\text{follower}}} \left(\gamma \frac{\text{Passenger Count}(Xpso^*_{\text{leader}}, Xpso_{\text{follower}})}{\text{Capacity}} + \delta \frac{\text{Bike Usability}(Xpso^*_{\text{leader}}, Xpso_{\text{follower}})}{\text{Total Bike Capacity}} \right) \tag{5.53}$$

The equilibrium is reached when the leader's and followers' strategies are mutually optimal. The leader's strategy x^*_{leader} minimizes their cost given the followers' strategies, and the followers' strategies x^*_{follower} maximize their utility given the leader's strategy.

Assume the leader chooses a route R_{leader} and departure time D_{leader}. The bus service, as a follower, responds by adjusting its route R_{bus} and departure time D_{bus} to attract more passengers and maximize capacity utilization. This interdependent decision-making continues iteratively until equilibrium is reached.

The iterative process continues until the strategies of the leader and followers converge to an equilibrium. Convergence is typically achieved when changes in the objective functions between iterations fall below a predefined threshold:

$$\left| U^{(t)}_{\text{leader}} - U^{(t-1)}_{\text{leader}} \right| < \epsilon \tag{5.54}$$

$$\left| U^{(t)}_{\text{follower}} - U^{(t-1)}_{\text{follower}} \right| < \epsilon \tag{5.55}$$

where ϵ is the convergence criterion.

In conclusion, the leader, represented by the passenger, selects an initial strategy A_L. Following this, the transportation modes, which act as followers, optimize their strategies A_F based on the leader's chosen strategy. Once the followers have optimized their strategies, the leader updates their strategy A_L in response to the adjustments made by the followers. This iterative process continues, with the leader and followers continuously updating their strategies until convergence is achieved. This iterative process ensures that both leaders and followers reach near-optimal strategies that satisfy the objectives of minimizing travel time and cost, increasing ridership, and ensuring bike availability.

4. Update Personal Best and Global Best

Track the best position each particle has achieved so far, denoted as $pbest_{pa_i}$. If the current fitness of particle pa_i is better than the fitness of $pbest^t_{pa_i}$, update $pbest^{t+1}_{pa_i}$:

$$pbest^{t+1}_{pa_i} = \begin{cases} Xpso^{t+1}_{pa_i} & \text{if Fitness}(Xpso^{t+1}_{pa_i}) < \text{Fitness}(pbest^t_{pa_i}) \\ pbest^t_{pa_i} & \text{otherwise} \end{cases} \tag{5.56}$$

Global Best Update: If the current position of any particle is better than the global best, update the global best.

$$gbest^{t+1}_{pa} = \begin{cases} Xpso^{t+1}_{pa_i} & \text{if Fitness}(Xpso^{t+1}_{pa_i}) < \text{Fitness}(gbest^t_{pa}) \\ gbest^t_{pa} & \text{otherwise} \end{cases} \tag{5.57}$$

In the GT-MMT algorithm, the stopping criterion can either reach the maximum number of iterations $I_{\max}$ or when the solutions have converged to a satisfactory level. The change in the best fitness value is below a certain threshold ϵ for a predefined number of consecutive iterations.

Let I be the current iteration number, $I_{\max}$ be the maximum number of iterations, and ϵ be the convergence threshold. The stopping criteria can be represented as follows:

$$\text{Stop if:} \quad I \geq I_{\max} \quad \text{or} \quad |Fitness(gbest^{t+1}_{pa}) - Fitness(gbest^t_{pa})| < \epsilon \tag{5.58}$$

where $Fitness(gbest^t_{pa})$ is the best fitness value at iteration t, and $|Fitness(gbest^{t+1}_{pa}) -$

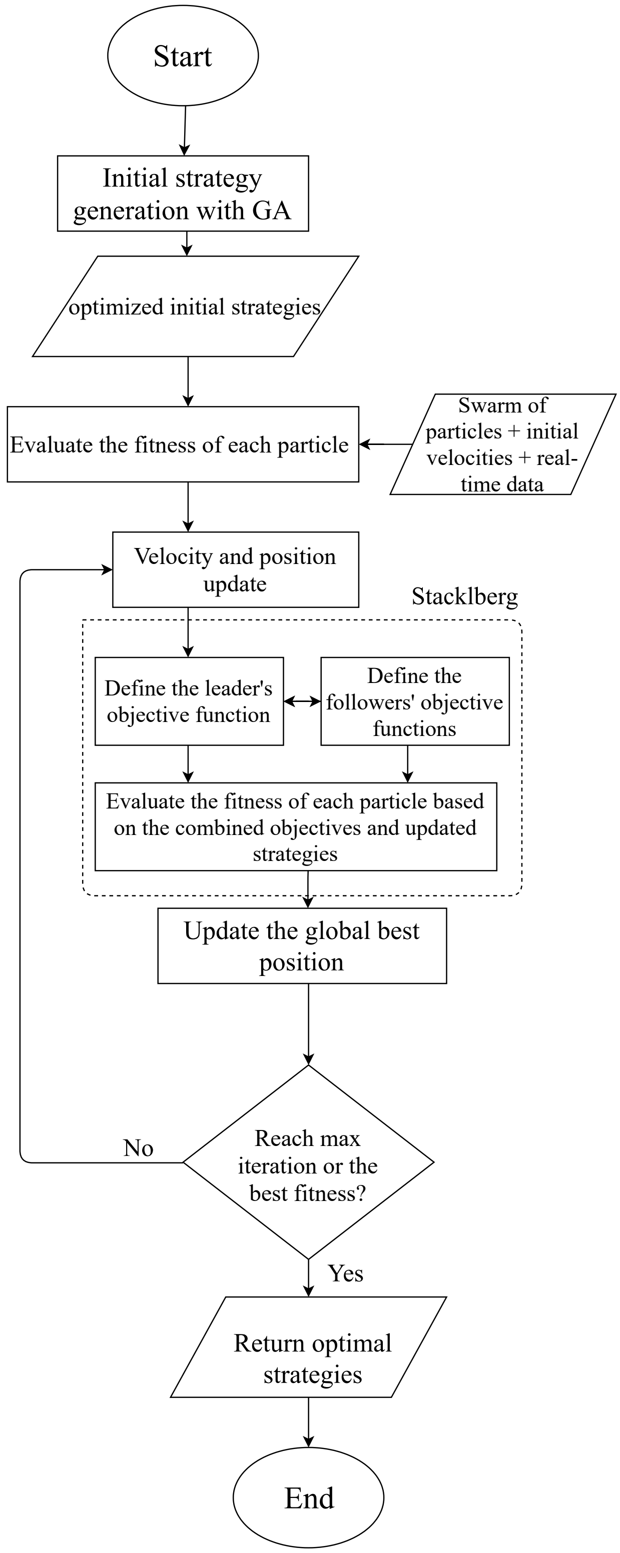

Figure 5.3: The overview diagram for the GT-MMT algorithm

$Fitness(gbest^t_{pa})|$ is the change in the best fitness value between iterations t and $t+1$.

The overview diagram for the GT-MMT algorithm is shown in figure 5.3.

5.5 Simulation and Results

Figure 3.2 presents a partial map of the City of Ottawa with three transportation forms (bus, car ride-sharing, and bicycle) to evaluate our GT-MMT model. Simulation setups are presented in Table 5.2. To obtain part of the Ottawa map, we use Simulation of Urban Mobility (SUMO), and to set the route, we use OMNET++. SUMO randomly sets up the mobility traffic scenario. This simulation measures the movement of car ride-sharing, public buses, and bicycles in the City of Ottawa from 12:00 a.m. until 11:59 p.m. For more detailed information on the simulation settings and parameters used in the GT-MMT model, please refer to the simulation section 3.6 in the 3 which provides comprehensive insights into the initialization processes, network configuration, transportation mode parameters, and the integration of real-time data to ensure accurate and realistic simulations.

5.5.1 Experiment Evaluations and Results Discussion

By using the Monte Carlo technique to find the average value after 10000 running times, our simulations are conducted for over 10000 passengers. This simulation generates the random movement of passengers, car ride-sharing, public buses, and bicycles in the City of Ottawa during 24 hours.

Table 5.2: Simulation parameters for the GT-MMT model

Vehicles types	Car ride-sharing, Public bus and bicycle
Simulation time	12:00 a.m.-11:59 p.m.
Time set length	1 minute
Number of cars ride-sharing	10
Number of Public buses	20
Number of Bicycle	30
Number of Passengers	10000
number of the public bus stations	10
number of the car ride-sharing stations	8
number of the bicycle stations	6
simulation period	12 a.m-11:59 p.m

1. Average Trip Time and Average Trip Price Variation

Figure 5.4 shows the average trip time and the average trip price variation for the GT-MMT model for the whole day. In this figure, the average trip price is constant and in the lowest value from midnight until 6:00, from 10:00 to 11:00, and from 14:00 until midnight. However, this value increases between 6:00 and 9:00 and also between 11:00 and 13:00. For the average trip time, the highest values are recorded from 4:00 to 6:00 and from 19:00 until 20:00. We note that the time is significantly decreased starting at 6:00 until it reaches the lowest value at 10:00. However, the time substantially increases after midday before it becomes stable between 16:00 and 18:00.

2. Travel Time and Fare Comparison Between GT-MMT and DMT Model

Figure 5.5 and Figure 5.6 show the comparison of the average trip time and the average trip fare between the multi-mode transportation system based on GT and another model of multi-mode transportation that does not use the GT approach.

From Figure 5.5, we note that the price is the same from midnight until 6:00 in the GT-MMT model and in the model without GT-MMT. After six o'clock, the price for GT-MMT starts to increase slightly until eight o'clock before decreasing. The price increases at 11 o'clock and then decreases beginning at 2. The value remains almost constant for the rest of the day at the lowest cost compared to the model without GT-MMT. In Table 5.3, the average trip price-saving rate between the GT-MMT model and model without GT-MMT is zero at the beginning of the day and 70 $ from 14:00 until midnight. However, the savings rate between the two models is almost constant and equal to 23.3.

Figure 5.6 represents the time trip variation value for the two models. We note that the average time value when the passengers use the GT-MMT and the model without GT-MMT is almost the same from midnight until 6:00. After 6:00, the time slightly drops for the GT-MMT model, and then it returns to a slight ascent in the middle of the day before becoming constant at four p.m. and continuing at the 20 min to the end of the day. As for the model without GT-MMT, the time increases slightly after six o'clock, increases significantly after one o'clock, and reduces and continues to rise to 4:00.

The time reduction is calculated using the equation:

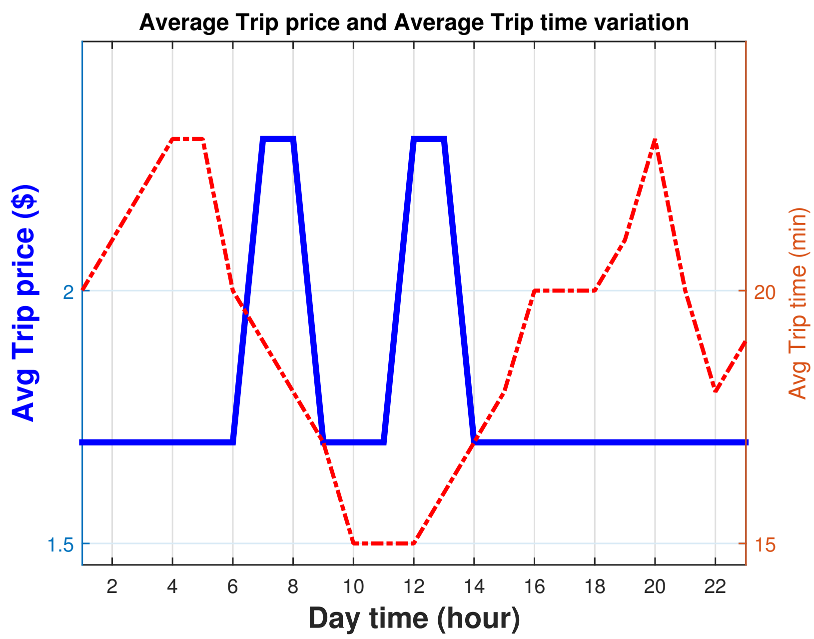

Figure 5.4: Average trip time and average trip price variation

$$\text{Time Reduction}(\%) = \left(\frac{\text{Initial Travel Time} - \text{Optimized Travel Time}}{\text{Initial Travel Time}} \right) \times 100$$

In Table 5.4, the savings rate of the average trip time between GT-MMT and the model without GT-MMT is zero from midnight until 6:00. After that, it starts to increase significantly until reaching 64.5 min at 16:00.

According to the Stackelberg equilibrium game theory model, every player is trying to achieve the utility function within the model that best meets optimization. The passengers want to reach their destination for the lowest price and in the least amount of time; cars and buses want to increase the number of passengers to maximum capacity, and bicycles reduce the availability. So:

1. In general, the number of passengers and the number of available vehicles is higher during the peak period between 7:00 and 9:00 and between 14:00 and 18:00. From the previous example, we can explain why time is at its lowest level as in Figure 5.4, where the abundance of vehicles and a large number of passengers with the freedom to choose the type or types of transportation lead to a reduction in the average trip time it takes for a passenger to reach his destination.

2. As shown in Figure 5.5, the total price in the two systems is almost the same at the

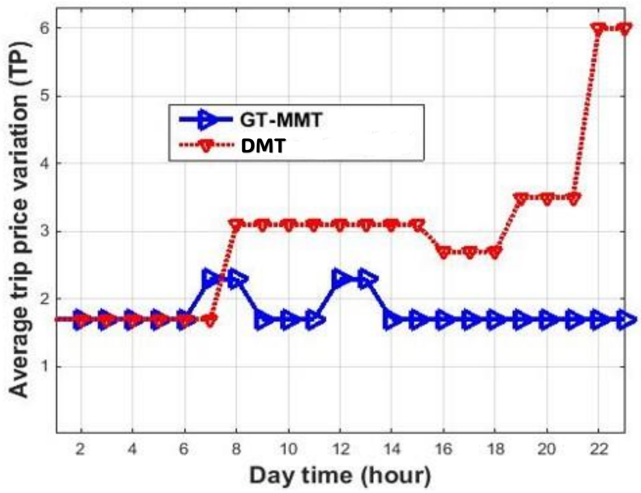

Figure 5.5: Average trip price variation

Table 5.3: Saving rate of average trip price variation with and without GT-MMT

Day time (hour)	**With GT-MMT ($)**	**DMT ($)**	**Saving rate (%)**
12:00 a.m - 6:00 a.m	1.8	1.8	**0**
6:01 a.m - 9:00 a.m	2.3	3	**23.3**
9:01 a.m - 2:00 p.m	2.3	3	**23.3**
2:01 p.m - 11:59 p.m	1.8	6	**70**

beginning of the simulation when the traffic of the city is not heavy. However, during the morning rush hour, while the average price in the multi-mode system that does not use the GT approach increases significantly, the average price in the GT-MMT increases slightly. We observe that the rise in price is low in the GT-MMT scheme but high in the system without GT-MMT. The savings rate for the total trip price when using GT-MMT is higher than the savings rate when the GT-MMT system is not used, as shown in Table 5.3.

3. From Figure 5.6, similar to the trip price value, the trip period time without the GT-MMT model rapidly and continuously increases. In contrast, the trip period time is stable and short in the GT-MMT model. More time is saved when moving from the origin to the

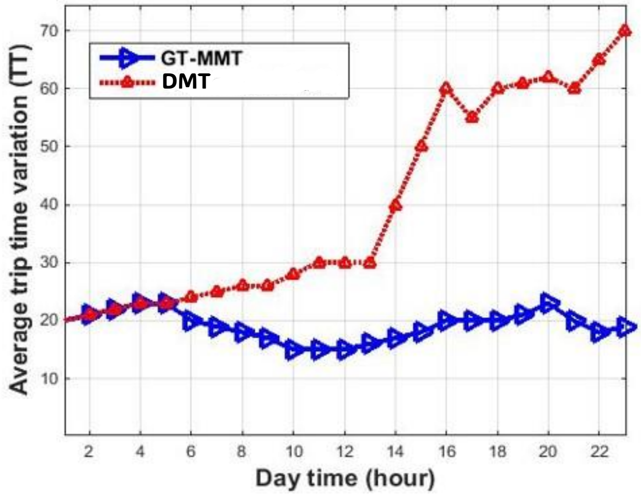

Figure 5.6: Average trip time variation

Table 5.4: Saving rate of average trip time variation with and without GT-MMT

Day time (hour)	**With GT-MMT** (min)	**DMT** (min)	**Saving rate** (%)
12:00 a.m - 6:00 a.m	22	22	**0**
6:01 a.m - 1:00 a.m	10.8	23	**53**
1:01 a.m - 4:00 p.m	10.9	50	**78.2**
4:01 p.m - 11:59 p.m	22	62	**64.5**

destination by using the GT-MMT scheme, as we can see in Table 5.4.

3. Some Metrics Comparison Between GT-MMT and Dijkstra's Algorithm

a. Travel Time Comparison

The travel time comparison table shows the initial travel time (Dijkstra's Algorithm) and the optimized travel time (GT-MMT) for each passenger, along with the percentage reduction in travel time. The time reduction is calculated using the equation:

Table 5.5: Comparison of initial travel time (using Dijkstra's Algorithm) and Optimized travel time (using GT-MMT)

Passenger ID	**Initial Travel Time [Dijkstra's Algorithm] (min)**	**Optimized Travel Time [GT-MMT](min)**	**Time Reduction (%)**
1	45	30	33.3
2	60	40	33.3
3	50	35	30.0
4	55	38	30.9
5	70	45	35.7

$$\text{Time Reduction}(\%) = \left(\frac{\text{Initial Travel Time} - \text{Optimized Travel Time}}{\text{Initial Travel Time}} \right) \times 100 \qquad (5.59)$$

From the table, we observe that the GT-MMT model significantly reduces travel times for all passengers, with reductions ranging from 30.0% to 35.7%. This demonstrates the effectiveness of the optimization techniques (GA, PSO, Stackelberg) in decreasing travel times and enhancing the efficiency of the transportation system.

b. Travel Cost Comparison

Table 5.6: Comparison of Initial and Optimized Costs

Passenger ID	**Initial Cost [using Dijkstra's Algorithm] ($)**	**Optimized Cost [using GT-MMT] ($)**	**Cost Reduction (%)**
1	12.00	8.50	29.2
2	15.00	10.00	33.3
3	10.00	7.00	30.0
4	13.00	9.00	30.8
5	20.00	12.50	37.5

The cost comparison table lists the initial and optimized travel costs for each passenger, as well as the percentage reduction in cost. The cost reduction is calculated using the equation:

$$\text{Cost Reduction}(\%) = \left(\frac{\text{Initial Cost} - \text{Optimized Cost}}{\text{Initial Cost}}\right) \times 100 \tag{5.60}$$

The table reveals that the GT-MMT model effectively reduces travel costs for all passengers, with reductions ranging from 29.2% to 37.5%. This indicates that the model optimizes travel costs, making the transportation system more economical for users.

c. Mode Usage Distribution Comparison

Table 5.7: Distribution of mode usage before and after optimization

Mode	**Initial Usage (%)**	**Optimized Usage (%)**	**Change (%)**
Car	60	40	-33.3
Bus	25	35	+40.0
Bicycle	15	25	+66.7

The mode usage distribution table shows the percentage of usage for each mode of transportation before (using Dijkstra's algorithm) and after (using GT-MMT) optimization, along with the percentage change. The change in usage is calculated using the equation:

$$\text{Change}(\%) = \left(\frac{\text{Optimized Usage} - \text{Initial Usage}}{\text{Initial Usage}}\right) \times 100 \tag{5.61}$$

From the table, we observe a significant shift in the usage of transportation modes. Car usage decreases by 33.3%, while bus and bicycle usage increases by 40.0% and 66.7%, respectively. This suggests that the GT-MMT model promotes the use of more sustainable transportation modes, reducing reliance on cars.

4. Optimization Iteration Performance

The optimization iteration performance table illustrates how the average travel time and average cost change over iterations. The convergence metric indicates how close the model is to reaching a near-optimal solution. The convergence metric $M(k)$ at iteration k is calculated as follows:

$$M(k) = \alpha \cdot \left(\frac{T_{\text{best}}}{T_{\text{avg}}(k)}\right) + \beta \cdot \left(\frac{C_{\text{best}}}{C_{\text{avg}}(k)}\right) \tag{5.62}$$

Table 5.8: Optimization Iteration Performance

Iteration	Average Travel Time (min)	Average Cost ($)	Convergence Metric
1	60.0	15.00	0.75
10	55.0	13.00	0.85
20	50.0	11.00	0.90
30	45.0	9.50	0.95
40	40.0	8.00	0.98

Where $T_{\text{avg}}(k)$ is the average travel time at iteration k, $C_{\text{avg}}(k)$ is the average cost at iteration k, T_{best} is the best (lowest) possible travel time, C_{best} is the best (lowest) possible cost, and α, β are weight factors that balance the importance of travel time and cost.

Initially, at iteration 1, the average travel time is 60.0 minutes, the average cost is $15.00, and the convergence metric is 0.75. As the iterations progress, there is a steady improvement in these metrics. By iteration 10, the average travel time decreases to 55.0 minutes, and the average cost reduces to $13.00, resulting in an improved convergence metric of 0.85. Further iterations show continuous improvement, with iteration 20 showing an average travel time of 50.0 minutes and an average cost of $11.00, with a convergence metric of 0.90.

The table illustrates the effectiveness of the optimization process over 40 iterations. The average travel time and average cost decrease steadily with each set of iterations, reflecting the algorithm's capability to find more efficient solutions. The convergence metric, which combines both travel time and cost into a single measure, increases progressively, indicating that the algorithm is moving closer to the near-optimal solution.

By iteration 40, the average travel time is significantly reduced to 40.0 minutes, and the average cost drops to $8.00, with the convergence metric reaching 0.98. This high convergence metric demonstrates that the GT-MMT model effectively optimizes transportation parameters, approaching a near-optimal solution. The consistent improvement in travel time and cost highlights the robustness and efficiency of the optimization techniques (GA, PSO, Stackelberg) employed in the GT-MMT model.

Table 5.9: Mode switch analysis

Passenger ID	Initial Mode	Final Mode	Switch Reason	Time Saved (min)	Cost Saved ($)
1	Car	Bus	Reduced congestion	15	3.50
2	Car	Bicycle	Cost reduction	20	5.00
3	Bus	Ride-share	Cost reduction	10	2.00
4	Car	Bus	Cost reduction	17	4.00
5	Car	Bicycle	Cost reduction	25	7.50

5. Mode Switch Analysis

To calculate the values in the mode switch analysis table, we need to consider several factors, including the initial and final modes, reasons for switching, time saved, and cost saved:

$$T_{\text{saved}} = T_{\text{initial}} - T_{\text{final}} \tag{5.63}$$

$$C_{\text{saved}} = C_{\text{initial}} - C_{\text{final}} \tag{5.64}$$

Where T_{initial} represents the travel time for the initial mode, T_{final} represents the travel time for the final mode, C_{initial} represents the cost for the initial mode, C_{final} represents the cost for the final mode, T_{saved} represents the time saved by switching modes, and C_{saved} represents the cost saved by switching modes.

From the table, we observe that the mode switches have led to significant savings in both travel time and cost for the passengers. Passenger 1 switched from car to bus due to reduced congestion, saving 15 minutes and \$3.50. Passenger 2 and Passenger 5 switched from car to bicycle for cost reduction, saving 20 minutes and \$5.00, and 25 minutes and \$7.50, respectively. Passenger 3 switched from bus to ride-share, and Passenger 4 switched from car to bus, both for cost reduction, saving 10 minutes and \$2.00, and 17 minutes and \$4.00, respectively.

The table illustrates the benefits of optimizing transportation modes by focusing on reducing travel time and fare costs. The primary reasons for switching modes were reduced congestion and cost reduction, leading to overall improvements in efficiency. The significant time and cost savings demonstrate the effectiveness of the GT-MMT model in enhancing the transportation system.

The mode switch analysis shows that optimizing the transportation system by focusing on reducing travel time and fare costs can lead to substantial benefits for passengers. The consistent reduction in both metrics across different mode switches highlights the GT-MMT model's capability to provide efficient and cost-effective transportation solutions. This optimization not only enhances passenger satisfaction but also contributes to a more efficient urban transportation network.

