Towards cooperative urban traffic management:

Investigating voting for travel groups

Doctoral Thesis
(Dissertation)

to be awarded the degree

Doctor rerum naturalium (Dr. rer. nat.)

submitted by

Sophie Dennisen

from Dormagen

approved by the Faculty of Mathematics/Computer Science and Mechanical Engineering, Clausthal University of Technology,

Date of oral examination

22 July 2021

Dean

Prof. Dr. Jörg Müller

Chairperson of the Board of Examiners

Prof. Dr. Michael Prilla

Supervising tutor

Prof. Dr. Jörg Müller

Reviewer

Prof. Dr. Markus Fidler

Ire fortiter quo nullus viator ante iit

(Literally "Boldly go where no traveller has gone before"
- adapted from "Boldly go where no one has gone before" by Star Trek)

Acknowledgements

First, I would like to thank my colleagues from my home institute, in particular my supervisor Jörg P. Müller for his support, also Philipp Kraus and Malte Aschermann for creating and developing the framework LightJason, which I used to implement my simulation tool LightVoting. Furthermore, I would like to thank Philipp Kraus for his suggestions and ideas, for example regarding using Foursquare data for the preference generation.

I also would like to thank all of my colleagues in the SocialCars research training group, who provided moral or specialist support, or both. Here, I would like to emphasise the support and helpful feedback I received from all professors, including my second supervisor Markus Fidler, as well as professional discussions with the other PhD students, especially Paul Czioska and Aleksandar Trifunović, who I collaborated with to investigate voting in the context of a meeting-point recommendation system. Also, thank you to Paul for answering my questions regarding distance costs. To all PhD students as well as Andrea, Matthias and Inbal, thank you for providing a working environment I highly enjoyed.

Furthermore, I would like to thank my family, who has provided me with moral and emotional support those last years, especially my parents and my grandmother.

Last but not least, thank you to Malte for everything.

Abstract

In the last decades, intelligent transport systems have gained importance. We consider a subarea of cooperative traffic management, namely collective decision-making in groups of traffic participants. In the scenario we are studying, tourists visiting a city are asked to form travel groups and to agree on common points of interest. We focus on voting as a collective decision-making process. Our question is how different algorithms for the formation of travel groups and for determining common travel destinations differ with respect to system and user goals, where we define as system goal large groups and as user goals high preference satisfaction and low organisational effort. We aim at achieving a compromise between system and user goals.

What is new is that we investigate the inherent effects of different voting rules, voting protocols and grouping algorithms on user and system goals. Older works on collective decision-making in traffic focus on other target quantities, do not consider group formation, do not compare the effects of several voting algorithms, use other voting algorithms, do not consider clearly defined groups of vehicles, use voting for other applications or use other collective decision-making algorithms than voting.

In the main simulation series, we examine different grouping algorithms, voting protocols and committee voting rules. We consider sequential grouping vs. coordinated grouping, basic protocol vs. iterative protocol and the committee voting rules Minisum-Approval, Minimax-Approval and Minisum-Ranksum. The simulations were conducted using the newly developed simulation tool LightVoting, which is based on the multi-agent framework LightJason.

The experiments of the main simulation series show that the committee voting rule Minisum-Ranksum in most cases yields better than or as good results as the committee voting rules Minisum-Approval and Minimax-Approval. The iterative protocol tends to yield an improvement regarding preference satisfaction, at the cost of strong deterioriation regarding the group size. The coordinated grouping tends to yield an improvement regarding the preference satisfaction at relative small cost regarding the group size. This leads us to recommend the committee voting rule Minisum-Ranksum, the basic protocol and coordinated grouping in order to achieve a compromise between system and user goals. We also demonstrate the effect of different combinations of grouping algorithms and voting protocols on travel costs. Here, the combination of the basic protocol and coordinated grouping yields a compromise between preference satisfaction and traveller costs.

Additionally to the main simulation series, we provide an extended model which generates traveller preferences by combining attractiveness of the points of interest and distance costs based on the distances between the points of interest.

As further application of voting, we consider a meeting-point scenario where a range voting rule and a minimax voting rule are used to agree on meeting points. For smaller groups, the average maximum travel time is clearly higher for range voting. For larger groups, the difference decreases. For smaller groups, the average lateness for the group using minimax voting is high, for larger groups it decreases. Hence, it makes sense for smaller groups to use the minimax voting rule if one aims at fairer distribution of travel times, and to use the range voting rule if the goal is instead to avoid delay for the group.

For future work, it would be useful to adapt the simulation concept to take real-world conditions and requirements into account. Further possibilities for future work would be considering additional algorithms and models, such as considering combinatorial voting or running simulations based on the extended model, considering the role of financial incentives to encourage ridesharing or platooning and using the LightVoting tool for further research applications.

Zusammenfassung

In den letzten Jahrzehnten haben intelligente Verkehrssysteme an Bedeutung gewonnen. Wir betrachten einen Teilbereich des kooperativen Verkehrsmanagements, nämlich kollektive Entscheidungsfindung in Gruppen von Verkehrsteilnehmern. In dem uns interessierenden Szenario werden Touristen, die eine Stadt besuchen, gebeten, Reisegruppen zu bilden und sich auf gemeinsame Besuchsziele (Points of Interest) zu einigen. Wir konzentrieren uns auf Wählen als Gruppenentscheidungsverfahren. Unsere Fragestellung ist, wie sich verschiedene Algorithmen zur Bildung von Reisegruppen und zur Bestimmung gemeinsamer Reiseziele hinsichtlich der System- und Benutzerziele unterscheiden, wobei wir als Systemziel große Gruppen und als Benutzerziele hohe präferenzbasierte Zufriedenheit und geringen organisatorischen Aufwand definieren. Wir streben an, einen Kompromiss zwischen System- und Benutzerzielen zu erreichen.

Neu ist, dass wir die inhärenten Auswirkungen verschiedener Wahlregeln, Wahlprotokolle und Gruppenbildungsalgorithmen auf Benutzer- und Systemziele untersuchen. Ältere Arbeiten zur kollektiven Entscheidungsfindung im Verkehr konzentrieren sich auf andere Zielgrößen, betrachten nicht die Gruppenbildung, vergleichen nicht die Auswirkungen mehrerer Wahlalgorithmen, benutzen andere Wahlalgorithmen, berücksichtigen nicht klar definierte Gruppen von Verkehrsteilnehmern, verwenden Wahlen für andere Anwendungen oder betrachten andere Algorithmen zur kollektiven Entscheidungsfindung als Wahlen.

Wir untersuchen in der Hauptsimulationsreihe verschiedene Gruppenbildungsalgorithmen, Wahlprotokolle und Komiteewahlregeln. Wir betrachten sequentielle Gruppenbildung vs. koordinierte Gruppenbildung, Basisprotokoll vs. iteratives Protokoll und die Komiteewahlregeln Minisum-Approval, Minimax-Approval und Minisum-Ranksum. Die Simulationen wurden mit dem neu entwickelten Simulationswerkzeug LightVoting durchgeführt, das auf dem Multi-Agenten-Framework LightJason basiert.

Die Experimente der Hauptsimulationsreihe zeigen, dass die Komiteewahlregel Minisum-Ranksum in den meisten Fällen bessere oder ebenso gute Ergebnisse erzielt wie die Komiteewahlregeln Minisum-Approval und Minimax-Approval. Das iterative Protokoll tendiert dazu, eine Verbesserung hinsichtlich der präferenzbasierten Zufriedenheit zu erbringen, auf Kosten einer deutlichen Verschlechterung hinsichtlich der Gruppengröße. Die koordinierte Gruppenbildung tendiert dazu, eine Verbesserung hinsichtlich der präferenzbasierten Zufriedenheit zu erbringen bei relativ geringen Kosten in Bezug auf die Gruppengröße. Dies führt uns dazu, die Komiteewahlregel Minisum-Ranksum, das Basisprotokoll und die koordinierte Gruppenbildung zu empfehlen, um einen Kompromiss zwischen System- und Benutzerzielen zu erreichen. Wir demonstrieren auch die Auswirkungen verschiedener Kombinationen von Gruppenbildungsalgorithmen und Wahlprotokollen auf die Reisekosten. Hier bietet die Kombination aus Basisprotokoll und koordinierter Gruppenbildung einen Kompromiss zwischen der präferenzbasierten Zufriedenheit und den Reisekosten.

Zusätzlich zur Hauptsimulationsreihe bieten wir ein erweitertes Modell an, das die Präferenzen der Reisenden generiert, indem es die Attraktivität der möglichen Ziele und Distanzkosten, basierend auf den Entfernungen zwischen den möglichen Zielen, kombiniert.

Als weiteren Anwendungsfall von Wahlverfahren betrachten wir ein Verfahren zur Treffpunktempfehlung, bei dem eine Bewertungs-Wahlregel und eine Minimax-Wahlregel zur Bestimmung von Treffpunkten verwendet werden. Bei kleineren Gruppen ist die durchschnittliche maximale Reisezeit unter der Bewertungs-Wahlregel deutlich höher. Bei größeren Gruppen nimmt der Unterschied ab. Bei kleineren Gruppen ist die durchschnittliche Verspätung für die Gruppe unter der Minimax-Wahlregel hoch, bei größeren Gruppen nimmt sie ab. Es ist also sinnvoll für kleinere Gruppen, die Minimax-Wahlregel zu verwenden, wenn man eine fairere Verteilung der Reisezeiten anstrebt, und die Bewertungs-Wahlregel zu verwenden, wenn das Ziel stattdessen ist, Verzögerungen für die Gruppe zu vermeiden.

Für zukünftige Arbeiten wäre es sinnvoll, das Simulationskonzept anzupassen, um reale Bedingungen und Anforderungen berücksichtigen zu können. Weitere Möglichkeiten für zukünftige Arbeiten wären die Betrachtung zusätzlicher Algorithmen und Modelle, wie zum Beispiel die Betrachtung kombinatorischer Wahlen oder die Durchführung von Simulationen auf der Grundlage des erweiterten Modells, die Berücksichtigung der Rolle finanzieller Anreize zur Förderung von Ridesharing oder Platooning und die Nutzung des LightVoting-Tools für weitere Forschungsanwendungen.

Contents

Chapter 1

Introduction

Over the past decades, Intelligent Transport Systems (ITS) have gained increasing importance because of the proliferation of autonomous vehicles.

Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 defined ITS as advanced applications providing innovative services related to different modes of transport and traffic management. The intent is to enable a variety of user types to be better informed so that they can make safer, more coordinated and smarter usage of transport networks. [EU, 2010]

[Singh and Gupta, 2015] described four main areas of ITS:

- Advanced Traveller Information System: provides travellers with information both pre-trip and en route
- Advanced Traffic Management System: optimises vehicle movement using real-time information
- Advanced Public Transportation System: aims to make public transport more reliable
- Emergency Management System: develops transport systems that can provide help during emergency situations

[Sanderson et al., 2012] added another point of view and used a micro-meso-macro approach to describe the levels of ITS. Figure 1.1, which is taken from [Sanderson et al., 2012, p.78], gives a schematic overview of the approach. The three levels can be described as follows [Sanderson et al., 2012, p.77]:

- The micro level deals with and models individual behaviour, providing intelligence to the vehicle. At this level, the individual agents utilise all sensing equipment and on-board technology in order to construct a decision-making system concerned with the actual execution of driving. This may include safety issues (such as emergency breaking, steering and collision warning) and fulfilment of goals and preferences of drivers and passengers (such as driving speed and safety distance).
- The meso level deals with collective decision-making in groups or vehicle clusters. This level relies on Vehicle-to-Vehicle (V2V) communication, because the cluster

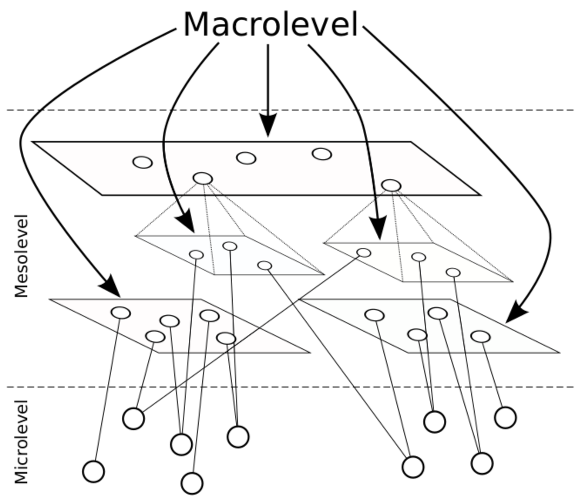

Figure 1.1: Micro-meso-macro framework

members need to inform each other about their intentions while negotiating to reach an agreement.

- The macro level deals with infrastructure or system-wide goals such as reducing congestion and pollution, making efficient use of the road network or managing the interaction of the network and other infrastructures such as energy systems, public transportation and emergency services.

With the emergence of intelligent autonomous traffic systems that can exchange information via Vehicle-to-X (V2X) communication, possible realisations of the meso level have gained relevance.

Parallel to the emergence of ITS, the development and usage of autonomous vehicles has recently been on the rise. These vehicles are autonomous insofar that they conduct traffic manoeuvres after eliciting the users' preferences. A prominent example is the development of the Google Car, which was presented in 2010 [Poczter and Jankovic, 2014]. In the 2010s, several partially autonomous vehicles were released by Mercedes (2013) [Stenquist, 2013] and Tesla (2015) [Nelson, 2015]. In 2016, Singapore launched a self-driving taxi service, implemented by nuTonomy [Watts, 2016]. Many car providers have plans regarding (further) development of autonomous vehicles. For example, Volkswagen is planning to reveal an autonomous production version of the Volkswagen microbus in 2022 [Bomey, 2021].

Classically, when investigating traffic situations, one either considers the effects of top-down, centralised management strategies (e.g. traffic-light management) that aim to improve target quantities such as emission or bottom-up, decentralised user strategies that aim to improve target quantities such as individual travel time. The availability of V2X communication enables coordination and cooperation in vehicle groups so that strategies or methods at the meso level can be developed, considering both the user and system perspectives. It can support collective decision-making for groups of traffic participants.

In the context of cooperative traffic management, several questions arise. For one, there is the question of how to conduct coordination between autonomous vehicles to handle real-time traffic manoeuvres. One possibility for coordination is the application of policy-based approaches, see e.g. the paper by [Aschermann et al., 2017], which investigated the optimisation potential for policy-based traffic management on 2+1 roadways. Another question is how to enable cooperation between several autonomous vehicles based on user preferences.

We focus on collective decision-making in the areas of platooning and ridesharing[1]. There are several arguments for using platoons. [Bergenhem et al., 2012] quoted the study by [Carbaugh et al., 1998], which found that the crash probability of severe crashes was lower on highways with platoons of autonomous vehicles than with individual ones. Additionally, [Bergenhem et al., 2012] showed that platooning led to higher road throughput, owing to the spacing between vehicles (2011 GCDC). As mentioned for example in [Haas and Friedrich, 2017], platooning is considered promising in terms of road utilisation and fuel reduction.

There are several possible methods for collective decisions at the meso level. We briefly describe some options.

- Reservation-based approaches: In reserved-based approaches as investigated by [Dresner and Stone, 2008] and [Vasirani and Ossowski, 2009], driver agents reserve resources by communicating with intersection managers.

- Market-based approaches: In the approach by [Grimaldo et al., 2012], each travel alternative is viewed as an allocation. Auctions are used as the allocation procedure, and for winner determination, a multi-criteria winner determination approach is used to merge the collected preferences.

- Negotiations: According to [Beer et al., 1999, p.1], "*negotiation is a key form of interaction that enables groups of agents to arrive at a mutual agreement regarding some belief, goal or plan, for example. Particularly because the agents are autonomous and cannot be assumed to be benevolent, agents must influence others to convince them to act in certain ways.*" They also explain that there are different forms of negotiation, including auctions, contract net protocols and argumentation.

- Auctions: [Kokkinogenis et al., 2019] used auction rules as a mechanism for tactical-level collective decision-making in platooning applications.

- Voting: This refers to an agreement to a common decision/goal/action via holding an election.

This dissertation investigates voting as a method for collective decision-making in transport systems. Thus far, we have described some recent technological developments necessary for applying collective decision-making in traffic management and have provided a short overview of collective decision-making approaches that can be applied to traffic. To continue this introduction, in Section 1.1, we describe how voting can be applied in traffic. In Section 1.2, we list some possible applications for voting in traffic management. Section

[1] While we consider platooning for our main application, we consider ridesharing in an excursus.

1.3 describes the problem we focus on in this dissertation, namely common destinations for travel groups. Section 1.4 depicts how the problem can be solved using voting with an example. The methodology is briefly described in Section 1.5. Finally, Section 1.6 outlines the structure of this dissertation.