6. Hourly Variation in Trip Metrics

Table 5.10: Hourly variation in trip metrics

Time Interval	**Average Trip Price ($)**	**Average Trip Time (min)**	**Price Change (%)**	**Time Change (%)**
00:00 - 01:00	2.00	20	0	0
01:00 - 02:00	2.00	22	0	10
02:00 - 03:00	2.00	21	0	-4.8
03:00 - 04:00	2.00	23	0	9.5
04:00 - 05:00	2.00	25	0	8.7
05:00 - 06:00	2.00	24	0	-4.0
06:00 - 07:00	2.50	22	25	-8.3
07:00 - 08:00	3.00	20	20	-9.1
08:00 - 09:00	3.50	19	16.7	-5.0
09:00 - 10:00	3.00	17	-14.3	-10.5
10:00 - 11:00	2.00	15	-33.3	-11.8
11:00 - 12:00	2.50	18	25	20.0
12:00 - 13:00	3.00	22	20	22.2
13:00 - 14:00	2.50	20	-16.7	-9.1
14:00 - 15:00	2.00	18	-20.0	-10.0
15:00 - 16:00	2.00	17	0	-5.6
16:00 - 17:00	2.00	17	0	0
17:00 - 18:00	2.00	18	0	5.9
18:00 - 19:00	2.00	20	0	11.1
19:00 - 20:00	2.00	22	0	10.0
20:00 - 21:00	2.00	21	0	-4.5
21:00 - 22:00	2.00	20	0	-4.8
22:00 - 23:00	2.00	19	0	-5.0
23:00 - 00:00	2.00	20	0	5.3

a. Average Trip Price and Time Calculation

Average Trip Price

$$\text{Average Trip Price} = \frac{\sum_{i=1}^{n} \text{Trip Price}_i}{n} \tag{5.65}$$

Average Trip Time

$$\text{Average Trip Time} = \frac{\sum_{i=1}^{n} \text{Trip Time}_i}{n} \tag{5.66}$$

b. Percentage Change Calculation

Price Change (%)

$$\text{Price Change (\%)} = \left(\frac{\text{Current Price} - \text{Previous Price}}{\text{Previous Price}} \right) \times 100 \tag{5.67}$$

Time Change (%)

$$\text{Time Change (\%)} = \left(\frac{\text{Current Time} - \text{Previous Time}}{\text{Previous Time}} \right) \times 100 \tag{5.68}$$

Explanation: From midnight to 6 AM, the trip price remains constant at $2.00 while trip times fluctuate slightly, suggesting low demand and stable pricing. For instance, from midnight to 1 AM, the trip time is 20 minutes, but it increases to 22 minutes between 1 AM and 2 AM. This slight fluctuation in trip times, with no change in trip price, indicates a stable yet low-demand period. Between 6 AM and 9 AM, trip prices increase significantly, reflecting the morning rush hour. Trip prices rise from $2.50 to $3.50, and trip times decrease from 22 minutes to 19 minutes. This pattern suggests optimized routing and increased service availability during peak hours to manage higher demand efficiently. The period from 9 AM to 10 AM shows a decrease in both trip prices and trip times, indicating reduced demand post-rush hour. Trip prices drop from $3.50 to $3.00, and trip times decrease from 19 minutes to 17 minutes. This reduction highlights the system's ability to adapt quickly to changing demand. From 10 AM to 11 AM, prices drop to their lowest at $2.00, and trip times continue to decrease to 15 minutes, reflecting near-optimal conditions with lower demand. This indicates a period of high efficiency in transportation services. During the midday period (11 AM to 1 PM), prices and trip times increase again, indicating a midday demand peak. Trip prices rise to $3.00, and trip times increase to 22 minutes, showing how the system responds to another peak in demand during lunch hours. The afternoon and evening periods (1 PM to 8 PM) show stable or slightly increasing trip prices with fluctuating trip times due to varied demand and traffic conditions. For example, from 2 PM to 3 PM, the trip price is $2.00 with a trip time of 18 minutes, while from 7 PM to 8 PM, the trip price remains $2.00, but trip times increase to 22 minutes, reflecting evening congestion.

Conclusion: The GT-MMT model dynamically adjusts trip prices and optimizes trip times based on demand fluctuations throughout the day. This results in efficient utilization of transportation resources, reducing congestion during peak hours and maintaining stable service levels during off-peak periods. This dynamic approach ensures a balance between supply and demand, leading to an overall improvement in urban mobility and user satisfaction.

7. Peak and Off-Peak Periods Comparison

Table 5.11: Peak and Off-Peak Periods Comparison

Period	**Average Trip Price ($)**	**Average Trip Time (min)**	**Price Difference ($)**	**Time Difference (min)**
Midnight-6 AM	2.00	24	-1.00	+6
6 AM-9 AM	3.00	20	+1.00	-4
9 AM-11 AM	2.50	16	-0.50	-4
11 AM-1 PM	2.75	20	+0.25	+4
1 PM-2 PM	2.50	19	-0.25	-1
2 PM-6 PM	2.00	18	-0.50	-1
6 PM-8 PM	2.00	21	0	+3
8 PM-Midnight	2.00	20	0	-1

a. Price Difference Calculation

Price Difference

$$\text{Price Difference} = \text{Peak Price} - \text{Off-Peak Price} \tag{5.69}$$

b. Time Difference Calculation

Time Difference

$$\text{Time Difference} = \text{Peak Time} - \text{Off-Peak Time} \tag{5.70}$$

Explanation: During the midnight to 6 AM period, the average trip price is \$2.00 with an average trip time of 24 minutes, reflecting low demand. This period shows stable pricing and higher trip times due to fewer available services and lower traffic volumes. The morning peak period from 6 AM to 9 AM has an increased average trip price of \$3.00 and a decreased trip time of 20 minutes. This indicates high demand and optimized routing to manage the increased number of passengers efficiently. The higher prices during this period help manage demand and ensure the availability of services. From 9 AM to 11 AM, the average trip price decreases to \$2.50, and the trip time decreases to 16 minutes. This reduction reflects the post-rush hour period where demand decreases, allowing for lower prices and improved trip times. During the midday peak from 11 AM to 1 PM, the average trip price increases to \$2.75, and the trip time increases to 20 minutes. This period indicates a resurgence in demand, likely due to lunch hours, leading to increased prices and longer trip times. The early afternoon period from 1 PM to 2 PM shows a slight decrease in both trip price (\$2.50) and trip time (19 minutes), reflecting reduced demand. This period benefits from slightly lower prices and improved trip times as the midday rush subsides. The off-peak afternoon period from 2 PM to 6 PM maintains a low average trip price of \$2.00 and a decreased trip time of 18 minutes. This period benefits from stable pricing and efficient trip times due to lower demand. The evening peak period from 6 PM to 8 PM shows a stable average trip price of \$2.00 and an increased trip time of 21 minutes due to evening congestion. Despite stable prices, the trip times reflect the impact of increased traffic during this period. The late evening period from 8 PM to midnight maintains a stable trip price of \$2.00 and a trip time of 20 minutes, reflecting stable demand and efficient service levels during lower traffic periods.

Conclusion: The comparison between peak and off-peak periods highlights the GT-MMT model's effectiveness in adjusting trip prices and optimizing trip times based on demand.

This dynamic pricing and routing strategy ensures efficient transportation during peak periods and maintains stable service levels during off-peak times, benefiting both the transportation system and urban areas. The model helps manage congestion, improve trip times, and maintain affordable prices, contributing to better urban mobility and user satisfaction.

8. Demand Response Analysis

Table 5.12: Demand response analysis

Time Interval	Demand Level	Average Trip Price ($)	Price Elasticity	Average Trip Time (min)	Time Elasticity
00:00 - 06:00	Low	2.00	0.00	24	0.00
06:00 - 09:00	High	3.00	0.50	20	-0.17
09:00 - 11:00	Moderate	2.50	-0.17	16	-0.20
11:00 - 13:00	High	2.75	0.10	20	0.20
13:00 - 14:00	Moderate	2.50	-0.09	19	-0.05
14:00 - 18:00	Low	2.00	-0.20	18	-0.05
18:00 - 20:00	High	2.00	0.00	21	0.08
20:00 - 00:00	Low	2.00	0.00	20	-0.05

a. Price Elasticity Calculation

Price Elasticity

$$\text{Price Elasticity} = \frac{\text{\% Change in Price}}{\text{\% Change in Demand}} \tag{5.71}$$

b. Time Elasticity Calculation

Time Elasticity

$$\text{Time Elasticity} = \frac{\text{\% Change in Time}}{\text{\% Change in Demand}} \tag{5.72}$$

Explanation: From midnight to 6 AM, demand is low with a stable average trip price of \$2.00 and trip time of 24 minutes. The price and time elasticities are both 0.00, indicating no change in prices or times in response to demand changes. This period shows stable pricing and higher trip times due to fewer available services and lower traffic volumes. The morning peak period from 6 AM to 9 AM has high demand with an average trip price of \$3.00. The price elasticity is 0.50, indicating that a 10% increase in demand results in a 5% increase in trip prices. The time elasticity is -0.17, indicating that a 10% increase in demand results in a 1.7% decrease in trip time due to optimized routing and increased service availability. From 9 AM to 11 AM, the demand is moderate with a trip price of \$2.50. The price elasticity is -0.17, indicating a decrease in price with reduced demand, and the time elasticity is -0.20, showing a significant decrease in trip time. This period reflects the post-rush hour with improved efficiency and reduced congestion. During the midday peak from 11 AM to 1 PM, demand is high with a trip price of \$2.75. The price elasticity is 0.10, showing a slight increase in price with increased demand, and the time elasticity is 0.20, indicating an increase in trip time due to midday congestion. The early afternoon period from 1 PM to 2 PM has moderate demand with a trip price of \$2.50. The price elasticity is -0.09, indicating a slight decrease in price with reduced demand, and the time elasticity is -0.05, showing a small decrease in trip time. The off-peak afternoon period from 2 PM to 6 PM has low demand with a trip price of \$2.00. The price elasticity is -0.20, indicating a decrease in price with reduced demand, and the time elasticity is -0.05, showing a slight decrease in trip time. The evening peak period from 6 PM to 8 PM has high demand with a trip price of \$2.00. The price elasticity is 0.00, indicating no change in price with increased demand, and the time elasticity is 0.08, showing a slight increase in trip time due to evening congestion. The late evening period from 8 PM to midnight has low demand, with a stable trip price of \$2.00 and trip time of 20 minutes. The price and time elasticities are both 0.00, indicating no change in prices or times in response to demand changes.

Conclusion: The demand response analysis demonstrates the GT-MMT model's ability to dynamically adjust prices and optimize trip times based on varying demand levels. The price and time elasticities indicate how sensitive trip prices and times are to changes in demand, reflecting the model's efficiency in managing transportation resources. This dynamic approach benefits urban areas by reducing congestion during peak periods and ensuring stable service levels during off-peak times. The GT-MMT model's flexibility in responding to demand changes helps maintain a balance between supply and demand, improving urban mobility and user satisfaction.

9. Performance Metrics: Convergence Rate

Table 5.13: Convergence rate for different population sizes

Population Size (N)	**Iterations to Converge**	**Observation**
20	50	Fast convergence but poor exploration of solution space
50	80	Moderate convergence with good exploration
100	120	Slow convergence with very thorough exploration
200	200	Very slow convergence, high computational cost

The iterations to converge are determined by monitoring the fitness value of the population over successive generations. Convergence is assumed when the change in the average fitness value between generations falls below a predefined threshold ϵ. Mathematically, if the average fitness value $\bar{Fitness}_t$ at generation t satisfies $|\bar{Fitness}_t - \bar{Fitness}_{t-1}| < \epsilon$ for a predefined number of consecutive generations, the algorithm is considered to have converged.

Observation

- **N = 20**: The algorithm converged quickly in 50 iterations, indicating that it reached a solution relatively fast. However, the quick convergence suggests that the algorithm might not have explored the solution space thoroughly, leading to near-optimal solutions. The small population size likely caused a lack of genetic diversity, which can result in premature convergence.

- **N = 50**: The algorithm required 80 iterations to converge, showing a balance between exploration and exploitation. The moderate population size provided enough genetic diversity to explore the solution space effectively while maintaining a reasonable convergence rate. This size allowed the algorithm to find better solutions than the smaller population without excessive computational cost.

- **N = 100**: Convergence took 120 iterations, which is slower compared to smaller populations. The larger population size allowed for a thorough exploration of the solution space, resulting in high-quality solutions. However, this came at the cost of slower convergence and higher computational demand.

- **N = 200**: The algorithm took 200 iterations to converge, the slowest among all tested sizes. The very large population ensured comprehensive exploration of the solution space but at a high computational cost. The slow convergence indicates that while the solution quality might be high, the time and resources required are substantial.

Conclusion

For the GT-MMT model, a population size of 50 provided the best balance between convergence rate and solution quality. Larger populations (N = 100, 200) offered thorough exploration, leading to potentially higher quality solutions but at a high computational cost. Smaller populations (N = 20) converged quickly but did not explore the solution space adequately, resulting in near-optimal solutions.

10. Performance Metrics: Solution Quality

Table 5.14: Solution quality for different population sizes

Population Size (N)	**Average Fitness Value**	**Observation**
20	0.65	Low solution quality due to poor exploration
50	0.85	Good solution quality with balanced exploration
100	0.90	High solution quality with thorough exploration
200	0.92	Very high solution quality but at a high computational cost

The average fitness value $\bar{F}$ of a population at generation t is calculated as the mean of the fitness values F_i of all individuals i in the population:

$$\bar{F} = \frac{1}{N} \sum_{i=1}^{N} F_i \tag{5.73}$$

where N is the population size and F_i is the fitness value of individual i.

Observation

- **N = 20**: The average fitness value was 0.65, indicating low solution quality due to insufficient exploration of the solution space. The small population size led to insufficient exploration of the solution space, resulting in a higher likelihood of the algorithm getting trapped in local optima. The lack of diversity in the population prevented the algorithm from finding better solutions.

- **N = 50**: The average fitness value improved to 0.85. This population size provided a good balance between exploration and exploitation, allowing the algorithm to find higher quality solutions. The moderate size ensured enough diversity to avoid premature convergence while maintaining computational efficiency.

- **N = 100**: The average fitness value was 0.90, showing high solution quality. The larger population size facilitated a thorough exploration of the solution space, enabling the algorithm to find better solutions. However, this came at the cost of slower convergence and higher computational demand.

- **N = 200**: The highest average fitness value of 0.92 was achieved with a population size of 200. While the solution quality was very high, the computational cost was significantly higher due to the large number of individuals and the slow convergence rate. This size ensured exhaustive exploration, but the practical feasibility depends on available computational resources.

For the GT-MMT model, a population size of 50 balanced solution quality and computational efficiency. Larger population sizes (N = 100, 200) resulted in higher solution quality due to a more thorough exploration of the solution space but required significantly more computational resources. Smaller populations (N = 20) produced lower-quality solutions because they did not adequately explore the solution space.

11. Computational Time of the GT-MMT Model

The computational time of the GT-MMT model is influenced by several factors, including the number of generations, population size, and complexity of the problem. The following table presents the computational time for different configurations of these parameters, calculated using the equation:

$$T_{total} = G \times (N \times T_f + N \times T_c) \tag{5.74}$$

Where T_{total} is the total computational time, G is the number of generations, N is the population size, T_f is the average time to evaluate the fitness of a single chromosome (assumed to be 0.1 seconds), and T_c is the average time for crossover and mutation per chromosome (assumed to be 0.05 seconds).

Table 5.15: Computational time for different configurations of the GT-MMT model

Population Size	**Number of Generations**	**Computational Time (seconds)**	**Observation**
20	100	$100 \times (20 \times 0.1 + 20 \times 0.05) = 300$	Quick execution, but may not explore solution space thoroughly.
20	200	$200 \times (20 \times 0.1 + 20 \times 0.05) = 600$	Slight increase in time, better exploration, but may still be insufficient.
50	100	$100 \times (50 \times 0.1 + 50 \times 0.05) = 750$	Moderate execution time, good balance between speed and exploration.
50	200	$200 \times (50 \times 0.1 + 50 \times 0.05) = 1500$	Longer execution, better exploration and solution quality.
100	100	$100 \times (100 \times 0.1 + 100 \times 0.05) = 1500$	Increased time due to larger population, thorough exploration.
100	200	$200 \times (100 \times 0.1 + 100 \times 0.05) = 3000$	Very long execution time, thorough exploration and high solution quality.

For a small population size of 20 and 100 generations, the computational time is 300 seconds. This configuration results in quick execution, but it may not explore the solution space thoroughly. When the population size is 20, and the number of generations is increased to 200, the computational time is 600 seconds, slightly increasing the time and offering better exploration, but it may still be insufficient. For a population size of 50 and 100 generations, the computational time is 750 seconds, providing a moderate execution time and a good balance between speed and exploration. Increasing the number of generations to 200 with a population size of 50 results in a computational time of 1500 seconds, allowing for better exploration and higher solution quality. For a larger population size of 100 and 100 generations, the computational time is 1500 seconds, where the increased time is due to the larger population, ensuring thorough exploration. Finally, for a population size of 100 and 200 generations, the computational time is 3000 seconds, resulting in a very long execution time and providing thorough exploration and potentially the highest solution quality.

In conclusion, the computational time of the GT-MMT model increases with both the population size and the number of generations. A moderate population size (e.g., 50) and a reasonable number of generations (e.g., 100-200) provide a good balance between computational efficiency and solution quality. Larger populations and more generations lead to better exploration and higher solution quality but at the cost of significantly increased computational time.

Chapter 6 Decentralised Game-Theoretic (DGT) Model

6.1 Overview

The Decentralized Game Theory (DGT) model is a community-based approach to optimizing urban transportation systems, Unlike the centralized GT-MMT model, which relies on a central control unit, the DGT model decentralizes decision-making, allowing individual participants to exercise autonomous control. The DGT model is designed to enhance the efficiency and responsiveness of the transportation network by empowering passengers and transportation modes (buses, ride-sharing services, and bicycles) to make decisions autonomously based on local information and real-time data. The scheme dynamically adjusts resources (transportation modes) based on real-time demand, ensuring that transportation services are efficiently utilized.

Unlike the GT-MMT model, where passengers act as leaders and transportation modes as followers in a hierarchical Stackelberg game, the DGT model employs a distributed version of Stackelberg game theory. In this approach, multiple local Stackelberg games occur simultaneously across the network, with different players taking on the roles of leaders and followers depending on the context. For instance, passengers may act as leaders when choosing modes of transport, while transportation modes adjust their operations as followers in response to passenger decisions. This decentralized structure allows for greater flexibility and adaptability, particularly in response to dynamic and localized changes in demand, traffic conditions, and resource availability.

A key distinction between the DGT and GT-MMT models is the use of fog computing architecture in the DGT model. Instead of relying on a centralized control unit to process data and make decisions, the DGT model leverages localized data processing through fog computing. This allows for real-time decision-making at the edge of the network, closer to the sources of data, thereby reducing latency and enhancing the system's ability to respond to immediate changes in the transportation environment.

The DGT model also employs advanced optimization algorithms, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Stackelberg game theory, similar to the GT-

MMT model. However, in the DGT model, these algorithms are applied in a decentralized manner, allowing individual transportation modes and passengers to optimize their strategies based on local information. The absence of a central authority means that each participant in the system must make decisions that balance their own objectives with the overall efficiency of the transportation network. As the GT-MMT model, each player has specific objectives. The riders aim to reach their destination within a maximum time limit at the minimum price. Meanwhile, the public buses and the ride-sharing services seek to maximize the number of passengers, and the bike-sharing services aim to increase their availability.

6.2 Main components of the DGT

The DGT model focuses on optimizing transportation within specific communities or neighbourhoods. Each community operates somewhat independently, with local decision-making tailored to its unique needs. However, the overall system remains interconnected, allowing for broader optimization when necessary. In each community, the transportation system seeks to reach a localized equilibrium where the strategies of passengers and transportation modes are balanced, ensuring that the transportation network operates efficiently within that specific area.

The Decentralized Game Theory (DGT) model, designed for community-based urban transportation optimization, consists of several key components that work together to manage and optimize the transportation system. These components ensure that the system is efficient, responsive, and adaptable to the needs of the community.

First: Participants (Players)

In the context of the DGT (Decentralized Game Theory) model, passengers may not necessarily be "leaders" in the traditional Stackelberg sense, as the DGT model emphasizes a more decentralized and community-based approach. Unlike the GT-MMT model, where passengers are explicitly modelled as leaders in a hierarchical Stackelberg game, the DGT scheme allows for a more distributed decision-making process.

1. **Passengers:** These are the individuals using the transportation system. In the DGT model, passengers actively participate by making decisions about their travel modes, routes,

and schedules. They act as leaders in localized Stackelberg games, influencing the operations of transportation modes based on their preferences and needs.

2. **Transportation Modes:** This includes buses, ride-sharing services, and bicycles. Each mode can act as a follower in the Stackelberg games, adjusting its operations (e.g., pricing, routing, and scheduling) in response to passenger decisions.

Second: Community Manager (CM)

The Community Manager (CM) plays a central role in the DGT model, overseeing the transportation service and ensuring security and privacy within the system. The CM manages several key components:

1. **Fog Computing Architecture:** Unlike a centralized system, the DGT model uses fog computing to enable local data processing and decision-making. Fog nodes are distributed throughout the network, close to where the data is generated. They process real-time data from sensors, GPS, and mobile applications to make immediate decisions that affect local transportation operations, such as rerouting buses, adjusting ride-sharing availability, or redistributing bicycles.

   Integrating fog nodes into a transportation network, particularly in the context of decentralized game theory (DGT) models, involves several steps. Fog nodes are essentially mini data centres located closer to the edge of the network, providing computational resources, storage, and services nearer to where the data is generated. This proximity reduces latency and bandwidth usage, making real-time processing and decision-making more efficient. The steps to integrate the fog node are as follows:

1. Identify Strategic Locations for Fog Nodes

- **Data Traffic Analysis**: Analyze data traffic patterns to identify high-traffic areas where data processing demand is high.
- **Proximity to Data Sources**: Place fog nodes close to data sources such as major transportation hubs, intersections, and areas with high passenger density.

2. **Define the Architecture**

- **Hierarchical Structure**: Establish a hierarchical structure where fog nodes operate between the central cloud and edge devices (sensors, vehicles, etc.).
- **Communication Protocols**: Define communication protocols for data exchange between edge devices, fog nodes, and the central cloud.

3. **Deployment of Fog Nodes**

- **Hardware and Software**: Select appropriate hardware and software solutions for fog nodes that support the required computational tasks and data storage needs.
- **Network Connectivity**: Ensure robust network connectivity between fog nodes and other network components to facilitate seamless data transmission.

4. **Data Processing and Management**

- **Local Data Processing**: Implement algorithms for local data processing at fog nodes to handle tasks such as real-time traffic monitoring, routing decisions, and passenger demand analysis.
- **Data Aggregation**: Aggregate data from multiple edge devices to derive insights and make informed decisions.
- **Distributed Database**: Use distributed databases to manage data across multiple fog nodes, ensuring data consistency and availability.

5. **Optimization and Decision-Making**

- **Real-Time Optimization**: Utilize real-time optimization algorithms such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) at fog nodes to dynamically adjust routes, schedules, and resource allocations.
- **Game Theory Integration**: Apply Stackelberg game theory principles where fog nodes act as intermediaries, optimizing follower strategies based on leader (passenger) decisions.

6. Security and Privacy

- **Data Encryption**: Implement data encryption techniques to ensure secure data transmission and storage.
- **Access Control**: Define access control policies to restrict data access to authorized entities only.

7. Monitoring and Maintenance

- **Performance Monitoring**: Continuously monitor the performance of fog nodes to identify and address any issues promptly.
- **Regular Maintenance**: Schedule regular maintenance to ensure the hardware and software components of fog nodes are functioning near-optimally.

In the context of the Decentralized Game Theory (DGT) model, fog nodes play a crucial role in enhancing the community-based transportation system by providing localized, real-time data processing and decision-making capabilities. Here's how fog nodes fit into the DGT model:

1. Local Optimization

Each fog node processes data locally and optimizes transportation strategies (e.g., routing, scheduling) for its specific area. This decentralization reduces the burden on a central server and allows for quicker, more responsive decision-making.

2. Community-Based Decisions

Fog nodes facilitate community-based decisions by collecting and processing data from local transportation modes and passengers, enabling a more tailored approach to meet the community's needs.

3. Real-Time Adaptation

By leveraging PSO and GA, fog nodes can continuously adapt to changing conditions, such as traffic patterns and passenger demands, ensuring near-optimal transportation network performance.