1.1 Voting in traffic

Historically, a common way to achieve consensus regarding a common decision/goal/action for a clearly defined group where the members have differing preferences has been to hold an election. Applying voting in traffic makes sense when groups of traffic participants (e.g. for ridesharing or platooning) must agree on a common goal or action.

When applying voting to traffic management, we consider automated vehicle agents who represent the respective travellers. For example, if three travellers need to agree on two of three possible destinations to visit as a group, each preference must be elicited by the vehicle agents and aggregated in an automated manner. Based on the aggregated preferences of the group, the travel group executes an itinerary. To agree on common destinations, different voting rules can be used.

1.2 Applications for voting in traffic management

There are several scenarios of cooperative traffic management in which the application of voting can be useful:

1. Leader election in platoons, for example in [Singh et al., 2018].
2. Election of a common platoon speed [Teixeira et al., 2018].
3. Meeting-point election for drivers and riders [Czioska et al., 2017]. This application is described in an excursus in Chapter 9.
4. Election of common destinations by visitors of an intraurban area as described by [Dennisen and Müller, 2015] and [Dennisen and Müller, 2016]. In this dissertation, the focus is on this last scenario.

1.3 Problem description: common destinations for travel groups

We illustrate the concept of voting in urban traffic management using a future traffic scenario. In this scenario, there are three types of stakeholders: travellers, vehicles and traffic management. The travellers arrive with their vehicles to visit Points of Interest

(POIs) in a city. We assume that the vehicles are autonomous pods which can be coupled for physical platooning. In the following, we give some examples for the development in the areas of pods, autonomous vehicles and vehicles which can be coupled.

Figure 1.2 depicts a miniature electric vehicle developed by the Micro company; it was announced in 2018 and has two seats [Sierzputowski, 2018] Figure 1.3 shows automated electric logistics vehicles which can be virtually coupled [DroidDrive GmbH, 2021a]. Figure 1.4 depicts the process of joining modular electric vehicles by Next Logistics. In 2018, two fully functional Next pods were presented at the Dubai World Government Summit [Next, 2020a].

Figure 1.2: Mini electric vehicle [Micro Mobility Systems AG, 2018]

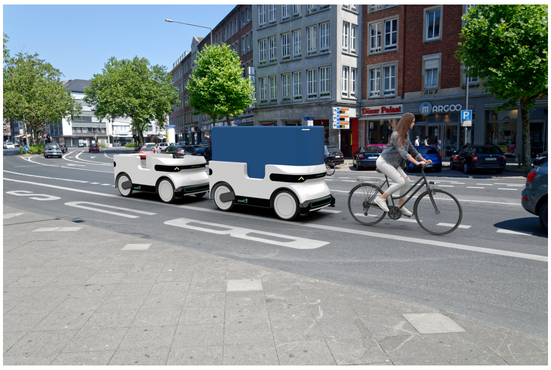

Figure 1.3: Virtual coupling of logistics vehicles [DroidDrive GmbH, 2021b]

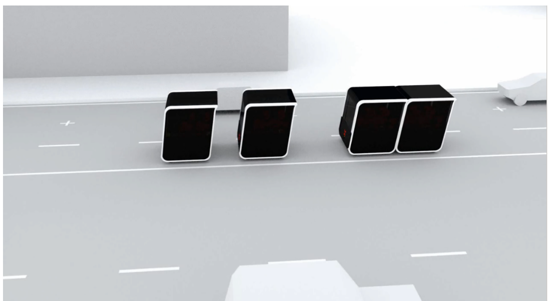

Figure 1.4: Coupling of pods [Next, 2020b]

These technologies enable it to make traffic more modular: travellers can use pods to visit cities and technologies for physical coupling enable organising travellers into groups that can communicate via V2X.

As mentioned above, there are advantages to platooning. In our scenario, we assume that, due to energy, environmental and safety concerns, urban traffic management encourages visitors to form travel groups at predefined assembly points. Especially with automation, fewer large platoons can be more easily managed than many small groups. This leads us to consider large groups as system goal in our scenario. As user goals in our scenario, we consider high preference fulfilment and low organisational effort. Preference fulfilment measures, given a traveller's preferences and a chosen set of common destinations for a platoon, how satisfied the traveller is with this choice. As for organisational effort, consider a set of travellers which arrive at a predefined assembly point and need to decide with whom to form groups and to agree on common destinations to visit. The more actions a traveller has to conduct and the longer the traveller has to wait for other travellers, the higher the organisational effort is for this traveller. The interaction between traffic management and user behaviour is guided by the overall goal of establishing a situation where there is a compromise between user and system goals to achieve acceptable levels of efficiency and safety.

In Figure 1.5, we schematically show the overall process for the urban visitors scenario. We have travellers with preferences who arrive at an assembly point and need to form travel groups, which have to agree on common destinations to visit.

This dissertation considers the following problem for the urban visitors scenario: *Given the traveller preferences, how should the travellers be grouped together and which POIs should they visit?*

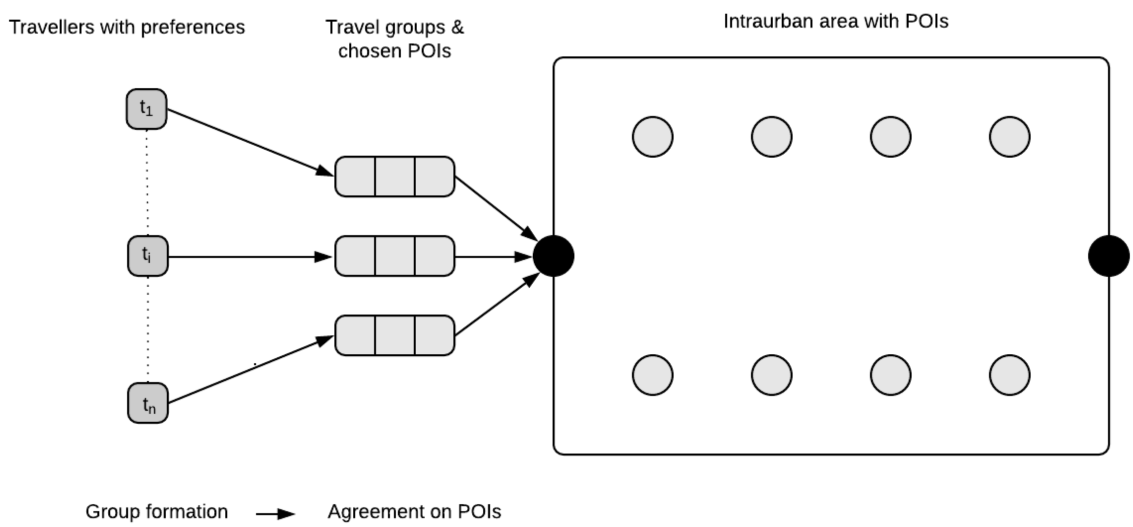

Figure 1.5: Process for the urban visitors scenario

Note that trip-planning is viewed as a downstream problem in this dissertation. We consider some examples for demonstrating how driving in groups can decrease individual travel costs, see Chapter 7. We also describe an extended model in Chapter 8 that considers distance costs. However, for the examples and the extended model, we assume that Travelling Salesperson Problem (TSP) approximations are used for the actual routing.

1.4 Solving the problem using voting

In this section, we use a small example to demonstrate how the problem described in Section 1.3 can be solved by using voting, see Figure 1.6, which is slightly adapted from [Dennisen and Müller, 2015, p.207]. Because several POIs have to be selected, we focus on committee elections.

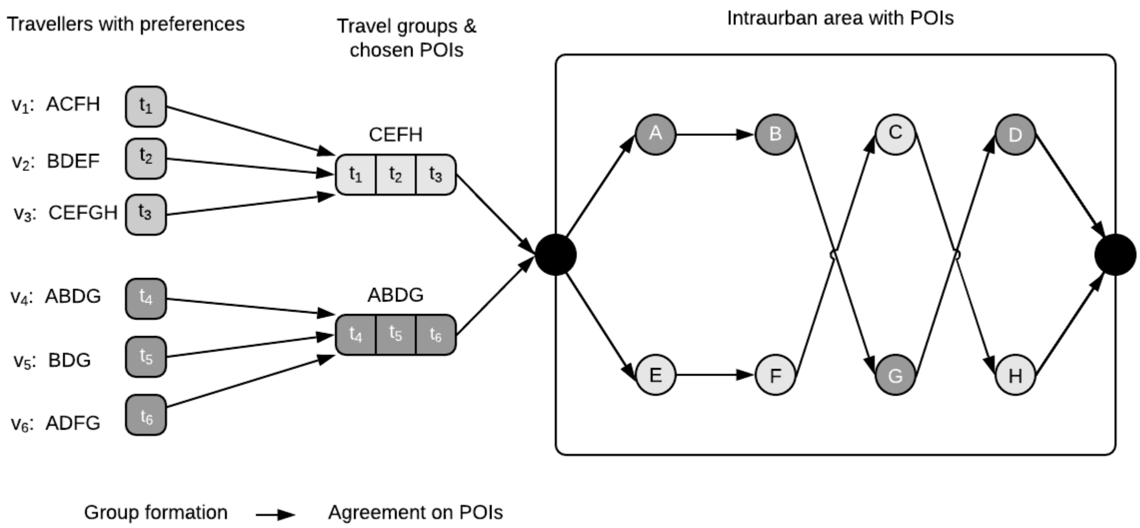

Figure 1.6: Example for urban visitors scenario

Six travellers $t_1, ..., t_6$ arrive at a defined assembly point desiring to visit an intraurban area. Each traveller t_i has a vote v_i that represents their preferences over the set of possible POIs $P = \{A, B, C, D, E, F, G, H\}$ in the intraurban area. In this example, we assume the votes to be Approval votes, which can be written as 01-vectors or as set representation. For example, vote v_1 is specified as $ACFH$, short for $\{A, C, F, H\}$. Thus, traveller t_1 approves of POIs A, C, F and H and disapproves of B, D, E and G. This can also be expressed (assuming the the fixed order $A, ..., H$) via the 01-vector $(1, 0, 1, 0, 0, 1, 0, 1)$. [Dennisen and Müller, 2015, p.207]

It is assumed that the travellers are sequentially grouped. Travellers t_1, t_2 and t_3 are grouped together into group G_1. The others (t_4, t_5 and t_6) are grouped into G_2. We call this straightforward grouping algorithm *sequential grouping*.

Furthermore, we assume a *basic voting protocol*, meaning that once the election for the filled group has been conducted, it can leave.

Lastly, we assume a *committee voting rule*, k-Minisum-Approval, where k specifies the size of the to-be-elected committee. We consider $k = 4$.

For our main study, we have those three important parameters: grouping algorithm, voting protocol and voting rule, and will consider further options for each.

In k-Minisum-Approval, a winning committee consists of k most often approved alternatives. In this example, in group G_1, POIs C, E, F and H are each approved by two travellers, whereas POIs A, B, D and G are each approved by one traveller, making $\{C, E, F, H\}$ a unique winning committee. In group G_2, the POIs D and G are each approved by three travellers, whereas POIs A and B are each approved by two. POIs C, E and H are approved by zero travellers. This makes $\{A, B, D, G\}$ a unique winning committee. [Dennisen and Müller, 2015, p.207f.]