For example, we have this scenario:

- **Urban Area with High Traffic:** In an urban area with high traffic, fog nodes are placed at major intersections and transportation hubs. These fog nodes collect real-time data from sensors on buses, ride-sharing vehicles, and bicycles.
- **Data Processing:** The fog nodes process this data to analyze traffic conditions, predict passenger demand, and optimize routes and schedules using PSO and GA.
- **Decision-Making:** Based on the processed data and optimization algorithms, the fog nodes make real-time decisions to adjust traffic signals, reroute vehicles, and update schedules, ensuring smooth and efficient transportation for passengers.
- **Game Theory Application:** In a DGT model, the fog nodes facilitate the implementation of Stackelberg game theory by allowing passengers (leaders) to make travel decisions, which are then optimized by the transportation modes (followers) based on local data processed at the fog nodes.

In conclusion, Integrating fog nodes into a decentralized transportation network involves careful planning and execution. By strategically placing fog nodes, defining a robust architecture, and leveraging advanced optimization algorithms, the transportation system can be enhanced in efficiency, responsiveness, and scalability. Fog nodes play a pivotal role in enabling real-time, community-based decision-making, ultimately improving the overall travel experience for passengers.

2. **Decentralized Application (DApp)** The CM is responsible for managing the DApp, which facilitates the interaction between the passengers and the transportation modes. The DApp handles tasks such as data collection, processing, and providing real-time updates to passengers.

3. **Local Stackelberg Games** Localized Stackelberg games are played within each fog node, where passengers and transportation modes interact. These games help optimize the transportation system's local segment, considering that area's specific needs and conditions. More details are in the subsection.

Third: Data Collection and Sensing Infrastructure

1. **Sensors:** Deployed throughout the transportation network, sensors collect real-time data on traffic conditions, vehicle speeds, environmental factors, and other relevant metrics. This data is crucial for making informed decisions at the local level.

2. **Second: GPS Device:** In vehicles, GPS devices provide accurate real-time location data, essential for tracking vehicle movements, optimizing routes, and adjusting schedules.

3. **Third: Mobile Applications:** Passengers interact with the system through mobile apps, where they enter travel requests, receive real-time updates, and provide feedback. The data from these interactions is fed into the system, helping to refine predictions and optimize resource allocation.

Fourth: Optimization Algorithms

1. **Genetic Algorithms (GA):** GA is used to explore a wide range of potential solutions for optimizing transportation strategies. It helps in evolving strategies over successive iterations to improve performance.

2. **Particle Swarm Optimization (PSO):** PSO is employed to fine-tune the solutions found by GA. It iteratively improves strategies by simulating the behaviour of particles (representing potential solutions) in search of a near-optimal strategy.

3. **Decentralized Stackelberg Game Theory:** In the DGT model, Stackelberg game theory is applied in a decentralized manner, allowing multiple local Stackelberg games to occur simultaneously across the network. This approach helps balance the interests of passengers and transportation modes by ensuring that their strategies are mutually near-optimal. The DGT model incorporates a feedback loop where the outcomes of decisions are

continuously monitored, and the system learns and adapts over time. This helps improve the accuracy of predictions, optimize strategies, and enhance overall system performance.

In the community-based Decentralized Game Theory (DGT) model, the concept of Stackelberg game theory is adapted to fit the decentralized structure of the system. Unlike the traditional centralized Stackelberg game, where there is a clear leader (e.g., passengers) and followers (e.g., transportation modes), the DGT model applies a more distributed and localized version of Stackelberg game theory.

The **Multiple Concurrent Stackelberg Games** is the best method that can be used to apply in the DGT model. This approach aligns perfectly with the decentralized nature of the DGT model, where decision-making is distributed across various localized interactions rather than being centralized. In a community-based transportation system, different areas or neighbourhoods might have unique transportation needs and behaviours, leading to numerous localized Stackelberg games happening simultaneously. Each game involves passengers (leaders) and transportation modes (followers), making decisions based on local conditions. The **Multiple Concurrent Stackelberg Games** has the best benefit over the other method, which we can conclude as follows:

- **Decentralized Decision-Making**: The DGT model thrives on the ability to make decisions based on local data and conditions. Multiple concurrent Stackelberg games allow for this by letting each community or segment of the transportation network operate its own game.
- **Flexibility and Responsiveness**: This approach enhances the system's flexibility, as it allows different parts of the transportation network to adapt to changes independently. If one neighbourhood experiences a sudden increase in demand, the local transportation modes can adjust without needing to coordinate with a central authority.
- **Scalability**: By breaking down the transportation network into smaller, manageable Stackelberg games, the DGT model can scale more effectively. This is particularly useful in large urban areas where a single centralized game would be too complex to manage.
- **Improved Efficiency**: Each localized Stackelberg game optimizes the transportation strategies for its specific context, leading to overall improvements in efficiency and user satisfaction across the network.

- **Real-World Applicability**: Urban transportation systems naturally operate in a decentralized manner, with different neighbourhoods and regions having distinct transportation patterns. Multiple concurrent Stackelberg games reflect this reality and allow the DGT model to be more practical and applicable in real-world scenarios.

In the GT-MMT model, the transportation system operates under a centralized framework where passengers are typically seen as the leaders in a hierarchical Stackelberg game. In this setup, passengers make decisions about their travel routes, modes of transportation (such as buses, ride-sharing, or bicycles), and other preferences. The transportation modes, acting as followers, adjust their operations (like scheduling, pricing, and routing) based on the decisions made by the passengers. This centralized approach ensures that the strategies of the transportation modes are aligned with the passengers' preferences, optimizing the system's overall efficiency.

However, in the DGT (Decentralized Game Theory) model, the approach to decision-making is fundamentally different. Instead of having a single, overarching Stackelberg game where passengers are always the leaders and transportation modes are always the followers, the DGT model uses a distributed version of Stackelberg game theory. This means that multiple, smaller Stackelberg games occur simultaneously across the transportation network, each one localized to specific areas or communities. Here are two examples to illustrate the concept:

Example 1: Localized Leadership in a Neighborhood

Imagine a neighborhood with a high demand for bike-sharing services in the morning due to a large number of residents commuting to work or school. In this scenario, the passengers in this neighbourhood act as leaders in the local Stackelberg game. They choose to use bike-sharing services based on factors like availability, cost, and convenience. The bike-sharing system, acting as the follower, responds by adjusting its operations—perhaps increasing the number of bikes available, modifying pricing, or altering the placement of bikes to meet the high demand.

In another part of the city, where the demand for ride-sharing is higher during the evening due to nightlife, the roles may reverse. Here, the ride-sharing service might take the lead by offering promotions or adjusting availability based on anticipated demand, with passengers following by choosing to use the service based on these incentives.

Example 2: Multiple Modes and Dynamic Roles

Consider a situation where a passenger is planning a trip that could involve multiple transportation modes—starting with a bus ride, followed by a short bike-sharing trip, and finishing with a ride-sharing service. In the DGT model, each of these decisions could be part of a separate Stackelberg game:

- **Bus Segment**: The bus service might act as the leader in this segment, setting schedules and routes based on general passenger demand data. The passenger, as a follower, chooses whether to take the bus or opt for an alternative based on this information.
- **Bike-Sharing Segment**: Upon arriving at the bus stop, the passenger might take the lead in deciding to use a bike-sharing service, while the bike-sharing system follows by ensuring bikes are available and adjusting the pricing if demand spikes.
- **Ride-Sharing Segment**: Finally, the ride-sharing service might anticipate the passenger's demand and act as the leader by offering discounts or near-optimal routes, with the passenger following by selecting the service based on these conditions.

So, the main key differences between the Stacklberg in the GT-MMT and the DGT models can be concluded as the following:

- **Decentralization**: Unlike the GT-MMT model, where decision-making is centralized and hierarchical, the DGT model distributes decision-making across multiple localized interactions. Each passenger and transportation mode engages in a smaller, context-specific Stackelberg game, which allows for greater flexibility and adaptability to local conditions.
- **Dynamic Role Assignment**: In the GT-MMT model, roles are fixed—passengers are always leaders, and transportation modes are followers. In contrast, the DGT model allows roles to shift dynamically depending on the situation. A transportation mode might act as a leader in one context (e.g., setting prices or schedules) and as a follower in another (e.g., adjusting services in response to passenger demand).
- **Localized Optimization**: Each localized Stackelberg game aims to optimize the specific segment of the transportation network it controls. This means that instead of one large

optimization process, there are multiple smaller ones happening concurrently, each tailored to the needs and conditions of its locality.

Fifth: Communication Infrastructure

V2V and V2I communications are vital in the DGT model. They enable decentralized decision-making and ensure that the transportation system operates efficiently and safely. These components, along with the other elements of the DGT model, contribute to a community-focused, responsive, and adaptable transportation system.

1. **Vehicle-to-Vehicle (V2V) Communication:** Enables vehicles to share information with each other about their current status, including speed, direction, and any relevant local conditions. This allows for real-time coordination, such as maintaining safe distances, avoiding collisions, and optimizing traffic flow.

   Let $V_{m_i}(t)$ represent the state vector of vehicle m_i at time t, which includes its position (x_{m_i}, y_{m_i}), speed Vs_{m_i}, and direction θ_{m_i}.

$$V_{m_i}(t) = \begin{bmatrix} x_{m_i}(t) \\ y_{m_i}(t) \\ Sp_{m_i}(t) \\ \theta_{m_i}(t) \end{bmatrix} \tag{6.1}$$

   The communication between two vehicles m_i and m_j can be represented as:

$$V_{m_j}(t) \xrightarrow{\text{V2V}} V_{m_i}(t) \tag{6.2}$$

   where $V_{m_j}(t) \xrightarrow{\text{V2V}} V_{m_i}(t)$ indicates that vehicle m_j sends its state information to vehicle m_i.

2. **Vehicle-to-Infrastructure (V2I) Communication** Facilitates the exchange of information between vehicles and the surrounding infrastructure, such as traffic lights, road signs, and fog nodes. This helps vehicles receive real-time updates on traffic conditions, road closures, and other important information, which is crucial for making informed routing

and scheduling decisions. This interaction helps gather broader traffic data and implement coordinated traffic management strategies.

Let $If_k(t)$ represent the state vector of infrastructure k at time t, which includes traffic light states, road conditions, and congestion levels.

$$If_k(t) = \begin{pmatrix} \text{traffic\_light\_state} \\ \text{road\_condition} \\ \text{congestion\_level} \end{pmatrix} \tag{6.3}$$

The communication between vehicle i and infrastructure k can be represented as:

$$If_k(t) \xrightarrow{\text{V2I}} V_{m_i}(t) \tag{6.4}$$

where $If_k(t) \xrightarrow{\text{V2I}} V_{m_i}(t)$ indicates that infrastructure k sends its state information to vehicle m_i.

3. **Inter-Node Communication** Fog nodes communicate with each other to share data and coordinate decisions that affect multiple areas. This ensures that while decisions are made locally, they still contribute to the overall efficiency of the transportation network.

4. **Passenger Interaction** The system communicates with passengers through mobile applications, providing real-time updates, notifications, and alternative suggestions based on current conditions.

These components work together in the DGT model to create a decentralized, community-focused transportation system that is responsive, efficient, and adaptable. The model leverages advanced optimization algorithms, real-time data processing, and localized decision-making to optimize transportation services and improve passengers' overall experience.

Figure 6.1 illustrates one community for the model with main components.

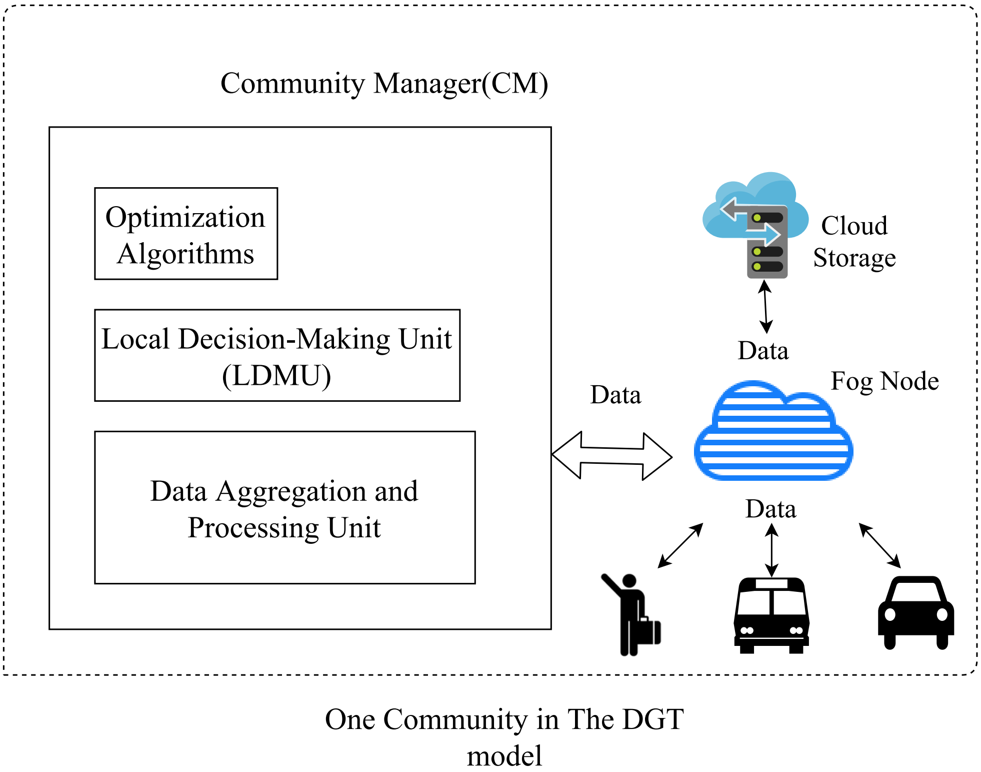

Figure 6.1: DGT model overview

6.3 Objectives function and Problem Formulation

The DGT model is designed to optimize the transportation network by incorporating a hierarchical decision-making process using the Stackelberg game theory. In this model, passengers act as leaders, and transportation modes (ride-sharing, buses, bicycles) act as followers. The primary objectives of the model are to minimize travel time and cost, maximize the number of passengers in ride-sharing and buses, and ensure bike availability.

Note: in the DGT scheme, we can use the objective function term or the utility function instead of each other to represent the main goals of the entire model or the main goal for each player in the model. The total fitness function, which is based on the objective function, for the DGT model equals the utility function of the passenger (U_{p_i}) and the utility function of each mode (ride-sharing(U_{c_j}), bus (U_{b_k}), and bike-sharing (U_{bk_n}).

To calculate the objectives for each player in the DGT involving riders, ride-sharing, we need to define the mathematical formulations for the utility functions of both the leaders (riders) and the followers (ride-sharing). These calculations involve a series of mathematical expressions that consider various factors like travel time, distance, cost, capacity, and number of passengers.

$$Fitness(A) = \alpha_{p_i} \cdot U_{p_i}(A_{p_i}) + \beta_{p_i} \cdot U_{c_j}(A_{c_j}) + U_{b_k}(A_{b_k}) + U_{bk_n}(A_{bk_n}) \tag{6.5}$$

where $Fitness(A)$ is the total fitness for all the players in the models and A is the set of the strategies of the players.

For the passenger, the utility function is to minimize the total trip time (TT_{p_i}) and the total trip fare (TF_{p_i}) (6.6).

$$U_{p_i}(A_{p_i}) = \min\left(\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}\right) \tag{6.6}$$

where α_{p_i} and β_{p_i} are the weighting factors for travel time and cost for passenger p_i, respectively. The trip time and fare equation are in the core chapter.

For ride-sharing, the objective function is maximizing the vacant seats on it, as we can see in equation (6.7).

$$U_{c_j}(A_{c_j}) = \max(1 - \frac{Nump_{e_x}^{c_j}}{Ca_c}) \tag{6.7}$$

where $Nump_{e_x}^{c_j}$ is the number of passengers in the car in route e_x, and Ca_c is the maximum capacity of the car.

Buses aim to maximize the number of passengers while following a fixed route with predefined stops. We can represent maximizing the vacant seats in the public bus as equation (6.8)

$$U_{b_k}(A_{b_k}) = \max(1 - \frac{Nump_{e_x}^{b_k}}{Ca_b}) \tag{6.8}$$

where $Nump_{e_x}^{b_k}$ is the number of passengers in the bus b_k, and Ca_b is the maximum capacity of the bus.

For the bicycle, the primary goal is to maximize the usage rate of bicycles across the network, ensuring efficient utilization and high availability. This objective (6.9) can be formulated as follows:

$$U_{bk_n}(A_{bk_n}) = \max\left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)}\right) \tag{6.9}$$

where U_{bk_n} is the utility function for the usage rate of bicycles, $Numbk_{use}(I_l)$ denotes the number of bicycles in use at location I_l, $Numbk_{av}(I_l)$ represents the total number of bicycles available at location I_l, and nbk is the number of locations or docking stations.

So, the equation 6.5 becomes:

$$Fitness(A) = \min \sum_{p_i=1}^{Nump} (\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}) + \max(1 - \frac{Nump_{ex}^{c_j}}{Ca_c}) + \max(1 - \frac{Nump_{ex}^{b_k}}{Ca_b}) + \max\left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)}\right) \tag{6.10}$$

The constraints in the DGT model ensure the feasibility and efficiency of the transportation system. Some key constraints include balancing vehicle departures and arrivals, matching rider requests with suitable transportation options, and adhering to maximum travel times and time windows. For instance, the constraint to ensure that the total distance of a passenger's trip is within the maximum allowable distance can be written as:

$$\sum_{i=1}^{n} D_{s_i,s_{i+1}} \leq MD_{p_i}^{m} \tag{6.11}$$

where $D_{s_i,s_{i+1}}$ is the distance between stops s_i and s_{i+1}, and $MD_{p_i}^{m}$ is the maximum allowable distance for the passenger.

In the core chapter, the whole objective functions, constraints, and variables are elaborated in detail. These include mathematical formulations for the Stackelberg game model, the fitness evaluation in Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), and the iterative process for real-time dynamic adjustments. The variables encompass the set of transportation modes (M), set of passengers (P), source points ($S_1^{p_i}$), destination points ($D_{np}^{p_i}$), travel times (TT_{p_i}), costs (TF_{p_i}), and other relevant parameters required for optimizing the transportation network. The comprehensive explanation thoroughly explains how the DGT model integrates these elements to achieve its optimization goals.

The following algorithms show the operation of implementing the DGT model in many steps. The final goal of the model can be implemented by evaluating these algorithms with all math equations, objective functions, and constraints related to the DGT. For a comprehensive understanding of the full notations, objectives, and constraints associated with the GT-MMT model, please refer to subsections 3.2.1, 3.2.2, 3.4.1, 3.4.2, 3.4.3, and 3.4.4. These subsections provide detailed explanations of the parameters, mathematical formulations, and constraints that guide the optimization processes within the DGT framework.

To make the problem easier to solve, we assume several relevant hypotheses. First, we did not consider the person's walking time between the stations on his or her trip. Second, the stations for the different vehicles are located in the same position. Third, while the passenger's origin is the first station of the journey, the last station is the destination. Finally, security and trust are assumed in this model.

The DGT facilitates a more coordinated and efficient urban mobility system by aligning the strategies of different transportation modes under a unified Stackelberg leadership framework. This thesis explores these alignments and the corresponding mathematical formulations that enable such strategic optimizations within these complex transportation networks.

In this thesis, we consider the following assumptions: In this thesis, several assumptions are made to support the model's framework and analysis. First, it is assumed that the public bus and the bicycle do not make any detours during their trips. Additionally, while the stations for car ride-sharing and public buses are located close to each other, they are positioned farther apart from the bicycle stations. The schedules for the public buses are considered to be fixed, both in terms of time and route. Security and trust within the system are assumed to be adequately addressed. Lastly, it is presumed that the Global Positioning System (GPS) collects data for the model, and the Geographical Information System (GIS) aids the players in utilizing this information. Please see Section (3.2) to read the full constraints. Besides the objective functions and constraints in Section (3.2), there are several constraints to ensure the proper functionality and safety of the community-based transportation system:

Constraint (6.12) network connectivity constraints, the Wi-Fi range limitation ensures that the distance between any two nodes NO_i and NO_j does not exceed the defined Wi-Fi range W.

$$TD(NO_i, NO_j) \leq W \quad \text{for all node pairs } (i, j) \tag{6.12}$$

where TD is the distance function, NO_i and NO_j represent individual nodes, and W is the Wi-Fi range.

In constraint (6.13), every mobile node NO_i must maintain a connectivity level above a specified minimum with the fog nodes F_k.

$$\sum_{i=1}^{n} Co(NO_i, F_k) \geq Co_{\min} \quad \text{for each mobile node } N_i \text{ and fog node } Fog_k \tag{6.13}$$

where Co represents connectivity and $Co_{\min}$ is the minimum required connectivity.

Furthermore, constraint (6.14) for effective node discovery within the network, the distance between any two nodes NO_i and NO_j must not exceed a discovery range Ra, ensuring all nodes are within detectable proximity of each other.

$$\text{For each node } i, j : TD(i, j) \leq Ra \tag{6.14}$$

Constraint (6.15) regarding geographic and mobility constraints, services must operate within designated geofenced areas G, with each service's location $LO(v_m)$ confined within these boundaries.

$$LO(m_i) \in G \quad \text{for vehicle } m_i \tag{6.15}$$

where $LO(m_i)$ is the location and G is the geofenced area.

Moreover, constraint (6.16), mobility patterns are regulated such that the speed of any vehicle Vs_{m_i} does not surpass the urban speed limit $Vs_{\max}$, maintaining safe and regulated travel speeds within the city environments.

$$Vs_{m_i} \leq Vs_{\max} \tag{6.16}$$

where Vs_{m_i} is the vehicle speed, and Vs_{max} is the urban speed limit.

6.4 DGT Optimization Algorithm

6.4.1 Overview and Main Components of the Algorithm

The Stackelberg game theory, PSO, and GA provide a hierarchical structure for the DGT model, which is a community-based model that uses them to satisfy the optimization. Here Fog computing is a local processing of data to reduce reliance on a central server, enabling faster and more responsive decision-making.

For a comprehensive understanding of the basic components and definitions of GA, PSO, and Stackelberg used in the DGT scheme, please refer to section 3.5. But if you want to read the basic elements for each algorithm, please read subsection 3.5.2 for the GA, subsection 3.5.3 for the PSO, and subsection 3.5.4 for the Stackelberg game theory. These subsections provide summary explanations of these algorithms that guide the optimization processes within the DGT framework. The central unit coordinates and optimizes the interactions between different transportation modes. The central unit collects data from all transportation modes and passengers, processes this data, and sends optimized instructions back to the transportation modes. This ensures a coordinated and efficient system-wide operation. For a comprehensive understanding of the full notations, objectives, and constraints associated with the DGT model, please refer to subsections 3.2.1, 3.2.2, 3.4.1, 3.4.2, 3.4.3, and 3.4.4. These subsections provide detailed explanations of the parameters, mathematical formulations, and constraints that guide the optimization processes within the DGT framework.

6.4.2 Details Operation of the Algorithm

The decentralized game theory (DGT) algorithm is designed for a community-based transportation model operating within a decentralized environment. Decision-making is distributed across fog nodes, which handle local optimizations and communicate with each other to achieve a cohesive system-wide strategy. The algorithm integrates Stackelberg game theory, Genetic Algorithms (GA), and Particle Swarm Optimization (PSO) to optimize transportation strategies within each community.

The outlines of the overall optimization process and several sub-algorithms in **DGT Optimization Algorithm** 5. The overview diagram for the algorithm is shown in Figure 6.2. The **Community Initialization and Population Generation using GA** sub-algorithm 6 focuses on generating the initial population for each community using GA. This process involves creating diverse strategy configurations as starting points for local optimizations. The **Local Optimization within Community (Using PSO and Stackelberg)** sub-algorithm 7 employs PSO and Stackelberg game theory to refine strategies within each community. It iteratively updates the strategies to minimize travel time, reduce costs, and optimize resource allocation according to the local conditions. The **Inter-Community Information Exchange** sub-algorithm 8 facilitates the exchange of strategic information between communities, ensuring that local optimizations are aligned with the overall system goals. The **Global Coordination and Strategy Aggregation** sub-algorithm 9 integrates the results from all communities, coordinating them into a cohesive global strategy. Finally, the **Global Convergence Check** sub-algorithm 10 evaluates whether the optimization process has met the convergence criteria or whether further iterations are required. Together, these algorithms enable the DGT model to operate efficiently and in a decentralized manner while optimizing transportation strategies across multiple communities.

The main algorithm for the Decentralized Game Theory (DGT) model is designed to optimize strategies within a community-based transportation system, using genetic algorithms (GA) for initial population generation and incorporating Particle Swarm Optimization (PSO) and Stackelberg Game Theory for local and global optimization. The algorithm iterates through local and global optimization processes until convergence is reached or a maximum number of iterations is completed.