Focusing on voting as a collective decision-making mechanism, we consider the following research question: *How do different algorithms used for creating travel groups and for determining common destinations compare regarding system and user goals?*

1.5 Methodology

For evaluation of the proposed algorithms, we conduct agent-based simulations based on two forms of preference generation. The first includes the generation of uniform distributed preferences; the second includes a more realistic type of preference generation that relies on the platform Foursquare, as described in Chapter 4. We compare several voting rules, voting protocols and grouping algorithms with respect to the quantities group size, preference dissatisfaction and organisational effort. We follow an approach similar to [Carley, 1999], where we use the simulation results to generate hypotheses.

1.6 Structure

This dissertation is structured as depicted in Figure 1.7. Chapter 2 and 3 yield a problem description. Chapter 2 yields fundamental definitions and describes related work from areas of Computational Social Choice, transport applications, collective decision-making in traffic management and multi-agent systems. In Chapter 3, we describe the research gap and our research question. Chapter 4 explains the used approaches in detail, i.e. our assumptions and the applied algorithms: voting rules, voting protocols and grouping algorithms. Chapters 5 and 6 comprise the evaluation part of the dissertation. Chapter 5 gives details on the implementation. Here, we describe the simulation architecture of the agent-based tool as well as the preferences generation and the statistical component. In Chapter 6, the experiments and results are discussed. In Chapter 7, we consider the effects of different combinations of voting protocols and grouping algorithms on travel costs. Chapter 8 describes an extension of the model described in Chapter 4 by taking distance costs into account when defining traveller preferences. In an excursus in Chapter 9, an alternative application for voting rules for cooperative traffic is described (i.e. voting to agree on meeting points for ridesharing). This application is part of Paul Czioska's

PhD research and was investigated in a joint paper. Chapter 10 concludes the thesis and gives an outlook on possible future works.