The inputs for the algorithm include a set of communities, denoted as $Com = \{1, 2, \ldots, Ncom\}$, with each community representing a local cluster within the decentralized model. Additionally, the algorithm requires initial global parameters to guide the overall optimization process. The algorithm also specifies the maximum number of local iterations, denoted as T_{local}, for optimizing within each community, as well as the maximum number of global iterations, T_{global}, for the global optimization process. Finally, a convergence threshold, TS, is defined to determine the stopping criterion for the optimization process. The output of the algorithm is the globally optimized strategies, denoted as $\mathbf{A}^*_{\text{global}}$, which represent the best overall strategies across all communities.

Algorithm 5 DGT Optimization Algorithm

Input: Set of communities $Com = \{1, 2, \ldots, Ncom\}$, Initial global parameters, Maximum local iterations T_{local}, Maximum global iterations T_{global}, Convergence threshold TS

Output: Global optimized strategies $\mathbf{A}^*_{\text{global}}$

Step 1: Initialize Local Parameters
Set local objectives, constraints, and variables for community i

Step 2: Set Global Parameters
Set: global iteration counter $t_{\text{global}} = 0$, global convergence metric $\Delta_{\text{global}} = \infty$, initialize global best strategy $\mathbf{A}^*_{\text{global}} = 0$, convergence threshold TS; initialize global coordination mechanisms for inter-community communication and coordination

while $t_{global} < T_{global}$ ***and not converged globally*** **do**
- **foreach** *Community* $i \in \mathcal{C}$ **do**
  - **Step 3: Community Initialization and Population Generation**
    Call Sub-Algorithm 6: Community Initialization and Population Generation using GA
  - **Step 4: Local Optimization within Each Community**
    while $t_{local} < T_{local}$ ***and not converged locally*** **do**
    - **Call Sub-Algorithm 7:** Local Optimization (Using PSO and Stackelberg Game Theory)
    - Increment local iteration counter t_{local}
- **Step 5: Inter-Community Information Exchange**
  Call Sub-Algorithm 8
- **Step 6: Global Coordination and Strategy Aggregation**
  Call Sub-Algorithm 9
- **Step 7: Global Convergence Check**
  Call Sub-Algorithm 10
- Increment global iteration counter t_{global}

return Global optimized strategies $\mathbf{A}^*_{\text{global}}$

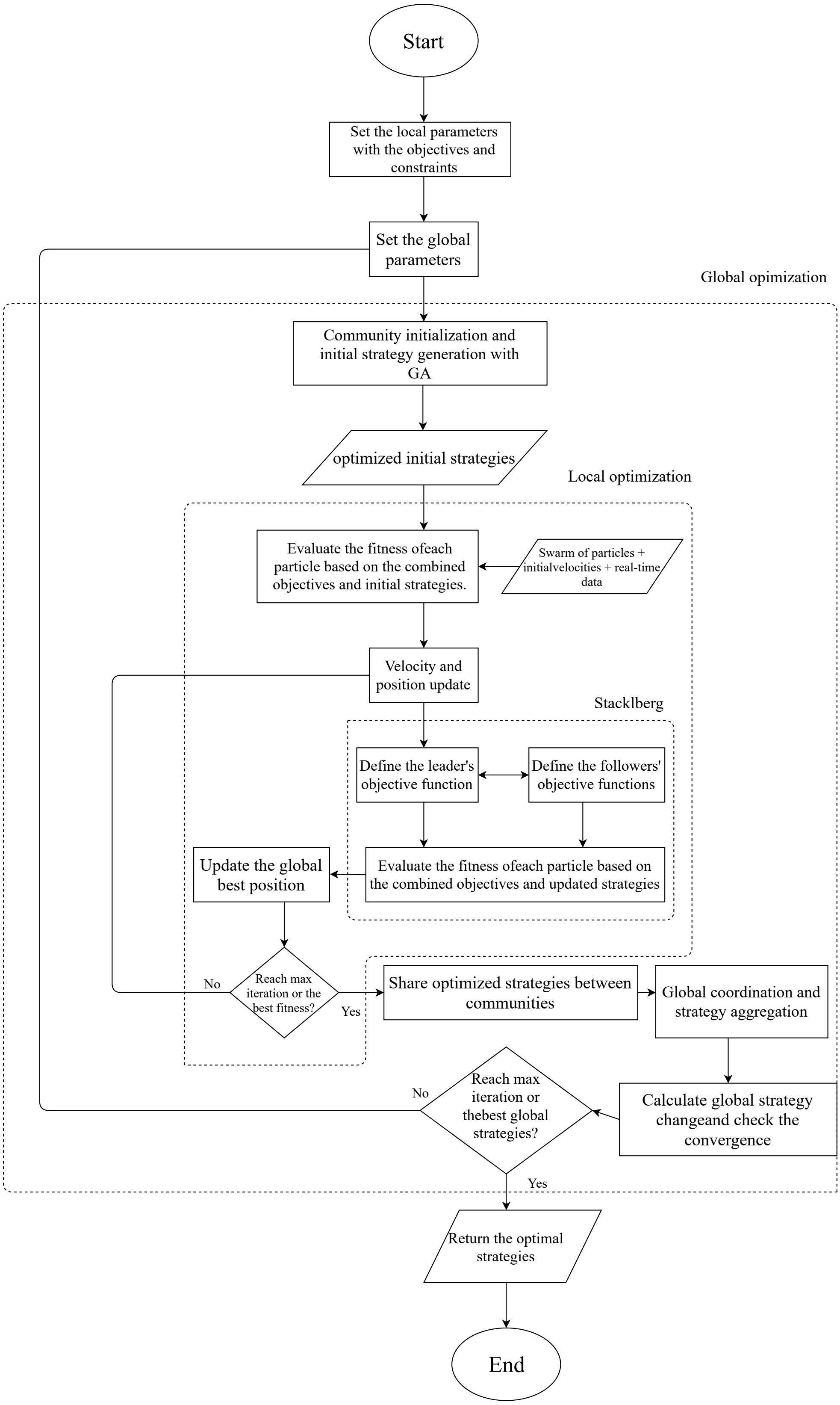

Figure 6.2: The overview diagram for the DGT algorithm

Step-by-Step Explanation

Step 1: Initialize Local Parameters

This step involves setting up the local objectives, constraints, and variables for each community i. These parameters are specific to the community and will guide the local optimization process.

Step 2: Set Global Parameters

The **global initialization step** is responsible for setting up the initial global parameters, including defining global objectives and constraints and initializing the global coordination mechanisms. This step lays the foundation for the optimization process across all communities. Global parameters, such as the global iteration counter $t_{\text{global}} = 0$, global convergence metric $\Delta_{\text{global}} = \infty$, and the global best strategy $\mathbf{A}^*_{\text{global}} = 0$, are initialized. Additionally, global coordination mechanisms are established to facilitate inter-community communication and coordination.

Step 3: Community Initialization and Population Generation using GA (sub-algorithm 6)

Process: Create initial strategies or the population (PO) for each transportation mode (ride-sharing, bus, bicycle). These strategies (A) are encoded as chromosomes, representing the potential solutions for the optimization problem. This involves evaluating the fitness of each strategy, selecting the best-performing ones, and applying crossover and mutation to create new strategies. The fitness function combines multiple objectives: minimizing travel time and fare for the passenger trip, increasing the number of passengers in ride-sharing and buses, and increasing the usability of the bike. The process of this step can be divided into five operations, as follows:

1. Create initial strategies (population)

The population is represented by multiple chromosomes. A chromosome (CH) is a vector that encodes a sequence of strategies for passenger, ride-sharing, bus, and bike-sharing in one combined structure. Randomly assign passengers to chromosomes while ensuring diversity:

$$\text{Passenger Group}_i = \{p_1, p_2, \ldots, p_n\} \quad \text{for} \quad i = 1, 2, \ldots, C$$

where n is the number of passengers assigned to each chromosome.

Here's how to represent a population of size n:

$$\mathbf{PO} = \begin{bmatrix} \mathbf{CH}_1 \\ \mathbf{CH}_2 \\ \vdots \\ \mathbf{CH}_n \end{bmatrix} \tag{6.17}$$

Where each $\mathbf{CH}_i$ is a chromosome representing the initial strategies for all players in the GT-MMT model. To read the details of the parameters and structure fare trip, see the section 3.4.1.

The combined chromosome for the GA in the GT-MMT model is a concatenation of these vectors:

$$\mathbf{CH} = \left[\mathbf{CH}_{\text{passenger}}, \mathbf{CH}_{\text{ride-sharing}}, \mathbf{CH}_{\text{bus}}, \mathbf{CH}_{\text{bike}}\right] \tag{6.18}$$

The passenger's strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{passenger}} = \begin{bmatrix} M_{\text{passenger}} & R_{\text{passenger}} & T_{\text{dep}} & F_{\text{passenger}} & M_{\text{transfer}} & U_{\text{passenger}} \end{bmatrix} \tag{6.19}$$

The component $M_{\text{passenger}}$ represents the mode of transportation selected by the passenger, such as a bus, ride-sharing, or bike-sharing. The $R_{\text{passenger}}$ component denotes the route chosen by the passenger for the journey. The departure time, indicated by T_{dep}, specifies when the passenger starts the journey. The $P_{\text{passenger}}$ component refers to the pricing strategy or cost consideration for the trip. The M_{transfer} component represents the strategy for transferring between modes during the journey. Finally, the $U_{\text{passenger}}$ component reflects the comfort and utility preferences of the passenger, which influence their satisfaction with the trip.

The bus's strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{bus}} = \begin{bmatrix} R_{\text{bus}} & Scht_{\text{bus}} & Ca_{\text{bus}} & F_{\text{bus}} \end{bmatrix} \tag{6.20}$$

The component R_{bus} represents the route selection strategy for the bus. The scheduling strategy, denoted by $Scht_{\text{bus}}$, determines the departure and arrival times as well as the service frequency. The capacity management strategy, indicated by Ca_{bus}, is focused on optimizing the use of the bus's seating and standing capacity. The pricing strategy, F_{bus}, involves setting the fare structure for bus services.

The ride-sharing strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{ride-sharing}} = \begin{bmatrix} R_{\text{ride-sharing}} & S_{\text{ride-sharing}} & C_{\text{ride-sharing}} & P_{\text{ride-sharing}} \end{bmatrix} \tag{6.21}$$

The component $R_{\text{ride-sharing}}$ represents the route selection strategy for the ride-sharing vehicle. The pickup and drop-off scheduling strategy, denoted by $S_{\text{ride-sharing}}$, determines the timing and sequence of passenger pickups and drop-offs. The capacity management strategy, indicated by $C_{\text{ride-sharing}}$, optimizes the utilization of the ride-sharing vehicle's seating capacity. Lastly, the pricing strategy, $P_{\text{ride-sharing}}$, involves setting the fare structure for ride-sharing services.

The bike-sharing strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{bike-sharing}} = \begin{bmatrix} S_{\text{bike-sharing}} & A_{\text{bike-sharing}} & M_{\text{bike-sharing}} & P_{\text{bike-sharing}} \end{bmatrix} \tag{6.22}$$

The component $S_{\text{bike-sharing}}$ represents the station placement and distribution strategy for the bike-sharing system. The bike availability management strategy, denoted by $A_{\text{bike-sharing}}$, ensures that bikes are available where and when needed. The maintenance and rebalancing strategy, indicated by $M_{\text{bike-sharing}}$, keeps bikes in good condition and properly distributed. Finally, the pricing strategy, $P_{\text{bike-sharing}}$, involves setting the fare structure for bike-sharing services.

2. Evaluate Initial Population

Fitness Evaluation: Fitness is calculated for each chromosome, generally combining the utilities for passengers and transportation modes. So the equation 6.5 can be written as:

$$Fitness(CH) = U_{p_i}(CH_{p_i}) + U_{c_j}(CH_{c_j}) + U_{b_k}(CH_{b_k}) + U_{bk_n}(CH_{bk_n}) \tag{6.23}$$

where $Fitness(CH)$ is the total fitness for all the players in the models, and CH is the set of chromosomes. The fitness function evaluates each chromosome based on the collective performance metrics, including total trip time, total fare, and capacity utilization. The goal is to find solutions that optimize these metrics for the entire group of passengers.

$$\text{Fitness}(\mathbf{CH}_i) = \alpha \cdot \left(\frac{1}{|P_i|} \sum_{j=1}^{|P_i|} TT_{p_i} \right) + \beta \cdot \left(\frac{1}{|P_i|} \sum_{j=1}^{|P_i|} TF_{p_i} \right) + \gamma \cdot \left(\frac{1}{M} \sum_{k=1}^{M} (1 - \frac{Nump_{e_x}^{c_j}}{Ca_c}) + \right) + \left(\frac{1}{M} \sum_{k=1}^{M} (1 - \frac{Nump_{e_x}^{b_k}}{Ca_b}) + \right) + \left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)} \right) \tag{6.24}$$

Where α, β, γ are weight factors that balance the importance of travel time, fare, and capacity utilization.

The parameter α_p is the weight factor for travel time for passenger p. The parameter β_p is the weight factor for cost for passenger p. The parameters γ_c, γ_b, and γ_{bk} are the weight factors for the number of passengers for the ride-sharing, bus, and bicycle, respectively. Additionally, $Numberc$, $Numberb$, and $Numberbk$ represent the number of ride-sharing, buses, and bicycles, respectively.

The total trip time and total fare for the group of passengers represented by a chromosome in the GT-MMT model are calculated by aggregating the individual travel times and fares for each passenger. Each passenger has their own origin and destination, so the calculations take into account these individual routes.

Calculate the travel time for each passenger based on their origin and destination, then sum these travel times to get the total trip time for the group. For a chromosome $\mathbf{C}_i$ representing a group of passengers, the total trip time is the sum of the individual travel times:

$$TT(CH_i) = \sum_{j=1}^{|PCH_i|} TT_{p_j} \tag{6.25}$$

Where $TT(CH_i)$ represents the total trip time for chromosome $\mathbf{CH}_i$. The notation $|PCH_i|$

indicates the number of passengers assigned to chromosome $\mathbf{CH}_i$. Finally, TT_{p_j} refers to the travel time for passenger TT_{p_j}.

Calculate the fare for each passenger based on their route and mode of transport. Sum these fares to get the total fare for the group. Similarly, the total fare for a chromosome $\mathbf{C}_i$ is the sum of the individual fares:

$$TF(CH_i) = \sum_{j=1}^{|PCH_i|} TF_{p_j} \tag{6.26}$$

Where $TF(CH_i)$ represents the total trip fare for chromosome $\mathbf{CH}_i$. The notation $|PCH_i|$ indicates the number of passengers assigned to chromosome $\mathbf{CH}_i$. Finally, TF_{p_j} refers to the travel fare for passenger TF_{p_j}.

The GT-MMT genetic algorithm's selection process effectively guides the evolutionary search toward better transportation strategies by focusing on high-fitness chromosomes. This shows how preferences and operational efficiencies interact in a multi-mode transportation system.

3. Selection Process

Selection: Select the best-performing strategies from the initial population using a tournament selection method.

Equations:

$$PO_{\text{selected}} = \text{Select}(PO_o, \text{Fitness}) \tag{6.27}$$

Tournament selection is a widely used method in genetic algorithms to select individuals for reproduction based on their fitness. In the context of the DGT model, tournament selection can be effectively applied to choose the best strategies (or chromosomes) for creating the next generation of solutions.

Steps of Tournament Selection method in the DGT Model

1. **Population Initialization:** Suppose we have a population PO consisting of PS individuals (chromosomes), where each CH_i represents a potential solution (strategies) in the GT-MMT

model. Each individual CH_i has an associated fitness value $Fitness(CH_i)$, which reflects how well it satisfies the objectives of the GT-MMT model (minimizing travel time, reducing costs for the passenger, maximizing bike availability, increasing the riders in the ride-sharing and bus)).

2. **Tournament Size:** Let the tournament size be denoted by Ts, where Ts is the number of individuals randomly selected from the population for each tournament. The tournament size Ts is a critical parameter that controls the selection pressure. A larger Ts increases the likelihood of selecting individuals with higher fitness.

3. **Tournament Selection Process:** For each selection, randomly choose Ts individuals from the population PO. Denote these selected individuals as $CH_1, CH_2, \ldots, CH_{Ts}$. Compare the fitness values of these individuals: $Fitness(CH_1), Fitness(CH_2), \ldots, Fitness(CH_{Ts})$. The individual with the highest fitness $Fitness(C_{\text{best}}) = \max(Fitness(CH_1), Fitness(CH_2), \ldots,$ $Fitness(CH_{Ts}))$ is selected as a parent for reproduction.

Mathematically, this can be represented as:

$$CH_{\text{selected}} = \arg \max_{j \in \{1,2,\ldots,T\}} Fitness(CH_j) \tag{6.28}$$

where CH_{selected} is the individual selected from the tournament.

4. **Repeat for the Desired Number of Parents:** Repeat the tournament selection process until the required number of parents PO_{parents} is selected to generate the offspring for the next generation. Typically, PO_{parents} is equal to PS, the size of the population, to maintain a constant population size across generations.

5. **Crossover and Mutation:** Use the selected parents PO_{parents} to produce offspring through crossover and mutation operations. These operations create new individuals (solutions) for the next generation. The fitness of the offspring is then evaluated based on the GT-MMT model's objectives.

6. **Repeat for Multiple Generations:** The tournament selection process, followed by crossover and mutation, is repeated across multiple generations until convergence criteria are met (maximum number of iterations or a minimal improvement in fitness).

Let's denote the population at generation Gt as $PO(Gt)$ and the fitness of an individual $CH_i(Gt)$ in this population as $Fitness(CH_i(Gt))$.

Selection Probability: The probability of an individual CH_i being selected in a tournament depends on its fitness relative to the other individuals in the tournament. Higher fitness values increase the likelihood of selection.

Expected Selection: The expected number of times an individual is selected for reproduction can be influenced by the tournament size Ts. If the tournament size is large, individuals with higher fitness values are more likely to be selected repeatedly.

Consider a population PO with a size of $PS = 10$ and a tournament size $Ts = 3$. The fitness values for each individual in the population are as follows: Individual CH_1 has a fitness value of 12, CH_2 has a fitness value of 8, CH_3 has a fitness value of 15, CH_4 has a fitness value of 7, CH_5 has a fitness value of 10, CH_6 has a fitness value of 14, CH_7 has a fitness value of 9, CH_8 has a fitness value of 13, CH_9 has a fitness value of 11, and CH_{10} has a fitness value of 6.

Randomly select three individuals, say CH_2, CH_5, CH_6. The fitness values are $F(CH_2) = 8$, $F(CH_5) = 10$, and $F(CH_6) = 14$. The best individual in this tournament is CH_6, with a fitness value of 14. CH_6 is selected as one of the parents for the next generation. This process is repeated until all required parents are selected.

4. Crossover Operation

Crossover: Combine parts of two parent solutions to generate new offspring. This process is called crossover.

Equations:

$$\text{Offspring} = \text{Crossover}(PO_{\text{selected}}) \tag{6.29}$$

Use genetic operations to generate new chromosomes, ensuring a mix of passengers:

$$\mathbf{CH}_{\text{new}} = \text{Crossover}(\mathbf{CH}_{\text{parent1}}, \mathbf{CH}_{\text{parent2}}) \tag{6.30}$$

Offspring: Offspring are new strategies generated from the crossover and mutation of selected parent strategies. They introduce diversity into the population and help explore new potential solutions.

In this thesis, the uniform crossover is used. Uniform crossover is a pivotal operation in genetic algorithms used to combine the genetic information of two parent chromosomes to produce offspring. In the GT-MMT model, uniform crossover allows for the independent consideration of each strategy component for recombination for optimizing transportation strategies. This facilitates the exploration of innovative solutions that efficiently balance the objectives of passengers and transportation providers. The ability of uniform crossover to generate diversified solutions is crucial for avoiding local optima and finding effective strategies for complex urban transportation systems. This operation is essential because it introduces new strategy combinations and enhances genetic diversity within the population, which is crucial for exploring a vast solution space and avoiding premature convergence.

In the GT-MMT framework, the uniform crossover operation is particularly important as it allows for combining different strategies from two-parent solutions. In this context, the leader is the passenger, and the followers are the ride-sharing, bus, and bike transportation modes. The uniform crossover operation can lead to offspring that inherit various advantageous traits from their parents, potentially improving the overall system performance.

Let $\mathbf{CH}_i$ and $\mathbf{CH}_j$ be two parent chromosomes. Each chromosome consists of strategies for the passenger and the transportation modes. The two parents can be represented as:

$$\mathbf{CH_i} = \left[\mathbf{CH}_{\text{passenger}}, \mathbf{CH}_{\text{ride-sharing}}, \mathbf{CH}_{\text{bus}}, \mathbf{CH}_{\text{bike}}\right] \tag{6.31}$$

$$\mathbf{CH_j} = \left[\mathbf{CH}_{\text{passenger}}, \mathbf{CH}_{\text{ride-sharing}}, \mathbf{CH}_{\text{bus}}, \mathbf{CH}_{\text{bike}}\right] \tag{6.32}$$

During the uniform crossover operation, an offspring chromosome $\mathbf{CH}_k$ can inherit some strategies from parent 1 (CH_i) and other strategies from parent 2(CH_j). Mathematically, this can be represented as:

The passenger's strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}^{new}_{\text{passenger}} = \left[M^i_{\text{passenger}} \quad R^i_{\text{passenger}} \quad T^j_{\text{dep}} \quad F^i_{\text{passenger}} \quad M^j_{\text{transfer}} \quad U^j_{\text{passenger}}\right] \tag{6.33}$$

the mode of transportation M^i_p from parent $\mathbf{C}_i$ and the departure time D_p from parent $\mathbf{C}_j$.

The bus's strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{bus}}^{new} = \begin{bmatrix} R_{\text{bus}}^{i} & Scht_{\text{bus}}^{j} & Ca_{\text{bus}}^{j} & F_{\text{bus}}^{i} \end{bmatrix} \tag{6.34}$$

The ride-sharing strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{ride-sharing}}^{new} = \begin{bmatrix} R_{\text{ride-sharing}}^{j} & S_{\text{ride-sharing}}^{i} & C_{\text{ride-sharing}}^{i} & P_{\text{ride-sharing}}^{j} \end{bmatrix} \tag{6.35}$$

The bike-sharing strategy in the GT-MMT model can be represented as a vector chromosome:

$$\mathbf{CH}_{\text{bike-sharing}}^{new} = \begin{bmatrix} S_{\text{bike-sharing}}^{i} & A_{\text{bike-sharing}}^{i} & M_{\text{bike-sharing}}^{j} & P_{\text{bike-sharing}}^{i} \end{bmatrix} \tag{6.36}$$

So, the new strategies (chromosomes) mathematically can be represented as:

$$CH_{new} = \begin{bmatrix} \mathbf{CH}_{\text{passenger}}^{new}, \mathbf{CH}_{\text{ride-sharing}}^{new}, \mathbf{CH}_{\text{bus}}^{new}, \mathbf{CH}_{\text{bike}}^{new} \end{bmatrix} \tag{6.37}$$

Where CH_{new} is derived from the respective parent strategies using the binary uniform crossover mask operation:

1. **Select Parents** Identify two parent chromosomes, $\mathbf{CH}_i$ and $\mathbf{CH}_j$, which will be used for crossover.

2. **Generate Binary Crossover Mask** Create a binary mask of the same length as the chromosomes. Each position in the mask is a binary value (0 or 1) chosen randomly.

   Example: If the length of the chromosome is 5, a possible binary mask could be $\mathbf{M} = [1, 0, 1, 0, 1]$.

3. **Apply Binary Mask to Create Offspring** The binary mask determines which genes (positions) to take from each parent. For each gene position k:

   - If the mask value $\mathbf{M}_k$ is 1, take the gene from parent $\mathbf{CH}_i$.
   - If the mask value $\mathbf{M}_k$ is 0, take the gene from parent $\mathbf{CH}_j$.

4. **Construct Offspring Chromosome** Combine the selected genes to form the new offspring chromosome $\mathbf{CH}_k$.

5. Mutation Operation

Mutation: Apply random changes to the offspring to introduce variations. The mutation alters one or more genes in the chromosome to explore new solutions.