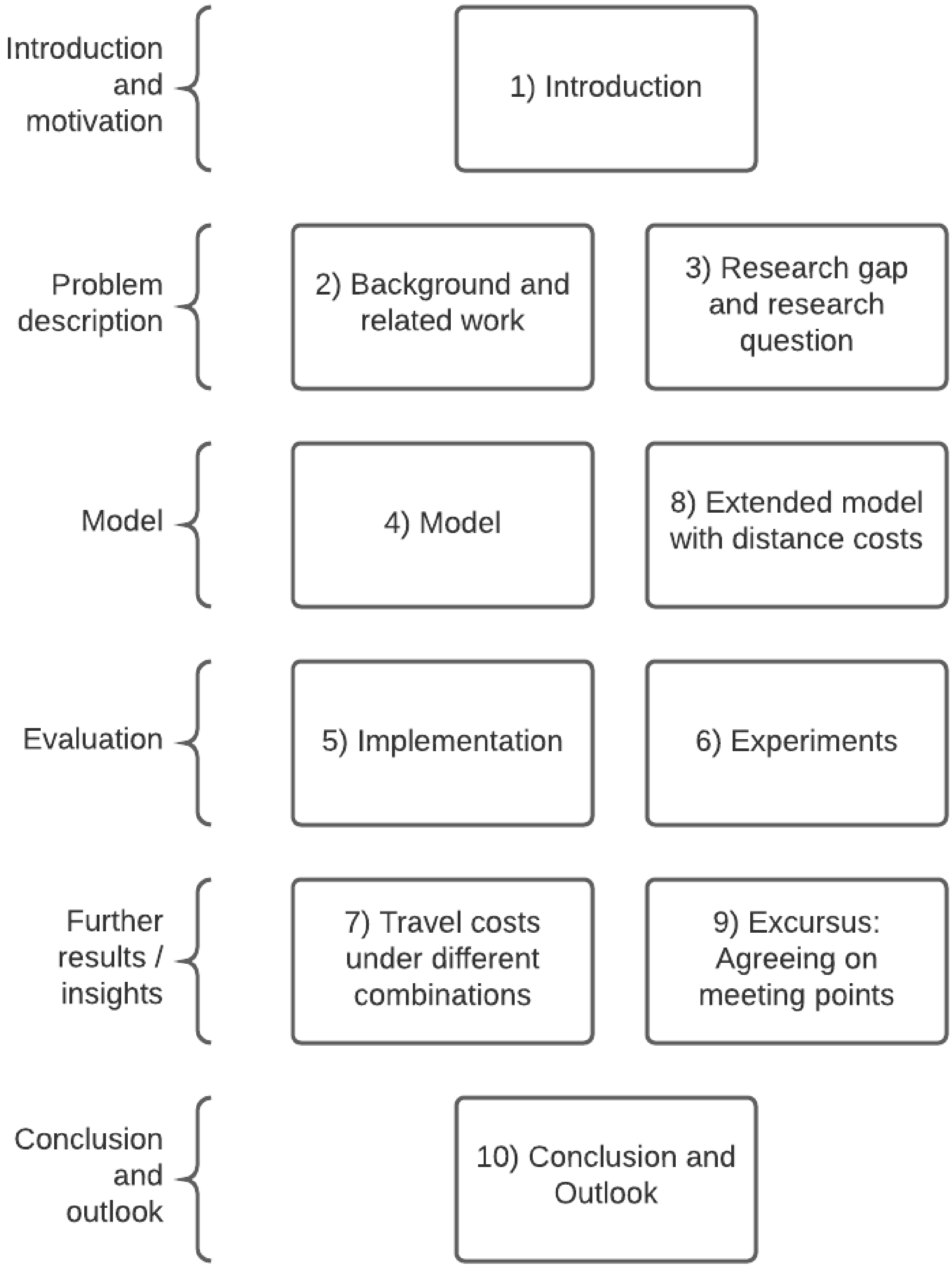

Figure 1.7: Dissertation structure

Chapter 2

Background and Related Work

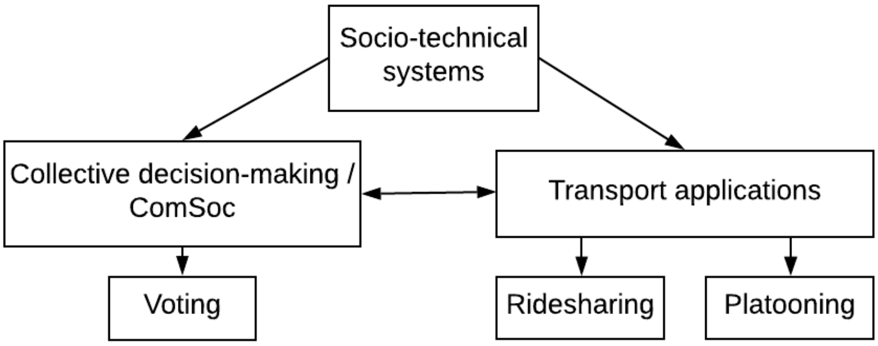

Figure 2.1: Topic overview

The research in this dissertation can be viewed as research on a specific type of socio-technical systems. A socio-technical system is defined by [Singh, 2013] as "*a microsociety in which autonomous parties interact with and about technical objects*". According to Singh, a challenge of socio-technical systems is to enable the effective collaboration of autonomous stakeholders whose interests may be imperfectly aligned. One possibility for achieving collaboration is the application of collective decision-making mechanisms. In our research topic, we consider socio-technical systems consisting of traffic participants who use voting as collective decision-making mechanism for collaboration. Hence, the research topic considered in this dissertation lies at the intersection of two domains:

1) Collective decision-making/Computational Social Choice

2) Transport applications

An overview of the classification scheme is shown in Figure 2.1. This leads us to consider related works in the area ComSoc, including voting, as well as related works on transport research in the areas ridesharing and platooning. We also consider related works on existing approaches for collective decision-making in traffic. Additionally, as described in Chapter 1, we use a multi-agent simulation for the evaluation of several voting algorithms for collective decisions in traffic-participant groups. Hence, we consider the subtopics collective decision-making/Computational Social Choice, transport applications, collective decision-making in traffic and agents.

In this chapter, we present necessary definitions and discuss related works. In the first section, related collective decision-making models are described, and in the second section, relevant transport applications are presented. The third section focuses on existing applications of collective decision-making methods in traffic, including voting applications. The fourth section explains agent definitions and considers related simulation frameworks and tools.

2.1 Collective decision-making & Computational Social Choice

In this section, we introduce the topic Computational Social Choice, based on [Brandt et al., 2016a] and [Rothe et al., 2012].

Classical Social Choice Theory is an old field which investigates the aggregation of individual preferences into a common choice. The paradigmatic situation in Social Choice Theory is the conduction of an election, i.e. creating consensus with the help of a voting rule. A voting rule is an algorithm that yields one or several winners from a candidate set based on a list of votes.

An early representative of the field Social Choice Theory is Ramon Llull, who proposed an early variant of the Copeland voting rule in the 13th century, as discussed e.g. by [Faliszewski et al., 2008]. Other prominent researchers are Condorcet, who proposed the Condorcet voting rule in 1785 [Condorcet, 1785] and Dodgson[1], who created the Dodgson rule [Dodgson, 1876]. In all of these voting rules, winner determination is based on pairwise comparisons.

A big group of voting rules is the group of position-based voting rules. Scoring protocols such as Plurality, Borda (proposed by [Borda, 1781]) and Veto belong to this group. In these rules, voters submit their preferences by ranking the candidates, and each candidate receives for each vote a score dependent upon its position in the vote.

Approval voting is another fundamental rule. As proposed by [Brams and Fishburn, 1978], a candidate is either disapproved or approved by a voter, and the candidate receives one point for approval and zero points for disapproval. The candidate(s) with the highest Approval score win(s).

Modern Social Choice Theory goes back to the impossibility theorem of Arrow, which was originally formulated in 1951 and revised in 1963 [Arrow, 1963]. It was later rephrased by [Taylor, 1995, Taylor, 2005]. It states that for elections with at least three candidates, it is impossible to find a preference-based voting rule that is simultaneously non-dictatorial, weakly Pareto-efficient and independent from irrelevant alternatives. This can be crudely interpreted as "the perfect voting rule does not exist".

Compared with classical Social Choice Theory, the area Computational Social Choice is relatively young. As described by [Brandt et al., 2016a, p.8], "*by the early 2000s*

[1]Better known as Lewis Carroll, the author of 'Alice's Adventures in Wonderland'

(the) trend towards studying collective decision-making in the tradition of classical Social Choice Theory [...] with a specific focus on computational concerns had reached substantial momentum." It should be noted that there were precursors in the 60s and 80s, such as the Gale–Shapley algorithm by [Gale and Shapley, 1962] and publications by [Bartholdi et al., 1989b] and [Bartholdi et al., 1989a] on the complexity of manipulating elections and winner determination.

While much of classical Social Choice Theory relates to voting, resource allocation and coalition formation gained importance with the emergence of Computational Social Choice.

The area resource allocation can be viewed as related to our scenario because we assign POIs to travellers in the urban visitors scenario. In resource allocation, divisible or indivisible resources are allocated to agents based on their preferences. Coalition formation is related to our application because we aim to divide the set of agents into subsets. In coalition formation, agents are divided into subsets based on their preferences.

Collective decision-making algorithms can be used for all types of agents. In the context of this dissertation, these algorithms are applied for vehicle agents based on the preferences of their passengers.