Equations:

$$\mathbf{CH}_{\text{mutated}} = \text{Mutation}(\mathbf{CH}_{\text{new}}) \tag{6.38}$$

The mutation process in the DGT introduces random changes to offspring chromosomes, facilitating the exploration of new strategic possibilities and enhancing genetic diversity. In the GT-MMT, the mutation operation within the genetic algorithm plays a pivotal role in introducing random variations to the offspring chromosomes. This mechanism is instrumental in exploring new strategic possibilities and preserving genetic diversity, which is vital for avoiding local optima and ensuring a thorough exploration of the solution space.

We integrate mutation probability and mutation functions with the steps of applying the five mutation operators by following these steps:

1. **Define Mutation Probability:** Each gene in the chromosome has a mutation probability μ.
2. **Mutation Functions:**
   - Define the mutation functions for each operator.
   - Apply the mutation functions based on the mutation probability.
3. **Apply the Five Operators:** Sequentially apply the mutation operators, but only if the random number generated is less than the mutation probability.

We define the following notations: **CH** is the chromosome, $\mathbf{A}_{\text{passenger}}$ is the passenger strategy vector, $\mathbf{A}_{\text{bus}}$ is the bus strategy vector, $\mathbf{A}_{\text{ride-sharing}}$ is the ride-sharing strategy vector, $\mathbf{A}_{\text{bike-sharing}}$ is the bike-sharing strategy vector, μ is the mutation probability, $\mathbb{P}$ is a random number between 0 and 1, and mutate(x, μ) is the mutation function for gene x with probability μ. The operation of the mutation can be as follows:

1. **Bit Flip Mutation**:

Suitable for binary decisions in the strategy, such as whether a specific route is active or whether a certain transportation mode is available. It flips bits (0 to 1 or 1 to 0) in the chromosome to explore different binary configurations.

$$\text{mutate}_{\text{bit-flip}}(x, \mu) = \begin{cases} 1 - x & \text{if } \mathbb{P} < \mu \\ x & \text{otherwise} \end{cases} \tag{6.39}$$

2. **Swap Mutation**:

   Effective for sequences where the order is important, such as the sequence of stops in a bus route or the order of passenger pickups in ride-sharing. It randomly selects two positions in the chromosome and swaps their values, maintaining the permutation's integrity while allowing exploration of different sequence orders.

$$\text{mutate}_{\text{swap}}(\mathbf{v}, \mu) = \begin{cases} \text{swap } (v_i, v_j) & \text{if } \mathbb{P} < \mu \\ \mathbf{v} & \text{otherwise} \end{cases} \tag{6.40}$$

   Here, v_i and v_j are randomly chosen positions.

3. **Scramble Mutation**: Useful when a portion of the strategy (like pricing or scheduling) needs to be shuffled without altering the overall structure too much. It randomly selects a subset of the chromosome and scrambles the genes within that subset to introduce localized variation.

$$\text{mutate}_{\text{scramble}}(\mathbf{v}, \mu) = \begin{cases} \text{scramble}(\mathbf{v}[i:j]) & \text{if } \mathbb{P} < \mu \\ \mathbf{v} & \text{otherwise} \end{cases} \tag{6.41}$$

   Here, $[i : j]$ is a randomly chosen subset.

4. **Inversion Mutation**: Reverses the order of a subset of genes within a chromosome, which is useful for strategies involving sequences, such as reversing the order of stops on a route. This mutation helps discover more efficient routing or scheduling configurations by exploring the reversed sequences.

$$\text{mutate}_{\text{inversion}}(\mathbf{v}, \mu) = \begin{cases} \text{invert}(\mathbf{v}[i:j]) & \text{if } \mathbb{P} < \mu \\ \mathbf{v} & \text{otherwise} \end{cases} \tag{6.42}$$

Here, $[i : j]$ is a randomly chosen subset.

5. **Gaussian Mutation**: This mutation is ideal for strategies that involve continuous variables, such as adjusting pricing, capacity allocations, or departure times. Gaussian mutation adds a small random value drawn from a Gaussian (normal) distribution to each gene in the chromosome. This helps fine-tune strategies by making small, incremental changes, which is particularly useful in fine-grained optimization problems.

$$\text{mutate}_{\text{gaussian}}(x, \mu, \sigma) = \begin{cases} x + \mathcal{N}(0, \sigma) & \text{if } \mathbb{P} < \mu \\ x & \text{otherwise} \end{cases} \tag{6.43}$$

Here, $\mathcal{N}(0, \sigma)$ is a Gaussian random variable with mean 0 and standard deviation σ.

Evaluate the fitness of the new population $\mathbf{PO}_i(Gt + 1)$. Increment the generation counter: $g = g + 1$. If $g < Gt$, go back to Step 4 (Selection Process) and repeat the process until the maximum number of generations is reached or convergence criteria are satisfied. If the criteria are met, terminate the algorithm and return the final population: **Return final population $\mathbf{PO}_i$ (optimized initial strategies).**

Step 4: Local Optimization within Community (Using PSO and Stackelberg) (sub-algorithm 7)

The DGT model critically involves local optimization within the community using Particle Swarm Optimization (PSO) and Stackelberg game theory. This process ensures that the strategies of various transportation modes and passengers are optimized within each community to meet local objectives.

The set of particles in the PSO, denoted as $\mathbf{Pa} = \{pa_1, pa_2, \ldots, pa_N\}$, represents possible strategy configurations, with each particle corresponding to a different configuration. The position of particle i at time t, represented as $x_i(t)$, signifies a specific strategy configuration, while the velocity of particle i at time t, denoted as $v_i(t)$, dictates how the position, and thus the strategy, is updated. The notation $p_{\text{best},i}(t)$ indicates the best position found so far by particle i, whereas $g_{\text{best}}(t)$ represents the best position found by any particle in the swarm, also known as the global best. In game theory, S_L denotes the leader's strategy, such as a passenger, and S_F denotes the follower's strategy, such as a transportation mode. The utility function of the leader is represented by $U_L(S_L, S_F)$, while the utility function of the follower is represented by $U_F(S_L, S_F)$.

The **inputs for the algorithm** include several key elements. First, the **Community** i represents a localized area or group within the decentralized model. Another input is the **Population** $\mathbf{P}_i$, which is the set of potential solutions (chromosomes) within community i. Additionally, the **Maximum local iterations** $T_{\textbf{local}}$ specify the maximum number of iterations for the local optimization process. Finally, the **Convergence threshold** ϵ is the threshold used to determine when the local optimization process has converged. **The output** is Optimized local strategies $\mathbf{S}_i^*$ which is best strategies found for community i.

The algorithm is started by setting the local iteration counter $t_{\text{local}} = 0$.

1. PSO Operations:

a. Particle Representation: Each particle in the swarm represents a potential strategy configuration for all players (passengers, ride-sharing, buses, bikes) within the community.

$$\mathbf{CH}_p = \begin{pmatrix} \mathbf{v}_{\text{passenger}} & \mathbf{v}_{\text{ride-sharing}} & \mathbf{v}_{\text{bus}} & \mathbf{v}_{\text{bike}} \end{pmatrix} \tag{6.44}$$

Passenger Strategy Vector $\mathbf{v}_{\textbf{passenger}}$:

$$\mathbf{v}_{\text{passenger}} = \begin{pmatrix} \text{Mode} & \text{Route} & T_{\text{dep}} & D & P & C & U & T_{\text{trip}} & F \end{pmatrix} \tag{6.45}$$

Ride-Sharing Strategy Vector $\mathbf{v}_{\textbf{ride-sharing}}$:

$$\mathbf{v}_{\text{ride-sharing}} = \begin{pmatrix} \text{Route} & T_{\text{dep}} & C_{\text{capacity}} & F_{\text{price}} & P_{\text{passenger count}} \end{pmatrix} \tag{6.46}$$

Bus Strategy Vector $\mathbf{v}_{\textbf{bus}}$:

$$\mathbf{v}_{\text{bus}} = \begin{pmatrix} \text{Route} & T_{\text{dep}} & C_{\text{capacity}} & F_{\text{fare}} & P_{\text{passenger count}} \end{pmatrix} \tag{6.47}$$

Bike-Sharing Strategy Vector $\mathbf{v}_{\text{bike}}$:

$$\mathbf{v}_{\text{bike}} = \begin{pmatrix} \text{Availability} & \text{Route} & T_{\text{dep}} & C_{\text{capacity}} & U_{\text{usability}} \end{pmatrix} \tag{6.48}$$

b. Velocity and Position Update: The velocity $\mathbf{v}_p$ and position $\mathbf{CH}_p$ of each particle p are updated using the following equations:

$$\mathbf{v}_p^{t+1} = w\mathbf{v}_p^t + c_1 r_1(\mathbf{pbest}_p - \mathbf{CH}_p^t) + c_2 r_2(\mathbf{gbest} - \mathbf{CH}_p^t) \tag{6.49}$$

$$\mathbf{CH}_p^{t+1} = \mathbf{CH}_p^t + \mathbf{v}_p^{t+1} \tag{6.50}$$

c. Fitness Evaluation: The fitness of each particle $\mathbf{CH}_p$ is evaluated based on a fitness function that considers the objectives of minimizing travel time and cost, and maximizing capacity utilization.

$$F(\mathbf{CH}_p) = \alpha \cdot \left(\frac{1}{n} \sum_{k=1}^{n} T_{\text{trip},k} \right) + \beta \cdot \left(\frac{1}{n} \sum_{k=1}^{n} C_{\text{trip},k} \right) - \gamma \cdot \left(\frac{1}{M} \sum_{m=1}^{M} \text{Capacity Utilization}_m \right) \tag{6.51}$$

d. Update Personal Best and Global Best:

The personal best position $\mathbf{pbest}_p$ for each particle is updated:

$$\mathbf{pbest}_p = \begin{cases} \mathbf{CH}_p^{t+1} & \text{if } F(\mathbf{CH}_p^{t+1}) < F(\mathbf{pbest}_p) \\ \mathbf{pbest}_p & \text{otherwise} \end{cases} \tag{6.52}$$

The global best position **gbest** is updated if the new global best fitness is better:

$$\mathbf{gbest} = \arg\min_{p \in \mathbf{P}_i} F(\mathbf{CH}_p) \tag{6.53}$$

2. Dynamic Role Assignment and Stackelberg Equilibrium:

This step integrates the Stackelberg game theory, where different players (passengers, ride-sharing, buses, bikes) assume the roles of leaders or followers depending on the scenario.

a. Role Assignment: Each player j in community i is assigned a role, either as a leader or follower, based on the current scenario.

b. Optimization for Leaders and Followers:

Passenger as Leader: When the passenger is the leader, they aim to minimize their travel time and cost:

$$\min \sum_{p_i=1}^{Nump} (\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}) \tag{6.54}$$

where α_{p_i} and β_{p_i} are the weighting factors for travel time and cost for passenger p_i, respectively. The transportation modes (ride-sharing, buses, bikes) adjust their strategies as followers.

Ride-Sharing as Leader: When ride-sharing is the leader, it aims to maximize capacity utilization:

$$U_{c_j} = \max(1 - \frac{Nump_{e_x}^{c_j}}{Ca_c}) \tag{6.55}$$

where $Nump_{e_x}^{c_j}$ is the number of passengers in the car in route e_x, and Ca_c is the maximum capacity of the car. Passengers, buses, and bikes adjust their strategies as followers.

Bus as Leader: Buses aim to maximize their capacity utilization:

$$U_{b_k} = \max(1 - \frac{Nump_{e_x}^{b_k}}{Ca_b}) \tag{6.56}$$

where $Nump_{e_x}^{b_k}$ is the number of passengers in the bus b_k, and Ca_b is the maximum capacity of the bus. Passengers, ride-sharing, and bikes adjust accordingly.

Bike-Sharing as Leader: Bike-sharing focuses on maximizing usability:

$$U_{bk_n} = \max\left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)}\right) \tag{6.57}$$

where U_{bk_n} is the utility function for the usage rate of bicycles, $Numbk_{use}(I_l)$ denotes the number of bicycles in use at location I_l, $Numbk_{av}(I_l)$ represents the total number of bicycles available at location I_l, and nbk is the number of locations or docking stations. Passengers and other modes adjust their strategies as followers.

After each iteration, the change in strategies $\Delta \mathbf{A}_i^{t_{\text{local}}+1}$ is calculated. The process continues until the change in strategies is less than the convergence threshold ϵ or the maximum number of local iterations T_{local} is reached.

Step 5: Inter-community information exchange (sub-algorithm 8)

Inter-community information exchange is a critical component of the DGT model. This process ensures that information flows between different communities, enabling them to make informed decisions that contribute to the overall optimization of the transportation network. Below is a detailed explanation of the Inter-community information exchange process, including mathematical equations, variables, and a real-world scenario.

In the DGT model, each community operates semi-autonomously, optimizing its transportation strategies based on local data. However, to avoid conflicts and ensure global coherence, communities must exchange information regularly. This process involves sharing relevant data about routes, schedules, passenger demands, and resource allocation across communities, which allows for coordinated decision-making that benefits the entire transportation system.

The variables and notations used in the algorithm are defined as follows. Com_i represents the i-th community. The strategy set for community Com_i is denoted by $\mathbf{A}_i$, while $\mathbf{I}_i$ indicates the information set for community Com_i. Information exchanged between community Com_i and community Com_j is represented by $\mathbf{E}_{ij}$. The global strategy contributions from community Com_i are denoted by $\mathbf{G}_i$. Additionally, T_{global} is the global iteration counter, and ϵ represents the convergence threshold.

The information exchange process can be described as follows:

1. Information Collection: Each community $\mathcal{C}_i$ collects local data and strategies, represented by $\mathbf{I}_i$. This includes:

$$\mathbf{I}_i = \{D_{p_i}, R_{s_i}, R_a\} \tag{6.58}$$

Where D_{p_i} represents passenger demand patterns, R_{s_i} represents route and schedule optimizations, and R_a represents resource allocations, such as bus capacities and bike availability.

2. Information Sharing: Communities exchange their information sets with neighbouring communities Com_j. The shared information $\mathbf{E}_{ij}$ from community Com_i to community Com_j includes:

$$\mathbf{E}_{ij} = \mathbf{I}_i \cap \mathbf{I}_j \tag{6.59}$$

This includes optimized routes, schedules, and demand forecasts.

3. Global Strategy Contribution: Each community updates its strategies based on the received information. This update is done to ensure that local optimizations do not conflict with global goals.

$$\mathbf{G}_i^{t+1} = f(\mathbf{S}_i^t, \mathbf{E}_{ij}) \tag{6.60}$$

where $f(\cdot)$ represents the strategy update function that integrates the exchanged information into the local strategy.

Consider a city with two communities Com_1 and Com_2, representing different districts. Each community has its own bus routes, ride-sharing services, and bike-sharing stations. The communities are adjacent, and some transportation modes, like buses and ride-sharing, serve both areas. Here's how Inter-Community Information Exchange plays out in this scenario:

- Community Com_1 has a high demand for morning bus routes, while Community Com_2 experiences peak demand in the evening. Both communities share a major bus route connecting a common business district.
- Initially, each community optimizes its transportation strategies independently. Community Com_1 might schedule more buses in the morning, while Community Com_2 focuses on evening services.
- Through Inter-Community Information Exchange, Community Com_1 shares its morning demand data and bus schedules with Community Com_2, and vice versa.
- Based on the exchanged information, both communities adjusted their strategies:
  - Community Com_1 may decide to send a few morning buses to Community Com_2 in the afternoon, optimizing bus utilization.

– Community Com_2 might allocate more ride-sharing vehicles in the evening to support the returning buses from Community Com_1.

Step 6: Global Coordination and Strategy Aggregation (sub-algorithm 9)

Global coordination and strategy aggregation in the DGT is a critical phase where strategies from different communities are harmonized to ensure that the overall transportation system works cohesively. This process involves the aggregation of locally optimized strategies from individual communities into a global strategy that benefits the entire network. Below is a detailed explanation of Global Coordination and Strategy Aggregation, including mathematical equations, variables, and a real-world scenario.

In the DGT model, each community optimizes its strategies locally based on local data and objectives. However, these local strategies must be coordinated to ensure that they contribute positively to the global objectives of the transportation system. Global Coordination and Strategy Aggregation involve collecting these local strategies, evaluating them in the context of the entire system, and updating them to form a cohesive global strategy.

The variable Com_i represents the i-th community. The strategy set for community Com_i is denoted as $\mathbf{A}_i$, while the aggregated global strategy set is represented by $\mathbf{A}_{\text{global}}$. The weight assigned to each community Com_i based on its importance or size is indicated by w_i. The global iteration counter is denoted as T_{global}, and the convergence threshold is represented by ϵ.

Global Strategy Aggregation Process

The process can be described mathematically as follows:

1. Collection of Local Strategies:

Each community Com_i has its optimized local strategy $\mathbf{A}_{local}$ after completing its local optimization. These strategies are collected for aggregation $\mathbf{A}_{\text{all}} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{Comn}\}$, where $Comn$ is the total number of communities.

2. Weighted Aggregation of Strategies:

The global strategy $\mathbf{S}_{\text{global}}$ is obtained by aggregating the local strategies, taking into account the relative importance or size of each community:

$$\mathbf{A}_{\text{global}} = \sum_{i=1}^{Comn} w_i \mathbf{A}_i \tag{6.61}$$

where w_i is the weight assigned to community $\mathcal{C}\wr\Updownarrow_i$. This weight could be based on factors like population size, transportation demand, or geographic area.

3. Global Fitness Evaluation:

The aggregated global strategy $\mathbf{A}_{\text{global}}$ is evaluated to determine its fitness in meeting the overall objectives of the transportation system:

$$Fitness_{\text{global}}^{t} = \alpha \cdot \left(\frac{1}{N} \sum_{k=1}^{N} TT_{\text{trip},k} \right) + \beta \cdot \left(\frac{1}{N} \sum_{k=1}^{N} TF_{\text{trip},k} \right) - \gamma \cdot \left(\frac{1}{M} \sum_{m=1}^{M} \text{Capacity Utilization}_m \right) \tag{6.62}$$

In this context, N represents the total number of passengers. The variables $TT_{\text{trip},k}$ and $TF_{\text{trip},k}$ denote the trip time and cost for passenger k, respectively. The symbol γ is a weight factor for capacity utilization.

4. Strategy Adjustment:

If global fitness does not meet the desired criteria, strategies are adjusted. This adjustment could involve modifying local strategies to better align with global objectives:

$$\mathbf{A}_i^{\text{new}} = \mathbf{A}_i + \lambda \cdot (\mathbf{A}_{\text{global}} - \mathbf{A}_i) \tag{6.63}$$

where λ is a learning rate that controls the degree of adjustment.

5. Convergence Check:

The process iterates until the global strategy $\mathbf{A}_{\text{global}}$ converges, i.e., when changes between iterations fall below a predefined threshold ϵ:

$$\Delta \mathbf{A}_{\text{global}}^{t+1} = \| \mathbf{A}_{\text{global}}^{t+1} - \mathbf{A}_{\text{global}}^{t} \| < \epsilon \tag{6.64}$$

Sample example: Consider a city where the DGT model is implemented to manage transportation across multiple districts (communities). Each community optimizes its local transportation strategy based on passenger demand, vehicle availability, and local traffic conditions. However, these communities share several key routes and transportation resources, requiring coordination to ensure efficient use of the transportation network.

- **Community A** might prioritize bus routes to handle a high volume of daily commuters.
- **Community B** could focus on ride-sharing to reduce traffic congestion.
- **Community C** may emphasize bike-sharing to promote eco-friendly transportation.

Step-by-Step Global Coordination

1. **Local Optimization**: Each community optimizes its local strategy $\mathbf{A}_i$ independently.
2. **Information Sharing**: Communities exchange strategies and relevant information (e.g., passenger flows, available vehicles).
3. **Global Aggregation**: The DGT system aggregates these local strategies using weighted averaging, where larger or more critical communities have a greater influence on the global strategy.
4. **Global Fitness Evaluation**: The global strategy $\mathbf{A}_{\text{global}}$ is evaluated. For example, the system may find that while Community A's strategy is locally efficient, it leads to bottlenecks when combined with Communities B and C strategies.
5. **Strategy Adjustment**: If necessary, the global strategy is refined. For instance, Community A may reduce its bus services slightly to allow better synchronization with Community B's ride-sharing schedules.
6. **Convergence**: The process continues until the global strategy is optimized, ensuring that all communities operate in a way that benefits the entire city's transportation network.

Step 7:Global Convergence Check (sub-algorithm 10)

In the DGT model, each community operates independently, optimizing its local strategies based on local data and objectives. However, the system as a whole must work cohesively. The Global

Convergence Check ensures that the aggregated strategies from all communities do not just perform well locally but also contribute to the overall efficiency of the transportation network. This involves checking whether the global optimization process has converged, i.e., whether further iterations would result in negligible improvements.

The variable $\mathbf{A}^t_{\text{global}}$ represents the global strategy set at iteration t. The term $Fitness^t_{\text{global}}$ denotes the fitness value of the global strategy at iteration t. The variable $\Delta Fitness^t_{\text{global}}$ indicates the change in global fitness between iterations t and $t-1$. The symbol ϵ is the convergence threshold, which is a small positive number that determines when the algorithm should stop. Finally, T_{global} represents the maximum number of global iterations allowed.

The convergence check involves comparing the change in global fitness between successive iterations. The process is mathematically represented as follows:

$$\Delta Fitness^t_{\text{global}} = \left| Fitness^t_{\text{global}} - Fitness^{t-1}_{\text{global}} \right| \tag{6.65}$$

Convergence is achieved when:

$$\Delta Fitness^t_{\text{global}} < \epsilon \quad \text{OR} \quad t \geq T_{\text{global}} \tag{6.66}$$

where ϵ is a predefined threshold that indicates when the change in fitness is sufficiently small to stop the optimization process, and T_{global} is the maximum number of iterations allowed.

Step-by-Step Global Convergence Check

1. **Initialize Global Parameters**:

   At the start of the optimization process, set $t = 0$ and $Fitness^0_{\text{global}}$ based on the initial global strategy $\mathbf{S}^0_{\text{global}}$.

2. **Evaluate Global Fitness**:

   For each iteration t, evaluate the global fitness $Fitness^t_{\text{global}}$ using the aggregated strategies from all communities:

$$Fitness^t_{\text{global}} = \alpha \cdot \left(\frac{1}{N} \sum_{k=1}^{N} TT_{\text{trip},k} \right) + \beta \cdot \left(\frac{1}{N} \sum_{k=1}^{N} TF_{\text{trip},k} \right) - \gamma \cdot \left(\frac{1}{M} \sum_{m=1}^{M} \text{Capacity Utilization}_m \right) \tag{6.67}$$

In this context, N represents the total number of passengers. The variables $TT_{\text{trip},k}$ and $TF_{\text{trip},k}$ denote the trip time and cost for passenger k, respectively. The symbol γ is a weight factor for capacity utilization.

3. **Check Convergence**:

   Calculate the change in global fitness:

$$\Delta Fitness^t_{\text{global}} = \left| Fitness^t_{\text{global}} - Fitness^{t-1}_{\text{global}} \right| \tag{6.68}$$

   Compare $\Delta Fitness^t_{\text{global}}$ with the convergence threshold ϵ:

   - If $\Delta Fitness^t_{\text{global}} < \epsilon$, the algorithm has converged, and the optimization process can stop.
   - If $t \geq T_{\text{global}}$, stop the process as the maximum number of iterations has been reached, even if convergence is not achieved.

4. **Update Global Strategy**:

   If convergence is not achieved, update the global strategy $\mathbf{A}^t_{\text{global}}$ based on feedback from the fitness evaluation. This could involve re-adjusting local strategies or re-aggregating the global strategy.

5. **Repeat the Process**:

   Increment the global iteration counter t and repeat the process until convergence is achieved or the maximum number of iterations T_{global} is reached.

Algorithm 6 Community Initialization and Population Generation using GA

Input: Community i, Local parameters, Population size PS

Output: Initialized population $\mathbf{Po}_i$ for community i

Step 1: Initialization

Set Parameters: Define the population size N, set the crossover probability pc, set the mutation probability p_m, define the number of generations G, and initialize the iteration counter $g = 0$.

Step 2: Generate Initial Population $\mathbf{PO}_i$ using GA

Generate an initial population $\mathbf{Po}_i = \{\mathbf{CH}_1, \mathbf{C}_2, \ldots, \mathbf{CH}_N\}$ where each chromosome $\mathbf{CH}_j$ ($j \in \{1, 2, \ldots, N\}$) is a vector representing a candidate solution (strategy):

$$\mathbf{CH}_j = \begin{bmatrix}\mathbf{v}_{\text{passenger}} & \mathbf{v}_{\text{ride-sharing}} & \mathbf{v}_{\text{bus}} & \mathbf{v}_{\text{bike}}\end{bmatrix}$$

Step 3: Evaluate Fitness of Initial Population

Evaluate the fitness $Fitness(\mathbf{CH}_j)$ of each individual $\mathbf{CH}_j \in \mathbf{Po}_i$.

Step 4: Selection Process

Select pairs of parent chromosomes from the population $\mathbf{Po}_i$ based on their fitness scores.

$$\mathbf{Po}_{\text{selected}} \leftarrow \text{SelectParents}(\mathbf{Po}_i, Fitness(\mathbf{Po}_i))$$

Step 5: Crossover operation

For each pair of selected parents, apply crossover with probability p_c to generate offspring:

$$\mathbf{CH}_{\text{child}} = \text{Crossover}(\mathbf{CH}_{\text{parent1}}, \mathbf{CH}_{\text{parent2}}, p_c)$$

Step 6: Mutation operation

Apply mutation to each gene in the offspring chromosomes with mutation probability p_m:

$$\mathbf{CH}_{\text{mutated}} = \text{Mutate}(\mathbf{CH}_{\text{child}}, p_m)$$

Step 7: Fitness Evaluation of New Population

$$Fitness(\mathbf{P}_i^{(g+1)}) = \text{EvaluateFitness}(\mathbf{P}_i^{(g+1)})$$

Step 8: Repeat or Terminate: Increment generation counter: $g = g + 1$

if $g < G$ **then**

  Go to Step 4: Selection Process

else

  Terminate and **Return final population $\mathbf{Po}_i$ (optimized initial strategies)**

Algorithm 7 Local Optimization within Community (Using PSO and Stackelberg)

Input: Com (number of the community), Population $\mathbf{Po}_i$, Maximum local iterations T_{local}, Convergence threshold ϵ

Output: Optimized local strategies $\mathbf{S}_i^*$

Set local iteration counter $t_{\text{local}} = 0$

while $t_{local} < T_{local}$ ***and not converged locally*** **do**

  Step 3.1: PSO Operations

  for *each particle* $p \in \mathbf{P}_i^t$ **do**

    Update velocity and position:

$$\mathbf{v}_p^{t+1} = w\mathbf{v}_p^t + c_1 r_1(\mathbf{pbest}_p - \mathbf{CH}_p^t) + c_2 r_2(\mathbf{gbest} - \mathbf{CH}_p^t) \tag{6.69}$$

$$\mathbf{CH}_p^{t+1} = \mathbf{CH}_p^t + \mathbf{v}_p^{t+1} \tag{6.70}$$

    Calculate the fitness Update $\mathbf{pbest}_p$ and $\mathbf{gbest}$ based on fitness

  Step 3.2: Dynamic Role Assignment and Stackelberg Equilibrium

  for *each player* $j \in$ *community* i **do**

    if *Passenger* j *is the leader* **then**

$$\min \sum_{p_i=1}^{Nump} (\alpha_{p_i} TT_{p_i} + \beta_{p_i} TP_{p_i}) \tag{6.71}$$

      The transportation modes (ride-sharing, buses, bikes) adjust their strategies as followers.

    else if *Ride-sharing is the leader* **then**

$$U_{c_j} = \max(1 - \frac{Nump_{ex}^{c_j}}{Ca_c}) \tag{6.72}$$

      Passengers, buses, and bikes adjust their strategies as followers.

    else if *Bus is the leader* **then**

$$U_{b_k} = \max(1 - \frac{Nump_{ex}^{b_k}}{Ca_b}) \tag{6.73}$$

      Passengers, ride-sharing, and bikes adjust their strategies as followers.

    else if *Bike-sharing is the leader* **then**

$$U_{bk_n} = \max\left(\frac{\sum_{n=1}^{nbk} Numbk_{use}(I_l)}{\sum_{n=1}^{nbk} Numbk_{av}(I_l)}\right) \tag{6.74}$$

      Passengers, ride-sharing, and buses adjust their strategies as followers.

  Update strategies $\mathbf{A}_i^{t+1}$ based on the Stackelberg equilibrium for each role configuration

  Step 3.3: Local Convergence Check

  Calculate change in local strategies $\Delta\mathbf{A}_i^{t_{\text{local}}+1}$ Check convergence: Converged $= \left(\Delta\mathbf{A}_i^{t_{\text{local}}+1} < \epsilon\right)$ OR $(t_{\text{local}} \geq T_{\text{local}})$ Increment local iteration counter t_{local}

Return optimized local strategies $\mathbf{A}_i^*$

Algorithm 8 Inter-Community Information Exchange

Input: Communities i and k, strategies $\mathbf{A}_i^{t+1}$, $\mathbf{A}_k^{t+1}$
Output: Updated strategies $\mathbf{A}_i^{t+1}$, $\mathbf{A}_k^{t+1}$

Information Exchange:
Share optimized strategies between communities i and k

Strategy Adjustment:
Adjust strategies based on exchanged information:

$$\mathbf{A}_i^{t+1} \leftarrow \text{Adjust}(\mathbf{A}_i^{t+1}, \mathbf{A}_k^{t+1}) \tag{6.75}$$

Algorithm 9 Global Coordination and Strategy Aggregation

Input: Aggregated local strategies $\mathbf{A}_i$ from all communities
Output: Global optimized strategy $\mathbf{A}^*_{\text{global}}$

Step 1: Aggregate Local Strategies
Combine local strategies into a global strategy $\mathbf{A}_{\text{global}}$
Aggregate strategies from all communities:

$$\mathbf{A}^*_{\text{global}} = \sum_{i=1}^{M} w_i \mathbf{A}_i^{t+1} \tag{6.76}$$

where w_i are weights assigned to each community based on its importance

Step 2: Evaluate Global Fitness
Evaluate the fitness of the global strategy $F(\mathbf{A}_{\text{global}})$

Step 3: Optimize Global Strategy (if necessary)
If global fitness is not near-optimal, adjust strategy components **Return Global optimized strategy $\mathbf{A}^*_{\text{global}}$**

Algorithm 10 Global Convergence Check

Input: Global strategy $\mathbf{A}_{\text{global}}^{t_{\text{global}}}$, Previous global strategy $\mathbf{A}_{\text{global}}^{t_{\text{global}}-1}$, Convergence threshold ϵ
Output: Boolean indicating global convergence

Step 1: Calculate Global Strategy Change
Calculate the change in global strategy: $\Delta\mathbf{A}_{\text{global}} = \left|\mathbf{A}_{\text{global}}^{t_{\text{global}}} - \mathbf{A}_{\text{global}}^{t_{\text{global}}-1}\right|$

Step 2: Check Convergence
Check if the change in strategy is below the convergence threshold: Global Converged = $(\Delta\mathbf{A}_{\text{global}} < \epsilon)$ OR $(t_{\text{global}} \geq T_{\text{global}})$

Return Boolean value indicating global convergence

6.5 Simulations and Results

6.5.1 Simulation Setup

The discussion in this section is focused on the effectiveness of the proposed model when the passenger uses different transportation modes on his trip. To evaluate the model in this thesis, the authors use OMNET++ to build the proposed model. The module interfaces and protocol messages have been implemented by extending $Inet$ and $Veins$ modules, which are written in C++. $Venis$ is used to realize the transportation mobility environment and $Inet$ framework to build a Wifi and the Fog layer with its services. Also, the researchers connect $Veins$ with SUMO to create the random mobility scenario. By integrating these tools and methodologies, a comprehensive analysis of the interactions between various transport modes and their users can be achieved in a dynamic, urban transportation setting.

To read the whole setup for the experiment, please refer to section 3.6. Here, the main addition is the creation of the Fog infrastructure. To create the fog infrastructure in the DGT (Decentralized Game Theory) model using simulation tools such as OMNeT++, INET, and related frameworks, follow these steps. These tools help simulate a distributed network environment where fog nodes (representing edge devices) process and exchange information locally, reducing the need for centralized processing.

1. **Define Fog node behavior in OMNeT++:** Define custom modules in OMNeT++ for fog nodes. Each fog node can be represented by a C++ class that inherits from the `cSimpleModule` class in OMNeT++. This class manages the fog node's behaviour, processing capabilities, and communication functions.

2. **Initialize the Fog nodes:**

   Initialize parameters for fog nodes in the `initialize` method. This can include processing power, storage capacity, and bandwidth limitations.

3. **Data processing and management:** Implement local data processing logic in the `handleMessage` method. When a message is received, the fog node can decide whether to process the data locally or forward it to other nodes based on specific criteria.

4. **Simulate communication between Fog nodes** Use the INET framework to simulate communication links between fog nodes, edge devices, and the cloud. Define communication channels using `cChannel` classes to represent different network properties (e.g., delay, bandwidth).

5. **Implement Fog services and decision-Making functions**

   Implement decision-making and optimization functions within the fog nodes to handle data aggregation, processing, and local optimization tasks.

6. **Simulate mobility and dynamic environments**

   If fog nodes represent mobile devices or vehicles, use mobility modules in INET to simulate their movement and dynamic changes in the network.

   Integrate with Other Components in the DGT Model Integrate fog nodes with the game-theoretical decision-making models, the Stackelberg game and Particle Swarm Optimization (PSO) to simulate decentralized optimization across communities.

7. **Visualization and analysis** Utilize OMNeT++'s built-in visualization tools to display the fog nodes' operations, data flows, and optimization outcomes.

By following these steps, you can simulate the fog infrastructure in the DGT model using OMNeT++, INET, and other relevant simulation tools, reflecting a community-based, decentralized approach to transportation optimization.

6.5.2 Experiment Evaluations and Results Discussion

1. DGT Performance

a. passenger's normalized average trip time

Figure 6.3 shows that the passenger's normalized average trip time differs from one transportation mean to another. We note that the average time value when the passengers use the public bus only or two transportation types or three modes is not much different from 9:00 until 16:00. The average value is usually identical if the passenger uses the public bus in combination with car ride-sharing or if he takes the public bus only. However, this value differs from 11:00 to 13:00 and also from 16:00 to 18:00. In Tables 6.1- 6.2, the researchers use the public bus as a reference to calculate the normalized average trip time-saving rate. In Table 6.1, the normalized average trip time-savings rate between using only the public bus and using the public buses with car ride-sharing is 8 at the beginning of the day and 15 at the end of the day. Table 6.1 shows the savings rate between taking the bus only over the whole journey versus taking all three means of transportation. The savings rate is 18 from 9:00 until 12:00, and this rate increases to reach 45 at 18:00.

b. passenger's normalized average trip price

From Figure 6.4, we note that the normalized average trip price when the passenger takes the three transportation modes (public bus, ride car-sharing and bicycle-sharing) is lower compared with using two modes or only one mode. In Table 6.4, the normalized average trip price-savings rate between using only the public bus and using the public buses in combination with car ride-sharing is 0 at the beginning of the day and 5 at the end of the day. The savings rate between taking the bus only during the whole journey and taking all three transportation means in combination is

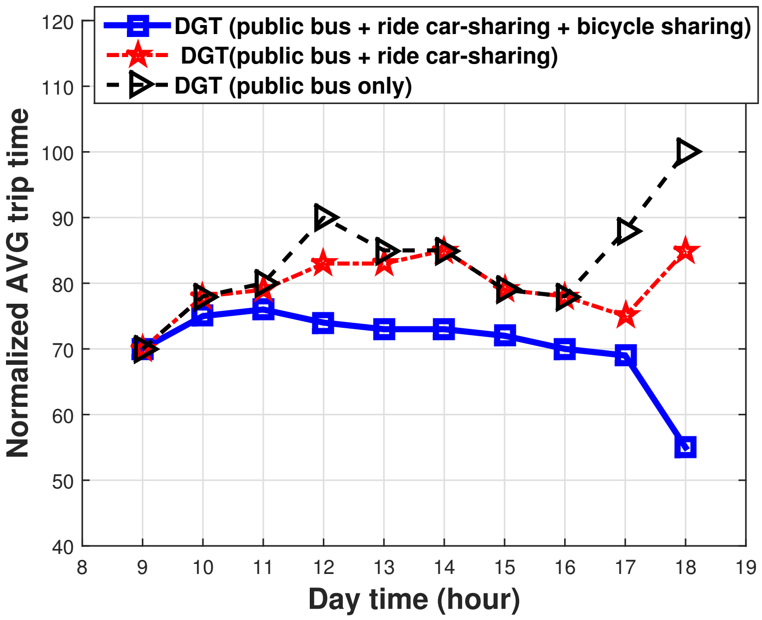

Figure 6.3: Normalized average trip time using the DGT with variety of transportation modes

Table 6.1: Normalized average trip time saving rate between using the DGT (public bus only) and using the DGT (public bus + ride car-sharing)

Day time (hour)	**DGT(public bus only)**	**DGT(public bus and ride-sharing) ($)**	**Saving rate (%)**
9:00-12:00 a.m	90	82	**8**
12:01-15:00 a.m	80	80	**0**
15:01-18:00	100	85	**15**

Table 6.2: Normalized average trip time saving rate between using the DGT (public bus only) and using the DGT (the three modes)

Day time (hour)	**DGT(public bus only) ($)**	**DGT(the three modes) ($)**	**Saving rate (%)**
9:00-12:00 a.m	90	72	**18**
12:01-15:00 a.m	80	71	**9**
15:01-18:00	100	55	**45**

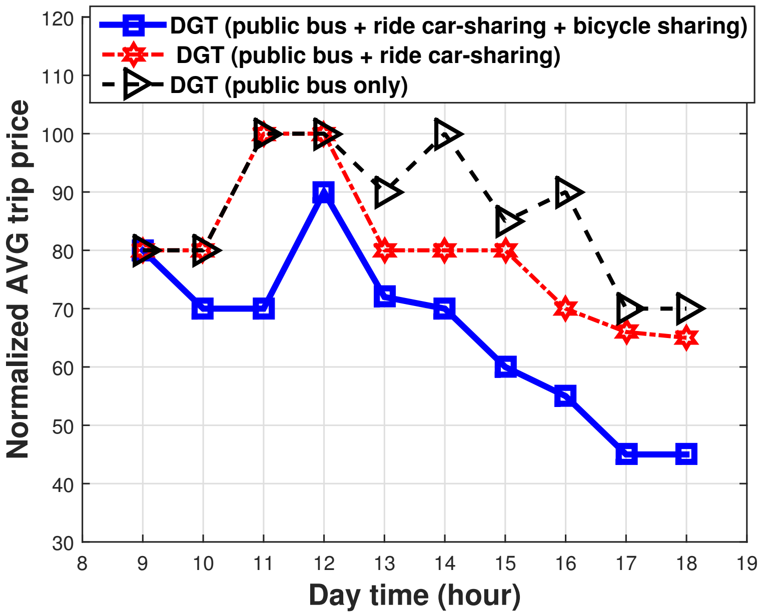

Figure 6.4: Normalized average trip price using the DGT with variety of transportation modes

10 from 9:00 until 12:00. This rate increases to 25 at 18:00, as evident in Table 6.3.

Conclusion

We note from the results above that the saving rate time between using the bus only and using the bus and car ride-sharing in combination is not significantly different and reaches zero at 15:00. In contrast, it is possible to reach the destination faster if the passenger uses the three types of transportation throughout the entire trip. As for the cost of the journey, the average trip price is almost equal whether public bus only or two types of transportation are used in combination. When the user uses the three modes (public bus, ride car-sharing and bicycle) the price is lower than using two or one type of transportation. We conclude from previous that for people who aim to arrive at the fastest time and the lowest cost, the best option provided by the proposed model is to move between the three different types of transportation. As for the passenger who does not want to take the bicycle for any reason, the most appropriate option is to ride the bus only without using any other means of transportation.

Table 6.3: Normalized average trip price saving rate between using the DGT (public bus only) and using the DGT (the three modes)

Day time (hour)	**DGT(public bus only) ($)**	**DGT(the three modes) ($)**	**Saving rate (%)**
9:00-12:00 a.m	100	90	**10**
12:01-15:00 a.m	85	60	**25**
15:01-18:00	70	45	**25**

2. Comparison DGT with model in [5]

As we mentioned previously, the DGT is multi-mode transportation to optimize the trip time and fare for the whole trip. In this subsection, we compare the DGT with the proposed model in [5], which optimized the trip cost only without considering the trip time. This study developed a multi-model to solve the problem of dynamic ridesharing within different days. The authors considered whether the user with a private car or not and can choose to be a single driver, a participant, or take public transportation.

Figure 6.5 shows that the normalized average trip price when using DGT is lower in comparison with the model presented in the reference [5]. The normalized average trip price in both models is not much different from 9:00 until noon. But in the afternoon, the normalized average trip

Table 6.4: Normalized average trip price saving rate between using the DGT (public bus only) and using the DGT (public bus + ride car-sharing)

Day time (hour)	**DGT(public bus only) ($)**	**DGT(public bus + ride car-sharing) ($)**	**Saving rate (%)**
9:00-12:00	100	100	**0**
12:01-15:00	85	80	**5**
15:01-18:00	70	65	**5**

in DGT drops and reaches 45 at 18:00. This decrease is not big for the scheme in [5]. In Table 6.5, the normalized average trip price-savings rate between using the DGT and using the model in [5] is 5 at the beginning of the day and 25 at the end of the day. From comparing

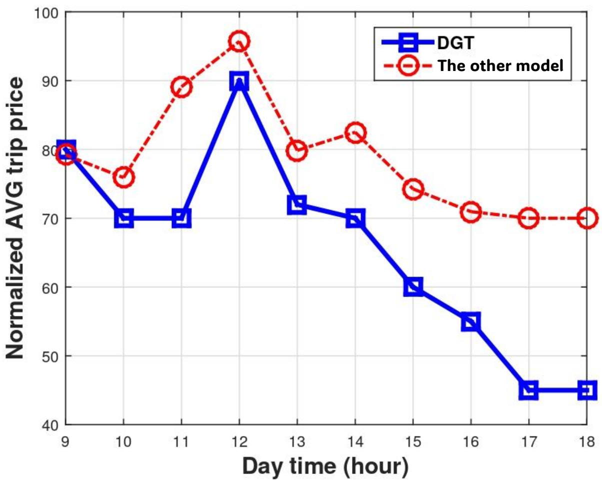

Figure 6.5: Normalized average trip price using the DGT with various transportation modes and using the model in reference [5]

Table 6.5: Normalized average trip price saving rate between using the DGT and using the model in [5]

Day time (hour)	**The model in reference [5] ($)**	**DGT model($)**	**Saving rate (%)**
9:00-12:00 a.m	95	90	**5**
12:01-15:00 a.m	75	60	**15**
15:01-18:00	70	45	**25**

Figure 6.5 and Figure 6.4, we can conclude that the journey normalized average price when the passenger takes the bus only in the DGT model is almost the same as the price when using the multi-transportation mode model in [5].

3. Comparison Between DGT and GT-MMT Model

To compare the DGT (Decentralized Game Theory) model and the GT-MMT (Game Theory Multi-Mode Transportation) model, we evaluate their performance in terms of travel time reduction and cost savings across different periods during the day. This comparison provides insights into how both models perform under varying traffic conditions, demand levels, and operational constraints.

a. Travel Time Comparison Table Across Different Periods

Table 6.6: Travel Time Comparison Across Different Periods

Period	**Model**	**Morning Peak (min)**	**Midday (min)**	**Evening Peak (min)**	**Percentage Improvement (%)**
Morning Peak	GT-MMT DGT	42.0 39.0	35.0 32.0	46.0 42.0	16.7 20.0
Midday	GT-MMT DGT	30.0 28.0	25.0 22.0	33.0 30.0	16.7 21.4
Evening Peak	GT-MMT DGT	45.0 42.0	37.0 33.0	50.0 46.0	17.8 21.4

The percentage improvement in travel time for each period is calculated using the formula:

$$\text{Percentage Improvement (\%)} = \left(\frac{T_{\text{GT-MMT}} - T_{\text{DGT}}}{T_{\text{GT-MMT}}} \right) \times 100 \tag{6.77}$$

In this formula, $T_{\text{GT-MMT}}$ represents the optimized average travel time for the GT-MMT model, and T_{DGT} denotes the optimized average travel time for the DGT model.

During the Morning Peak period, the DGT model achieves a 20.0% improvement in travel time compared to 16.7% for the GT-MMT model, indicating its superior performance in handling high-demand scenarios with its decentralized decision-making approach. In the Midday period, when traffic is generally lighter, both models perform well, but the DGT model still outperforms the GT-MMT model with a 21.4% improvement compared to 16.7%. This shows the DGT model's adaptability to fluctuating demand throughout the day. During the Evening Peak period, which typically involves heavy traffic, the DGT model again outperforms the GT-MMT model with a 21.4% improvement versus 17.8%, demonstrating its capability to dynamically adjust strategies based on real-time conditions. Overall, the DGT model consistently shows better travel time improvements across all periods, underscoring its advantage in dynamic and decentralized optimization.

b. Cost Savings Comparison Table Across Different Periods

Table 6.7: Cost Savings Comparison Across Different Periods

Period	**Model**	**Morning Peak ($)**	**Midday ($)**	**Evening Peak ($)**	**Percentage Cost Reduction (%)**
Morning Peak	GT-MMT DGT	12.00 11.00	9.50 8.50	14.00 12.50	20.0 22.7
Midday	GT-MMT DGT	8.50 8.00	7.00 6.00	9.50 8.50	17.6 25.0
Evening Peak	GT-MMT DGT	13.50 12.00	10.50 9.00	15.00 13.00	22.2 25.9

The percentage cost reduction for each period is determined using the following formula:

$$\text{Percentage Cost Reduction (\%)} = \left(\frac{C_{\text{GT-MMT}} - C_{\text{DGT}}}{C_{\text{GT-MMT}}} \right) \times 100 \tag{6.78}$$

Here, $C_{\text{GT-MMT}}$ is the optimized average cost for the GT-MMT model, while C_{DGT} represents the optimized average cost for the DGT model.

In the Morning Peak period, the DGT model achieves a 22.7% cost reduction compared to 20.0% for the GT-MMT model, highlighting its effectiveness in reducing costs during high-demand periods through localized decision-making. During the Midday period, the DGT model's cost reduction is 25.0%, significantly higher than the GT-MMT model's 17.6%, demonstrating its flexibility and efficiency in optimizing costs when demand is more variable. In the Evening Peak period, the DGT model again outperforms the GT-MMT model with a 25.9% cost reduction compared to 22.2%, reflecting its ability to better manage resources and minimize costs in high traffic conditions. Overall, the DGT model shows consistently higher cost savings across all periods, suggesting that its decentralized, community-based approach is more effective in managing transportation costs dynamically.

Conclusion

From the comparison of travel time and cost savings across different periods, the DGT model consistently outperforms the GT-MMT model. The decentralized decision-making and real-time adjustments in the DGT model provide greater flexibility and efficiency, leading to better optimization outcomes in various traffic and demand conditions. This indicates that the DGT model could be more suitable for complex urban environments where adaptability and localized decision-making are critical for efficient transportation management.

c. Travel Time Comparison between the GT-MMT and Different Communities

Table 6.8: Travel Time Comparison Between GT-MMT and DGT

Community	**Optimized Avg. Travel Time (GT-MMT) (min)**	**Optimized Avg. Travel Time (DGT) (min)**	**Improvement of DGT over GT-MMT (min)**	**Percentage Improvement of DGT over GT-MMT (%)**
Community 1	36.0	30.0	6.0	16.7
Community 2	42.0	35.0	7.0	16.7
Community 3	34.0	28.0	6.0	17.6
Community 4	48.0	38.5	9.5	19.8
Community 5	41.0	34.0	7.0	17.1

The improvement of DGT over GT-MMT in minutes is calculated by subtracting the optimized average travel time of the DGT model from that of the GT-MMT model:

$$\text{Improvement of DGT over GT-MMT (min)} = T_{\text{GT-MMT}} - T_{\text{DGT}} \tag{6.79}$$

The percentage improvement of DGT over GT-MMT is calculated using the formula:

$$\text{Percentage Improvement of DGT over GT-MMT} = \left(\frac{T_{\text{GT-MMT}} - T_{\text{DGT}}}{T_{\text{GT-MMT}}}\right) \times 100 \tag{6.80}$$

For example, in Community 1, the optimized average travel time using the GT-MMT model is 36.0 minutes, whereas the optimized average travel time using the DGT model is 30.0 minutes. The improvement of the DGT over the GT-MMT is 6.0 minutes, calculated as $36.0 - 30.0$. The percentage improvement is 16.7%, calculated as $\left(\frac{36.0-30.0}{36.0}\right) \times 100$.

This table directly compares the optimized average travel times achieved by the GT-MMT and DGT models. It highlights the absolute and percentage improvements of the DGT model over the GT-MMT model for each community.

The observations indicate that the DGT model shows consistent improvements over the GT-MMT model in reducing travel time across all communities. For example, in Community 1, the DGT model improves travel time by 6.0 minutes, which is a 16.7% improvement over the GT-MMT model. The conclusion drawn from this data is that the decentralized optimization of the DGT model offers better travel time reductions than the centralized GT-MMT model, reflecting the benefits of localized decision-making and dynamic adjustments.

d. Cost Savings Comparison between the GT-MMT and Different Communities

Table 6.9: Cost Savings Comparison Between GT-MMT and DGT

Community	**Optimized Avg. Cost (GT-MMT) ($)**	**Optimized Avg. Cost (DGT) ($)**	**Cost Reduction of DGT over GT-MMT ($)**	**Percentage Cost Reduction of DGT over GT-MMT (%)**
Community 1	12.00	10.50	1.50	12.5
Community 2	15.50	12.60	2.90	18.7
Community 3	14.50	11.55	2.95	20.3
Community 4	17.50	14.00	3.50	20.0
Community 5	12.50	9.80	2.70	21.6

The cost reduction of DGT over GT-MMT in dollars is calculated by subtracting the optimized average cost of the DGT model from that of the GT-MMT model:

$$\text{Cost Reduction of DGT over GT-MMT} = C_{\text{GT-MMT}} - C_{\text{DGT}} \tag{6.81}$$

The percentage cost reduction of DGT over GT-MMT is calculated using the formula:

$$\text{Percentage Cost Reduction of DGT over GT-MMT} = \left(\frac{C_{\text{GT-MMT}} - C_{\text{DGT}}}{C_{\text{GT-MMT}}}\right) \times 100 \tag{6.82}$$

For example, in Community 1, the optimized average cost using the GT-MMT model is \$12.00, whereas the optimized average cost using the DGT model is \$10.50. The cost reduction of the DGT over the GT-MMT is \$1.50, calculated as $12.00 - 10.50$. The percentage cost reduction is 12.5%, calculated as $\left(\frac{12.00-10.50}{12.00}\right) \times 100$.

This table directly compares the optimized average costs achieved by the GT-MMT and DGT models. It highlights the absolute and percentage cost reductions achieved by the DGT model over the GT-MMT model for each community.

The observations reveal that the DGT model consistently achieves cost reductions over the GT-MMT model across all communities. For instance, in Community 1, the DGT model reduces costs by \$1.50, which is a 12.5% reduction compared to the GT-MMT model. The conclusion drawn is that the community-based DGT model is more effective in reducing costs compared to the centralized GT-MMT model. This is likely due to the DGT model's ability to tailor strategies to local conditions, achieving more efficient resource allocation and cost savings.

Chapter 7 Conclusion and Future Work

7.1 Concluding Remarks

In this thesis, we design three dynamic multi-mode transportation models: DMT, GT-MMT, and DGT. The main concept of the three models—dynamic mobility traffic (DMT), game theory multi-mode transportation (GT-MMT), and decentralized game-theoretic (DGT)—is to enhance urban transportation efficiency by integrating multiple transportation modes and utilizing advanced optimization algorithms. The dynamic transportation models provide the much-needed flexibility and efficiency required to meet the demands of modern, fast-paced urban environments. They improve the capacity to manage networks, enhance the user experience, optimize resource utilization, and contribute to creating sustainable and resilient urban transport models. Multi-mode transportation schemes provide efficiency, environmental sustainability, and improved urban mobility advantages. Despite their complexity and implementation challenges, planners and policymakers must comprehend the dynamics, benefits, and obstacles associated with these models to enhance city transportation networks.

We develop a dynamic mobility traffic (DMT) model to enable passengers to use car ride-sharing and public buses either in combination or alone during their trip with the fastest time and lowest fare. At each station, passengers can be informed of the capacity of the car ride-share or public bus that will pass through the station, as well as the current position of each car or bus. The results indicate that the average stress levels from using car ride-sharing or public buses are similar. This stress level is higher than the stress level of using cars and buses in the near-optimal combination. In addition, the results highlight that passengers experience less satisfaction when only using public buses. Likewise, the satisfaction rate of passengers increases when using the appropriate combination of a public bus and car ride-share for transportation, thereby achieving lower costs and faster arrival. Furthermore, DMT is compared with a multi-load model in [140]. This comparison shows that our DMT is better in terms of lower fares and faster arrival times than a multi-load model.

The second model in this thesis is the GT-MMT model, which is based on the Stacklberg game theory. We allow the traveller to enjoy their trip by using a combination of bus, car ride-sharing, and bicycle. The GT-MMT model integrates advanced optimization techniques, Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Stackelberg game theory. The selection of the best trip combination is based on the Stackelberg equilibrium game theory model, where a leader defines the rules of the game while the rest of the players follow the leader. In this research, the passenger is the leader, and the rest of the vehicles follow the leader to define the strategies and utility functions. The utility function varies from player to player within the scope of the same game theory. While the passenger aims to reach the destination in the lowest possible time and for the lowest price depending on the type of vehicle they prefer to ride in, for the bus and the car, the goal is to try to have the seats vacant as often as possible, while for the bike, the goal is to maximize usage. Simulation results show that the GT-MMT gives riders a near-optimal trip at a lower cost and faster time, increasing user satisfaction regarding city transportation services.

The third model is called the DGT algorithm, as presented in this thesis. The decentralized game-theoretic (DGT) model offers a transformative approach to urban transportation management by leveraging community-based decision-making and fog computing. Unlike centralized models, the DGT model distributes computational power and decision-making across multiple localized fog nodes, enabling real-time, context-sensitive optimizations. Through a network of interconnected communities, each equipped with its own local strategies and objectives, the DGT model effectively balances global coordination with local autonomy. The DGT model integrates advanced optimization techniques, Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and decentralized Stackelberg game theory. It enhances the efficiency and responsiveness of transportation services. This decentralized framework not only improves travel times and reduces costs but also increases system resilience and adaptability to local conditions, making it a robust solution for modern urban mobility challenges. The "Multiple Concurrent Stackelberg Games" approach best embodies the principles of the DGT model by enabling localized, decentralized decision-making, improving flexibility, and enhancing overall system efficiency. This approach ensures that the transportation system can respond dynamically to varying local conditions and needs, making it the most suitable application of the Stackelberg concept in a community-based model like DGT.

This thesis uses the tools OMNET++, Inet, Veins, and OpenStreetMap modules to build an integrated simulation system. Using OpenStreetMap, part of a map of the City of Ottawa is taken as a case study. This simulation helps study the behaviour of the proposed model during the day in the City of Ottawa. OMNeT++ is a discrete event simulation environment primarily used for building network simulators. While it is commonly associated with networks, OMNeT++ is actually generic enough to be used for any discrete event simulation. This makes it versatile for various simulation needs, including those outside of networking. OpenStreetMap (OSM) is a collaborative project that creates a free editable map of the world, built by a community of mappers who contribute and maintain data about roads, trails, cafes, railway stations, and much more worldwide. The Veins framework provides a comprehensive and rich platform for conducting experiments that study the interplay between vehicular movement and network performance, proving invaluable for vehicular communication systems research. The INET framework is a crucial simulation library for the OMNeT++ simulation environment, extensively used for modelling and simulating various communication network protocols. INET serves as a core framework for many specialized simulation models and is actively developed and maintained with a large community of users and contributors.

The three proposed models can be easily applied to cities other than Ottawa because the equations take into account the different traffic conditions and designs of roads within cities, which are generated by a SUMO. Indeed, the model is suitable for short trips (rather than long trips) within a small city area.

7.2 Future Work

Based on the studies and conclusions in this thesis, several areas could be considered for future research. These are summarized as follows:

1. Improve the three models by considering user experience to improve the transportation service further and make mobility in the city more efficient, convenient, and enjoyable.
2. Improve the proposed algorithms, DGT and GT-MMT, by using data analytics technology in the infrastructure for the model. This will help to manage the proposed model better and facilitate more transparent and secure communication between players.

3. Extend future work by considering more transportation forms, such as trains. Also, electrified transportation can be an open issue for our work. We plan to study the privacy protection of our system regarding location-based services (LBS), especially for reviewing the publication of users and for tracking transportation forms' states.

4. A potential future enhancement for the GT-MMT model is developing an adaptive Stackelberg game framework incorporating learning mechanisms. This would involve using reinforcement learning algorithms to allow both the leader (passenger) and followers (transportation modes) to continuously learn and adapt their strategies based on observed behaviour and outcomes.

5. Integration of electric and autonomous vehicles into the GT-MMT Framework. This integration would require modifying the existing game-theoretic framework to account for these vehicles' unique operational characteristics and constraints, such as battery life management for electric vehicles and route optimization for autonomous vehicles. This would enhance the model's relevance in modern smart cities.

6. Enhances the inter-community coordination through multi-agent systems in the DGT model. By modelling each community as an intelligent agent capable of negotiation and cooperation, the DGT model can enhance its ability to manage inter-community transportation flows and resource allocation more efficiently, especially in densely populated urban areas.

References

[1] M. Bin Hariz, D. Said, and H. T. Mouftah, "Game theoretic approach for public multi-mode transportation in smart cities," *IET Networks*, vol. 10, no. 5, pp. 201–216, 2021.

[2] C.-H. Lin, M.-K. Jiau, and S.-C. Huang, "A cloud computing framework for real-time carpooling services," in *2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012)*. IEEE, 2012, pp. 266–271.

[3] N. Dimokas, K. Kalogirou, P. Spanidis, and D. Kehagias, "A mobile application for multimodal trip planning," in *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 2018, pp. 1–8.

[4] M. Oszczypała, J. Ziółkowski, J. Małachowski, and A. Legas, "Nash equilibrium and stackelberg approach for traffic flow optimization in road transportation networks—a case study of warsaw," *Applied Sciences*, vol. 13, no. 5, p. 3085, 2023.

[5] B. Wei, M. Saberi, F. Zhang, W. Liu, and S. T. Waller, "Modeling and managing ridesharing in a multi-modal network with an aggregate traffic representation: A doubly dynamical approach," *Transportation Research Part C: Emerging Technologies*, vol. 117, p. 102670, 2020.

[6] R. Scarinci, I. Markov, and M. Bierlaire, "Network design of a transport system based on accelerating moving walkways," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 310–328, 2017.

[7] N. Steurer and D. Bonilla, "Building sustainable transport futures for the mexico city metropolitan area," *Transport Policy*, vol. 52, pp. 121–133, 2016.

[8] J. Mercier, M. Carrier, F. Duarte, and F. Tremblay-Racicot, "Policy tools for sustainable transport in three cities of the americas: Seattle, montreal and curitiba," *Transport Policy*, vol. 50, pp. 95–105, 2016.

[9] P. Van Den Bossche, G. Maggetto, and M. Liccardo, "Citelec—electric vehicles on the move in europe's cities," *Journal of power sources*, vol. 40, no. 1-2, pp. 17–2, 1992.

[10] B. T. Morris, C. Tran, G. Scora, M. M. Trivedi, and M. J. Barth, "Real-time video-based traffic measurement and visualization system for energy/emissions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1667–1678, 2012.

[11] V. Milanés, J. Godoy, J. Villagrá, and J. Pérez, "Automated on-ramp merging system for congested traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 500–508, 2010.

[12] D. Çelebi, A. Yörüsün, and H. Işık, "Bicycle sharing system design with capacity allocations," *Transportation research part B: methodological*, vol. 114, pp. 86–98, 2018.

[13] M. J. Kalajahi, L. Khazini, Y. Rashidi, and S. Z. Heris, "Development of reduction scenarios based on urban emission estimation and dispersion of exhaust pollutants from light duty public transport: Case of tabriz, iran," *Emission Control Science and Technology*, vol. 6, no. 1, pp. 86–104, 2020.

[14] A. Ghadge, H. Wurtmann, and S. Seuring, "Managing climate change risks in global supply chains: a review and research agenda," *International Journal of Production Research*, vol. 58, no. 1, pp. 44–64, 2020.

[15] M. Ashifuddin Mondal and Z. Rehena, "Intelligent traffic congestion classification system using artificial neural network," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 110–116.

[16] T.-Y. Chen and R.-C. Jou, "Using hlm to investigate the relationship between traffic accident risk of private vehicles and public transportation," *Transportation Research Part A: Policy and Practice*, vol. 119, pp. 148–161, 2019.

[17] P. Maheshwari, P. Kachroo, A. Paz, and R. Khaddar, "Development of control models for the planning of sustainable transportation systems," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 474–485, 2015.

[18] S. Ionita, "Autonomous vehicles: from paradigms to technology," in *IOP Conference Series: Materials Science and Engineering*, vol. 252, no. 1. IOP Publishing, 2017, p. 012098.

[19] V. Albino, U. Berardi, and R. M. Dangelico, "Smart cities: Definitions, dimensions, performance, and initiatives," *Journal of urban technology*, vol. 22, no. 1, pp. 3–21, 2015.

[20] A. Goralzik, A. König, L. Alčiauskaitė, and T. Hatzakis, "Shared mobility services: an accessibility assessment from the perspective of people with disabilities," *European transport research review*, vol. 14, no. 1, pp. 1–12, 2022.

[21] S.-C. Huang, M.-K. Jiau, and C.-H. Lin, "A genetic-algorithm-based approach to solve carpool service problems in cloud computing," *IEEE Transactions on intelligent transportation systems*, vol. 16, no. 1, pp. 352–364, 2014.

[22] X. Hu, W. Xiong, W. Li, and L. Ke, "Application scenarios of wireless sensor networks for urban transportation: A survey," in *2015 34th Chinese Control Conference (CCC)*. IEEE, 2015, pp. 7688–7691.

[23] A. Muñoz-Villamizar, J. R. Montoya-Torres, and J. Faulin, "Impact of the use of electric vehicles in collaborative urban transport networks: A case study," *Transportation Research Part D: Transport and Environment*, vol. 50, pp. 40–54, 2017.

[24] R. Ceccato and M. Diana, "Substitution and complementarity patterns between traditional transport means and car sharing: a person and trip level analysis," *Transportation*, vol. 48, no. 4, pp. 1523–1540, 2021.

[25] X. Hu, N. K. Giang, J. Shen, V. C. Leung, and X. Li, "Towards mobility-as-a-service to promote smart transportation," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. IEEE, 2015, pp. 1–5.

[26] T. Röhr and M. Rovigo, "Public service approach to car-sharing in mid-sized towns: the example of belfort (france)," *IET Intelligent Transport Systems*, vol. 11, no. 7, pp. 403–410, 2017.

[27] P.-C. Chen, H.-Y. Hsieh, K.-W. Su, X. K. Sigalingging, Y.-R. Chen, and J.-S. Leu, "Predicting station level demand in a bike-sharing system using recurrent neural networks," *IET Intelligent Transport Systems*, vol. 14, no. 6, pp. 554–561, 2020.

[28] M. C. Compagnucci, T. Kono, and S. Teramoto, "Legal aspects of decentralized and platform-driven economies," in *Legal tech and the new sharing economy*. Springer, 2020, pp. 1–11.

[29] J. Hamari, M. Sjöklint, and A. Ukkonen, "The sharing economy: Why people participate in collaborative consumption," *Journal of the association for information science and technology*, vol. 67, no. 9, pp. 2047–2059, 2016.

[30] S. Shaheen and N. Chan, "Mobility and the sharing economy: Potential to facilitate the first-and last-mile public transit connections," *Built Environment*, vol. 42, no. 4, pp. 573–588, 2016.

[31] J. Clark and A. Curl, "Bicycle and car share schemes as inclusive modes of travel? a socio-spatial analysis in glasgow," *Social Inclusion*, vol. 4, no. 3, pp. 83–99, 2016.

[32] P. Minett and J. Pearce, "Estimating the energy consumption impact of casual carpooling," *Energies*, vol. 4, no. 1, pp. 126–139, 2011.

[33] F. Roli, "Clacsoon: Carpooling in urban areas."

[34] M. Barth and M. Todd, "Simulation model performance analysis of a multiple station shared vehicle system," *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 4, pp. 237–259, 1999.

[35] S. Galland, L. Knapen, N. Gaud, D. Janssens, O. Lamotte, A. Koukam, G. Wets *et al.*, "Multi-agent simulation of individual mobility behavior in carpooling," *Transportation Research Part C: Emerging Technologies*, vol. 45, pp. 83–98, 2014.

[36] A. Najmi, T. H. Rashidi, and T. Waller, "A multimodal multi-provider market equilibrium model: A game-theoretic approach," *Transportation Research Part C: Emerging Technologies*, vol. 146, p. 103959, 2023.

[37] H. Huang, D. Bucher, J. Kissling, R. Weibel, and M. Raubal, "Multimodal route planning with public transport and carpooling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3513–3525, 2018.

[38] B. Sanjukta, B. Sourav, and C. Chinmay, "Iot-based smart transportation system under real-time environment. iet: big data-enabled internet of things: challenges and opportunities (ch. 16)," ISBN 978-1-78561-637-2, Tech. Rep., 2019.

[39] R. Kager, L. Bertolini, and M. Te Brömmelstroet, "Characterisation of and reflections on the synergy of bicycles and public transport," *Transportation Research Part A: Policy and Practice*, vol. 85, pp. 208–219, 2016.

[40] L. Chunlin, L. Yanpei, L. Youlong, and Z. Min, "Collaborative content dissemination based on game theory in multimedia cloud," *Knowledge-Based Systems*, vol. 124, pp. 1–15, 2017.

[41] R. Latha, P. Vetrivelan, and M. Jagannath, "Balancing emergency message dissemination and network lifetime in wireless body area network using ant colony optimization and bayesian game formulation," *Informatics in Medicine Unlocked*, vol. 8, pp. 60–65, 2017.

[42] H. Shayanfar, A. S. Lahiji, J. Aghaei, and A. Rabiee, "Generation expansion planning in pool market: A hybrid modified game theory and improved genetic algorithm," *Energy Conversion and Management*, vol. 50, no. 5, pp. 1149–1156, 2009.

[43] E. S. Okpoti, I.-J. Jeong, and S. K. Moon, "Decentralized determination of design variables among cooperative designers for product platform design in a product family," *Computers & Industrial Engineering*, vol. 135, pp. 601–614, 2019.

[44] R. Palma-Behnke, G. Jiménez-Estévez, L. S. Vargas, E. Handschin, F. Uphaus, and E. Hauptmeier, "A day-ahead energy market simulation framework for assessing the impact of decentralized generators on step-down transformer power flows," *International Journal Of Electrical Power & Energy Systems*, vol. 35, no. 1, pp. 10–20, 2012.

[45] D. Paret and J.-P. Huon, *Secure connected objects.* John Wiley & Sons, 2017.

[46] M. B. Hariz, D. Said, and H. T. Mouftah, "Mobility traffic model based on combination of multiple transportation forms in the smart city," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC).* IEEE, 2019, pp. 14–19.

[47] M. Bin Hariz, D. Said, and H. T. Mouftah, "A dynamic mobility traffic model based on two modes of transport in smart cities," *Smart Cities*, vol. 4, no. 1, pp. 253–270, 2021.

[48] M. B. Hariz, D. Said, and H. T. Mouftah, "Game theoretic approach for a multi-mode transportation in smart cities," in *2020 International Symposium on Networks, Computers and Communications (ISNCC).* IEEE, 2020, pp. 1–6.

[49] M. Bin Hariz, D. Said, and H. T. Mouftah, "Decentralised game-theoretic management for a community-based transportation system," *IET Smart Cities*, vol. 2, no. 4, pp. 181–190, 2020.

[50] M. Stiglic, N. Agatz, M. Savelsbergh, and M. Gradisar, "Enhancing urban mobility: Integrating ride-sharing and public transit," *Computers & Operations Research*, vol. 90, pp. 12–21, 2018.

[51] M. Müller-Hannemann, F. Schulz, D. Wagner, and C. Zaroliagis, "Timetable information: Models and algorithms," in *Algorithmic Methods for Railway Optimization.* Springer, 2007, pp. 67–90.

[52] T. Pajor, "Multi-modal route planning," *Universität Karlsruhe*, 2009.

[53] M. Furuhata, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transportation Research Part B: Methodological*, vol. 57, pp. 28–46, 2013.

[54] N. Agatz, A. L. Erera, M. W. Savelsbergh, and X. Wang, "Dynamic ride-sharing: A simulation study in metro atlanta," *Procedia-Social and Behavioral Sciences*, vol. 17, pp. 532–550, 2011.

[55] D. Alipour and H. Dia, "A systematic review of the role of land use, transport, and energy-environment integration in shaping sustainable cities," *Sustainability*, vol. 15, no. 8, p. 6447, 2023.

[56] I. Appaji and P. Raviraj, "Effectiveness of recent methodologies of intelligent transportation system," *International Journal of Intelligent Transportation Systems Research*, pp. 1–10, 2023.

[57] Y. Wang, X. Lin, F. He, and M. Li, "Designing transit-oriented multi-modal transportation systems considering travelers' choices," *Transportation Research Part B: Methodological*, vol. 162, pp. 292–327, 2022.

[58] C. Filippi, G. Guastaroba, L. Peirano, and M. G. Speranza, "Trends in passenger transport optimisation," *International Transactions in Operational Research*, 2023.

[59] G. Gabdullina, M. Gilmanov, A. Akhmetgareeva, S. Khusainova, M. Khamidullin, and G. Gareeva, "Assessment of the resource utilization efficiency of transport companies," *Transportation Research Procedia*, vol. 63, pp. 1055–1060, 2022.

[60] G. G. de Oliveira, Y. Iano, G. C. Vaz, and K. Suriyan, "Data collection and analysis applied to intelligent transportation systems: a case study on public transportation," *Discover Artificial Intelligence*, vol. 3, no. 1, p. 13, 2023.

[61] S. Fan and Z. Yang, "Safety and security co-analysis in transport systems: Current state and regulatory development," *Transportation research part A: policy and practice*, vol. 166, pp. 369–388, 2022.

[62] M. Kolat, B. Kővári, T. Bécsi, and S. Aradi, "Multi-agent reinforcement learning for traffic signal control: A cooperative approach," *Sustainability*, vol. 15, no. 4, p. 3479, 2023.

[63] L. Schlicher and V. Lurkin, "Stable allocations for choice-based collaborative price setting," *European Journal of Operational Research*, vol. 302, no. 3, pp. 1242–1254, 2022.

[64] F. Ahmad, Z. Shah, and L. Al-Fagih, "Applications of evolutionary game theory in urban road transport network: A state of the art review," *Sustainable Cities and Society*, p. 104791, 2023.

[65] Y. Yan, L. Peng, T. Shen, J. Wang, D. Pi, D. Cao, and G. Yin, "A multi-vehicle game-theoretic framework for decision making and planning of autonomous vehicles in mixed traffic," *IEEE Transactions on Intelligent Vehicles*, 2023.

[66] T. Harks and J. Schwarz, "A unified framework for pricing in nonconvex resource allocation games," *SIAM Journal on Optimization*, vol. 33, no. 2, pp. 1223–1249, 2023.

[67] F. Ahmad and L. Al-Fagih, "Game theory applications in micro and macroscopic simulation in transportation network: A comprehensive review," *IEEE Access*, 2023.

[68] L. Wang, K. Pan, Q. Zhao, L. Zhang, and L. Zhang, "Research on queue equilibrium control algorithm of urban traffic based on game theory," *Applied Sciences*, vol. 13, no. 3, p. 1781, 2023.

[69] G. Zardini, N. Lanzetti, G. Belgioioso, C. Hartnik, S. Bolognani, F. Dörfler, and E. Frazzoli, "Strategic interactions in multi-modal mobility systems: A game-theoretic perspective," *arXiv preprint arXiv:2308.04820*, 2023.

[70] H. Wang, X. Li, L. Yu, and Y. Jiang, "Game analysis on urban traffic congestion charging," in *Proceedings of the Sixth International Forum on Decision Sciences*. Springer, 2020, pp. 219–227.

[71] J. Jiang, F. Peng, and B. Zhao, "Mutual exclusion algorithm for real-time traffic dispatching based on game theory," in *IOP Conference Series: Materials Science and Engineering*, vol. 782, no. 3. IOP Publishing, 2020, p. 032058.

[72] L.-m. Liu and Y.-b. Lv, "Research on dynamic adjustment strategy of metro fare based on game theory," in *CICTP 2015*, 2015, pp. 3798–3805.

[73] S. Sharma, S. Bhattacharya, D. Kiran, B. Hu, M. Prandtstetter, and B. Azzopardi, "Optimizing the scheduling of electrified public transport system in malta," *Energies*, vol. 16, no. 13, p. 5073, 2023.

[74] H.-i. Cheong, J. J. E. Macias, R. Karamanis, M. Stettler, A. Majumdar, and P. Angeloudis, "Policy and strategy evaluation of ridesharing autonomous vehicle operation: A london case study," *Transportation Research Record*, p. 03611981231160157, 2023.

[75] P. Raimondo, A. Serianni, N. Palmieri, and G. Potrino, "Improving intelligent transportation system (its) introducing a fog cooperative strategy," in *2018 26th Telecommunications Forum (TELFOR)*. IEEE, 2018, pp. 1–4.

[76] A. F. Santamaria, M. Tropea, P. Fazio, P. Raimondo, F. De Rango, and M. Voznak, "A decentralized its architecture for efficient distribution of traffic task management," in *2018 11th IFIP wireless and mobile networking conference (WMNC)*. IEEE, 2018, pp. 1–5.

[77] S. Di Martino, R. Galiero, C. Giorio, F. Ferrucci, and F. Sarro, "A matching-algorithm based on the cloud and positioning systems to improve carpooling." in *DMS*. Citeseer, 2011, pp. 90–95.

[78] Y.-H. Kuo, J. M. Leung, and Y. Yan, "Public transport for smart cities: Recent innovations and future challenges," *European Journal of Operational Research*, vol. 306, no. 3, pp. 1001–1026, 2023.

[79] O. Linovski and D. M. Baker, "Community-designed participation: Lessons for equitable engagement in transportation planning," *Transportation Research Record*, p. 03611981221145131, 2023.

[80] R. Sun, F. Xie, S. Huang, and Y. Shao, "Construction and characteristics analysis of the xi'an public transport network considering single-mode and multi-mode transferring," *Sustainability*, vol. 16, no. 9, p. 3846, 2024.

[81] Q. Han, H. Wang, and J. Wang, "Multi-mode/signal biosensors: Electrochemical integrated sensing techniques," *Advanced Functional Materials*, p. 2403122, 2024.

[82] R. Geisberger, D. Luxen, S. Neubauer, P. Sanders, and L. Volker, "Fast detour computation for ride sharing," *arXiv preprint arXiv:0907.5269*, 2009.

[83] M. Zhu, X.-Y. Liu, and X. Wang, "An online ride-sharing path-planning strategy for public vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 616–627, 2018.

[84] S.-C. Huang, M.-K. Jiau, and Y.-P. Liu, "An ant path-oriented carpooling allocation approach to optimize the carpool service problem with time windows," *IEEE Systems Journal*, vol. 13, no. 1, pp. 994–1005, 2018.

[85] S. Kazi, M. Bagasrawala, F. Shaikh, and A. Sayyed, "Smart e-ticketing system for public transport bus," in *2018 International Conference on Smart City and Emerging Technology (ICSCET)*. IEEE, 2018, pp. 1–7.

[86] V. Armant and K. N. Brown, "Minimizing the driving distance in ride sharing systems," in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*. IEEE, 2014, pp. 568–575.

[87] M. Zhu, X.-Y. Liu, F. Tang, M. Qiu, R. Shen, W. Shu, and M.-Y. Wu, "Public vehicles for future urban transportation," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 12, pp. 3344–3353, 2016.

[88] J. Jung, R. Jayakrishnan, and J. Y. Park, "Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing," *Computer-Aided Civil and Infrastructure Engineering*, vol. 31, no. 4, pp. 275–291, 2016.

[89] Y. Huang, R. Jin, F. Bastani, and X. S. Wang, "Large scale real-time ridesharing with service guarantee on road networks," *arXiv preprint arXiv:1302.6666*, 2013.

[90] R.-H. Li, L. Qin, J. X. Yu, and R. Mao, "Optimal multi-meeting-point route search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 770–784, 2015.

[91] H. Hosni, J. Naoum-Sawaya, and H. Artail, "The shared-taxi problem: Formulation and solution methods," *Transportation Research Part B: Methodological*, vol. 70, pp. 303–318, 2014.

[92] Q.-P. Gu, J. L. Liang, and G. Zhang, "Approximate ridesharing of personal vehicles problem," *Theoretical Computer Science*, vol. 871, pp. 30–50, 2021.

[93] L. Melis and K. Sörensen, "The real-time on-demand bus routing problem: The cost of dynamic requests," *Computers & Operations Research*, vol. 147, p. 105941, 2022.

[94] Y. Xu, Y. Tong, Y. Shi, Q. Tao, K. Xu, and W. Li, "An efficient insertion operator in dynamic ridesharing services," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[95] A. Bit-Monnot, C. Artigues, M.-J. Huguet, and M.-O. Killijian, "Carpooling: the 2 synchronization points shortest paths problem," in *13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems.* Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[96] X. Li, J. Gao, C. Wang, X. Huang, and Y. Nie, "Ride-sharing matching under travel time uncertainty through data-driven robust optimization," *IEEE Access*, vol. 10, pp. 116 931–116 941, 2022.

[97] Y. Bie, Y. Cong, M. Yang, and L. Wang, "Coordinated scheduling of electric buses for multiple routes considering stochastic travel times," *Journal of Transportation Engineering, Part A: Systems*, vol. 149, no. 8, p. 04023069, 2023.

[98] J. Dibbelt, T. Pajor, and D. Wagner, "User-constrained multimodal route planning," *Journal of Experimental Algorithmics (JEA)*, vol. 19, pp. 1–19, 2015.

[99] K. Aissat and S. Varone, "Carpooling as complement to multi-modal transportation," in *International Conference on Enterprise Information Systems*. Springer, 2015, pp. 236–255.

[100] S. Varone and K. Aissat, "Multi-modal transportation with public transport and ride-sharing," in *Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS 2015)*, 2015.

[101] S. Yan and C.-Y. Chen, "A model and a solution algorithm for the carpooling problem with pre-matching information," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 512–524, 2011.

[102] M. Mallus, G. Colistra, L. Atzori, M. Murroni, and V. Pilloni, "Dynamic carpooling in urban areas: design and experimentation with a multi-objective route matching algorith," *Sustainability*, vol. 9, no. 2, p. 254, 2017.

[103] R. Olszewski, P. Pałka, and A. Turek, "Solving "smart city" transport problems by designing carpooling gamification schemes with multi-agent systems: The case of the so-called "mordor of warsaw"," *Sensors*, vol. 18, no. 1, p. 141, 2018.

[104] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," in *Algorithm engineering*. Springer, 2016, pp. 19–80.

[105] T. Akiba, Y. Iwata, K.-i. Kawarabayashi, and Y. Kawata, "Fast shortest-path distance queries on road networks by pruned highway labeling," in *2014 Proceedings of the sixteenth workshop on algorithm engineering and experiments (ALENEX)*. SIAM, 2014, pp. 147–154.

[106] L. Allulli, G. F. Italiano, and F. Santaroni, "Exploiting gps data in public transport journey planners," in *International Symposium on Experimental Algorithms*. Springer, 2014, pp. 295–306.

[107] X. Yu, H. Miao, A. Bayram, M. Yu, and X. Chen, "Optimal routing of multimodal mobility systems with ride-sharing," *International Transactions in Operational Research*, vol. 28, no. 3, pp. 1164–1189, 2021.

[108] R. S. Thangaraj, K. Mukherjee, G. Raravi, A. Metrewar, N. Annamaneni, and K. Chattopadhyay, "Xhare-a-ride: A search optimized dynamic ride sharing system with approximation guarantee," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 2017, pp. 1117–1128.

[109] X. Pi, W. Ma, and Z. S. Qian, "A general formulation for multi-modal dynamic traffic assignment considering multi-class vehicles, public transit and parking," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 369–389, 2019.

[110] E. Cangialosi, A. Di Febbraro, and N. Sacco, "Designing a multimodal generalised ride sharing system," *IET Intelligent Transport Systems*, vol. 10, no. 4, pp. 227–236, 2016.

[111] A. Araldo, S. Gao, R. Seshadri, C. L. Azevedo, H. Ghafourian, Y. Sui, S. Ayaz, D. Sukhin, and M. Ben-Akiva, "System-level optimization of multi-modal transportation networks for energy efficiency using personalized incentives: Formulation, implementation, and performance," *Transportation Research Record*, vol. 2673, no. 12, pp. 425–438, 2019.

[112] A. Baggag, S. Abbar, T. Zanouda, and J. Srivastava, "Resilience analytics: coverage and robustness in multi-modal transportation networks," *EPJ Data Science*, vol. 7, pp. 1–21, 2018.

[113] D. Bucher, D. Jonietz, and M. Raubal, "A heuristic for multi-modal route planning," in *Progress in Location-Based Services 2016*. Springer, 2017, pp. 211–229.

[114] U. M. Aïvodji, S. Gambs, M.-J. Huguet, and M.-O. Killijian, "Meeting points in ridesharing: A privacy-preserving approach," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 239–253, 2016.

[115] P. Czioska, A. Trifunović, S. Dennisen, and M. Sester, "Location-and time-dependent meeting point recommendations for shared interurban rides," *Journal of Location Based Services*, vol. 11, no. 3-4, pp. 181–203, 2017.

[116] Y. Liu, P. Bansal, R. Daziano, and S. Samaranayake, "A framework to integrate mode choice in the design of mobility-on-demand systems," *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 648–665, 2019.

[117] P. Kumar and A. Khani, "An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102891, 2021.

[118] B. Zhou, Q. Song, Z. Zhao, and T. Liu, "A reinforcement learning scheme for the equilibrium of the in-vehicle route choice problem based on congestion game," *Applied Mathematics and Computation*, vol. 371, p. 124895, 2020.

[119] A. B. Sherif, K. Rabieh, M. M. Mahmoud, and X. Liang, "Privacy-preserving ride sharing scheme for autonomous vehicles in big data era," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 611–618, 2016.

[120] J. Zhang, D. Wen, and S. Zeng, "A discounted trade reduction mechanism for dynamic ridesharing pricing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1586–1595, 2015.

[121] S. Banerjee, D. Freund, and T. Lykouris, "Pricing and optimization in shared vehicle systems: An approximation framework," *Operations Research*, vol. 70, no. 3, pp. 1783–1805, 2022.

[122] Z. Bian, X. Liu, and Y. Bai, "Mechanism design for on-demand first-mile ridesharing," *Transportation research part B: methodological*, vol. 138, pp. 77–117, 2020.

[123] H. Billhardt, A. Fernández, and S. Ossowski, "Smart recommendations for renting bikes in bike-sharing systems," *Applied Sciences*, vol. 11, no. 20, p. 9654, 2021.

[124] H. Singh, M. Kavianipour, M. Ghamami, and A. Zockaie, "Adoption of autonomous and electric vehicles in private and shared mobility systems," *Transportation research part D: transport and environment*, vol. 115, p. 103561, 2023.

[125] Y. Li, X. Li, and S. Zhang, "Optimal pricing of customized bus services and ride-sharing based on a competitive game model," *Omega*, vol. 103, p. 102413, 2021.

[126] J. Zhang, M. Meng, and Z. David, "A dynamic pricing scheme with negative prices in dockless bike sharing systems," *Transportation Research Part B: Methodological*, vol. 127, pp. 201–224, 2019.

[127] M. Li, G. Hua, and H. Huang, "A multi-modal route choice model with ridesharing and public transit," *Sustainability*, vol. 10, no. 11, p. 4275, 2018.

[128] R. Auad, K. Dalmeijer, C. Riley, T. Santanam, A. Trasatti, P. Van Hentenryck, and H. Zhang, "Resiliency of on-demand multimodal transit systems during a pandemic," *Transportation Research Part C: Emerging Technologies*, vol. 133, p. 103418, 2021.

[129] M. Chaturvedi and S. Srivastava, "A multi-modal ride sharing framework for last mile connectivity," in *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2022, pp. 824–829.

[130] L. Balzer, M. Ameli, L. Leclercq, and J.-P. Lebacque, "Dynamic tradable credit scheme for multimodal urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 149, p. 104061, 2023.

[131] L. Piao, Q. Ai, and S. Fan, "Game theoretic based pricing strategy for electric vehicle charging stations," 2015.

[132] S. Li, F. Fei, D. Ruihan, S. Yu, and W. Dou, "A dynamic pricing method for carpooling service based on coalitional game analysis," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2016, pp. 78–85.

[133] E. Magsino, G. R. Ching, F. M. Espiritu, and K. Go, "Evaluating stable matching methods and ridesharing techniques in optimizing passenger transportation cost and companionship," *ISPRS International Journal of Geo-Information*, vol. 11, no. 11, p. 556, 2022.

[134] B. Hu, M. Hu, and H. Zhu, "Surge pricing and two-sided temporal responses in ride hailing," *Manufacturing & Service Operations Management*, vol. 24, no. 1, pp. 91–109, 2022.

[135] H. Ren and R. Chen, "Porcelain supply chain coordination considering the preferences of consumers against the background of e-commerce," *Sustainability*, vol. 15, no. 13, p. 9877, 2023.

[136] S. Rahiminia, A. Mehrabi, M. Pourseyed Aghaee, and A. Jamili, "Adopting a bi-level optimization method for the freight transportation problem: A multi-objective programming approach," *Transportation Research Record*, vol. 2677, no. 2, pp. 490–504, 2023.

[137] E. R. Magsino, G. R. C. Ching, F. M. M. Espiritu, and K. D. Go, "A game theory-based pricing technique for ridesharing pairings," in *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*. IEEE, 2023, pp. 1–5.

[138] A. Najmi, T. Waller, and T. H. Rashidi, "Equity in network design and pricing: A discretely-constrained mpec problem," *Transportation Research Part A: Policy and Practice*, vol. 176, p. 103800, 2023.

[139] J. S. Arias-Rojas, J. F. Jiménez, and J. R. Montoya-Torres, "Solving of school bus routing problem by ant colony optimization," *Revista EIA*, no. 17, pp. 193–208, 2012.

[140] D. M. Miranda, R. S. de Camargo, S. V. Conceição, M. F. Porto, and N. T. Nunes, "A multi-loading school bus routing problem," *Expert Systems with Applications*, vol. 101, pp. 228–242, 2018.

[141] R. Massobrio, G. Fagúndez, and S. Nesmachnow, "Multiobjective evolutionary algorithms for the taxi sharing problem," *International Journal of Metaheuristics*, vol. 5, no. 1, pp. 67–90, 2016.

[142] A. Namoun, "Extracting the factor influencing the use of community-based traveler information systems by analysing commuting critical incidents," *J. Theor. Appl. Inf. Technol*, vol. 96, no. 24, pp. 8132–8144, 2018.

[143] J. Guo, J. Long, X. Xu, M. Yu, and K. Yuan, "The vehicle routing problem of intercity ride-sharing between two cities," *Transportation Research Part B: Methodological*, vol. 158, pp. 113–139, 2022.

[144] Z. Bian, Y. Bai, X. Liu, and B. Wang, "An online hybrid mechanism for dynamic first-mile ridesharing service," *Transportation Research Part C: Emerging Technologies*, vol. 138, p. 103585, 2022.

[145] S. Yoo, H. Kim, and J. Lee, "Combining reinforcement learning with genetic algorithm for many-to-many route optimization of autonomous vehicles," *IEEE Access*, vol. 12, pp. 26 931–26 942, 2024.

[146] M. Kang, S.-K. Kim, J. T. Felan, H. R. Choi, and M. Cho, "Development of a genetic algorithm for the school bus routing problem," *International Journal of Software Engineering and Its Applications*, vol. 9, no. 5, pp. 107–126, 2015.

[147] N. Ta, G. Li, T. Zhao, J. Feng, H. Ma, and Z. Gong, "An efficient ride-sharing framework for maximizing shared route," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 2, pp. 219–233, 2017.

[148] D. O. Santos and E. C. Xavier, "Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6728–6737, 2015.

[149] B. Cao, L. Alarabi, M. F. Mokbel, and A. Basalamah, "Sharek: A scalable dynamic ride sharing system," in *2015 16th IEEE International Conference on Mobile Data Management*, vol. 1. IEEE, 2015, pp. 4–13.

[150] S. Navathe, W. Wu, S. Shekhar, X. Du, S. Wang, and H. Xiong, *Database Systems for Advanced Applications.* Springer, 2016.

[151] S. Yeung, E. Miller, and S. Madria, "A flexible real-time ridesharing system considering current road conditions," in *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, vol. 1. IEEE, 2016, pp. 186–191.

[152] P. Cheng, H. Xin, and L. Chen, "Utility-aware ridesharing on road networks," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1197–1210.

[153] M. Nourinejad and M. J. Roorda, "Agent based model for dynamic ridesharing," *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 117–132, 2016.

[154] L. Duan, Y. Wei, J. Zhang, and Y. Xia, "Centralized and decentralized autonomous dispatching strategy for dynamic autonomous taxi operation in hybrid request mode," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 397–420, 2020.

[155] N. V. Bozdog, S. Voulgaris, H. Bal, and A. Van Halteren, "Peer matcher: Decentralized partnership formation," in *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems*. IEEE, 2015, pp. 31–40.

[156] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5872–5881.

[157] S. B. Cheikh, S. Hammadi, and C. Tahon, "Based-agent distributed architecture to manage the dynamic multi-hop ridesharing system," in *2014 IEEE 13th International Symposium on Network Computing and Applications*. IEEE, 2014, pp. 101–104.

[158] Y. Zhong, L. Gao, T. Wang, S. Gong, B. Zou, and D. Yu, "Achieving stable and optimal passenger-driver matching in ride-sharing system," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2018, pp. 125–133.

[159] N. V. Bozdog, M. X. Makkes, A. Van Halteren, and H. Bal, "Ridematcher: peer-to-peer matching of passengers for efficient ridesharing," in *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE, 2018, pp. 263–272.

[160] R. Nair and E. Miller-Hooks, "Equilibrium network design of shared-vehicle systems," *European Journal of Operational Research*, vol. 235, no. 1, pp. 47–61, 2014.

[161] D. Kim, J. Ko, and Y. Park, "Factors affecting electric vehicle sharing program participants' attitudes about car ownership and program participation," *Transportation Research Part D: Transport and Environment*, vol. 36, pp. 96–106, 2015.

[162] P.-C. Lin, C.-W. Shen, and J. Wang, "The bus-type taxi: a better demonstration of ride-sharing than uber taiwan," *Transportation Planning and Technology*, vol. 44, no. 7, pp. 714–725, 2021.

[163] X. Wang and R. Zhang, "Carpool services for ride-sharing platforms: Price and welfare implications," *Naval Research Logistics (NRL)*, vol. 69, no. 4, pp. 550–565, 2022.

[164] A. K. Agrawal, S. Yadav, A. A. Gupta, and S. Pandey, "A genetic algorithm model for optimizing vehicle routing problems with perishable products under time-window and quality requirements," *Decision Analytics Journal*, vol. 5, p. 100139, 2022.

[165] Y. Hikima, Y. Akagi, N. Marumo, and H. Kim, "Online matching with controllable rewards and arrival probabilities," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2022, pp. 1825–1833, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2022/254

[166] A. Fielbaum, X. Bai, and J. Alonso-Mora, "On-demand ridesharing with optimized pick-up and drop-off walking locations," *Transportation research part C: emerging technologies*, vol. 126, p. 103061, 2021.

[167] H. Wu, Y. Chen, L. Wang, and G. Ma, "E-ride: An adaptive event-driven windowed matching framework in ridesharing," *IEEE Access*, vol. 10, pp. 43 799–43 811, 2022.

[168] A. Fielbaum, "Optimizing a vehicle's route in an on-demand ridesharing system in which users might walk," *Journal of Intelligent Transportation Systems*, vol. 26, no. 4, pp. 432–447, 2022.

[169] X. Lin, Z. Liang, L. Shen, F. Zhao, X. Liu, P. Sun, and T. Cao, "Reinforcement learning method for the multi-objective speed trajectory optimization of a freight train," *Control Engineering Practice*, vol. 138, p. 105605, 2023.

[170] V. C. Karels, W. Rei, L. P. Veelenturf, and T. Van Woensel, "A vehicle routing problem with multiple service agreements," *European Journal of Operational Research*, vol. 313, no. 1, pp. 129–145, 2024.

[171] R. Narayanam and Y. Narahari, "A game theory inspired, decentralized, local information based algorithm for community detection in social graphs," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 1072–1075.

[172] L. Cipolina-Kun, V. Yazdanpanah, S. Stein, and E. H. Gerding, "A proportional pricing mechanism for ridesharing services with meeting points," in *PRIMA 2022: Principles and Practice of Multi-Agent Systems: 24th International Conference, Valencia, Spain, November 16–18, 2022, Proceedings*. Springer, 2022, pp. 523–539.

[173] U. Bhaskar, L. Fleischer, and E. Anshelevich, “A stackelberg strategy for routing flow over time,” *Games and Economic Behavior*, vol. 92, pp. 232–247, 2015.

[174] J. Gao and F. You, “A stochastic game theoretic framework for decentralized optimization of multi-stakeholder supply chains under uncertainty,” *Computers & Chemical Engineering*, vol. 122, pp. 31–46, 2019.

[175] X. Zhang, H. Zhang, H.-J. Huang, L. Sun, and T.-Q. Tang, “Competitive, cooperative and stackelberg congestion pricing for multiple regions in transportation networks,” *Transportmetrica*, vol. 7, no. 4, pp. 297–320, 2011.

[176] P.-y. Nie, L.-h. Chen, and M. Fukushima, “Dynamic programming approach to discrete time dynamic feedback stackelberg games with independent and dependent followers,” *European Journal of Operational Research*, vol. 169, no. 1, pp. 310–328, 2006.

[177] R. Nair and E. Miller-Hooks, “Equilibrium design of bicycle sharing systems: the case of washington dc,” *EURO Journal on Transportation and Logistics*, vol. 5, no. 3, pp. 321–344, 2016.

[178] Y. Dai, Y. Gao, H. Gao, and H. Zhu, “Real-time pricing scheme based on stackelberg game in smart grid with multiple power retailers,” *Neurocomputing*, vol. 260, pp. 149–156, 2017.

[179] C. Lei, Z. Jiang, and Y. Ouyang, “Path-based dynamic pricing for vehicle allocation in ridesharing systems with fully compliant drivers,” *Transportation research procedia*, vol. 38, pp. 77–97, 2019.

[180] D. B. Rokhlin and G. A. Ougolnitsky, “Stackelberg equilibrium in a dynamic stimulation model with complete information,” *Automation and Remote Control*, vol. 79, no. 4, pp. 701–712, 2018.

Appendix A-Confidence Intervals Calculation

There are several methods to compute the confidence intervals (CI) for the proposed models, and one of the most commonly used approaches is the quantitative technique.

In this method, n represents the number of simulation runs. The key metrics, such as average user satisfaction and average transportation stress level (in the DMT model), average trip time and average trip price variation (in the GT-MMT model), and normalized average trip time using different transportation modes (in the DGT model), are measured by averaging the results across n successive runs. Multiple simulations are conducted to ensure no correlations in the presented results.

The results from a series of runs are gathered to calculate the confidence interval, aiming for a 95% confidence level, with a reliability factor ($Z_{\text{value}} = 1.96$).

The mean value of the collected runs $R_1, R_2, \ldots, R_n$ is calculated using Equation A.1:

$$\bar{R} = \frac{1}{n} \sum_{i=1}^{n} R_i \tag{A.1}$$

The variance (V_r^2) is computed by Equation A.2:

$$V_r^2 = \frac{1}{n-1} \sum_{i=1}^{n} (R_i - \bar{R})^2 \tag{A.2}$$

The standard deviation (σ) is determined by Equation A.3:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (R_i - \bar{R})^2} \tag{A.3}$$

The confidence interval for the sample mean is calculated using Equation A.4:

$$CI = \bar{R} \pm Z_{\text{value}} \times \frac{\sigma}{\sqrt{n}} \tag{A.4}$$

Where $\bar{R}$ is the sample mean, n is the sample size (number of runs), and Z_{value} is the corresponding value for a 95% confidence level.

The upper and lower bounds of the 95% confidence interval are calculated using Equations A.5 and A.6:

$$U(CI) = \bar{R} + Z_{\text{value}} \times \frac{\sigma}{\sqrt{n}} \tag{A.5}$$

$$L(CI) = \bar{R} - Z_{\text{value}} \times \frac{\sigma}{\sqrt{n}} \tag{A.6}$$

The results collected are found to lie within the calculated 95% confidence interval, as determined by Equations A.5 and A.6. In our results, the confidence interval (CI) was found to be very small and negligible, indicating that the values are tightly clustered around the mean. As a result, we did not include the CI bars on the curve, as they would not provide any additional meaningful insight.