In the following subsections, we define several terms from Computational Social Choice, namely basic concepts and definitions, committee voting rules, coalition formation, group-activity selection problems and resource allocation.

2.1.1 Voting: basic concepts and definitions

Here, we recapitulate some voting definitions. Apart from the definition of voting protocols and grouping algorithms, these stem from [Rothe et al., 2012, p.122f., p.135f.]. The explanations also closely follow this source.

An *election or preference profile* (C, V) consists of

- a set of candidates C
- a list of voters V

How the preferences of the voters are represented as votes depends on the voting rule. In this dissertation, we restrict ourselves to two types of votes: Approval votes and complete linear orders.

An *Approval vote* for a voter v and candidates $C = \{c_1, ..., c_m\}$ is a $\{0,1\}^m$ vector. For position i in the vote, a '0' means that voter v disapproves of candidate c_i, and a '1' means that voter v approves of candidate c_i.

A *complete linear order* for voter v and candidates $C = \{c_1, ..., c_m\}$ specifies a strict ranking over all candidates, with the most liked candidate ranked first.

Mathematically, a complete linear order is defined as a relation $>$ with the following properties [Rothe et al., 2012, p.123, translated and punctuation adapted]:

- *Total: for any two candidates c and d in C, it either holds $c > d$ or $d > c$*
- *Transitive: for any three candidates c, d and e in C, it follows from $c > d$ and $d > e$ that $c > e$*
- *Symmetrical: for any two candidates c and d in C, it follows from $c > d$ that $d > c$ does not hold*

A *voting rule* is a rule that determines the winner(s) of a given election (C, V) and is defined as a social choice correspondence

$$\{(C, V)|(C, V) \text{ is a preference profile}\} \to \mathcal{P}(C).$$

[Rothe et al., 2012, p.122, translated]

A *social choice function* is defined as a mapping

$$\{(C, V)|(C, V) \text{ is a preference profile}\} \to C$$

[Rothe et al., 2012, p.123, translated] that assigns to each given election exactly one winner.

Let $\mathcal{R}(C)$ denote the set of all complete linear orders over C. Then, a *social welfare function* is a mapping

$$\{(C, V)|(C, V) \text{ is a preference profile}\} \to \mathcal{R}(C).$$

[Rothe et al., 2012, p.123, translated and corrected]

In the context of this dissertation, a *grouping algorithm* is an algorithm defining how an agent is assigned to a voter group.

A *voting protocol* is an interaction protocol used when applying a voting rule; it describes the interaction between the chair and the voters once they have joined the electorate.

Next, we provide examples for some voting rules mentioned in Section 2.1: Condorcet, Copeland/Llull, Dodgson, Plurality, Veto and Borda. All examples and the respective explanations are taken or slightly adapted from [Rothe et al., 2012].

2.1.1.1 Condorcet

As defined by [Condorcet, 1785], in a Condorcet election, a Condorcet winner is the candidate that defeats all other candidates by a strict majority in pairwise comparisons. The example in Table 2.1 is adapted from [Rothe et al., 2012, p.127]. The votes are given in form of complete linear orders. In this example, candidate A wins each pairwise comparison it is involved in and is thus the Condorcet winner of the election.

Vote	$A:B$	$A:C$	$A:D$	$B:C$	$B:D$	$C:D$
v_1: $D > B > A > C$	B	A	D	B	D	D
v_2: $B > D > A > C$	B	A	D	B	B	D
v_3: $A > C > D > B$	A	A	A	C	D	D
v_4: $C > A > D > B$	A	C	A	C	D	C
v_5: $C > A > B > D$	A	C	A	C	B	C
Winner of pairwise comparison	A	A	A	C	D	D

Table 2.1: Election with results for pairwise comparisons

2.1.1.2 Copeland and Llull

In this section, we closely follow [Rothe et al., 2012, p.128ff.] and explain the Copeland and Llull voting rule. The Copeland voting rule respects the Condorcet winner, that is, if a Condorcet winner exists, Copeland chooses this candidate. The Copeland voting rule as proposed by [Copeland, 1951] works like this. If for a pairwise comparison there is a winner with a strict majority, this candidate is rewarded with one point, and the other candidate obtains zero points. If in a pairwise comparison there is no winner having a strict majority, both candidates get one-half point each. The candidate with the highest score wins.

[Faliszewski et al., 2009] proposed the Copeland family of voting rules. For a rational number α with $0 \leq \alpha \leq 1$, Copeland$^\alpha$ works similar to the Copeland voting rule, except that for a tie in a pairwise comparison, the candidate receives α points. The Llull voting rule corresponds to Copeland^1 with α set to 1.

Table 2.2 shows an election without Condorcet winner, as taken from [Rothe et al., 2012, p.130]. The votes are again given in form of complete linear orders, and we consider the original Copeland voting rule, Copeland$^{1/2}$. Candidates A and C are the Copeland$^{1/2}$ winners with the following Copeland$^{1/2}$ scores for the election:

$C^{1/2}Score(A) = 1 + 2 * 1/2 = 2$

$C^{1/2}Score(B) = 1/2$

$C^{1/2}Score(C) = 2$

$C^{1/2}Score(D) = 1 + 1/2 = 1.5$

2.1.1.3 Dodgson voting rule

In this section, we explain the Dodgson voting rule, closely following [Rothe et al., 2012, p.131]. In the Dodgson voting rule, the votes are given in form of complete linear orders,

Vote	$A:B$	$A:C$	$A:D$	$B:C$	$B:D$	$C:D$
v_1: $A > D > C > B$	A	A	A	C	D	D
v_2: $C > D > B > A$	B	C	D	C	D	C
v_3: $B > D > A > C$	B	C	D	C	D	C
v_4: $A > C > D > B$	B	A	D	B	B	D
v_5: $A > C > D > B$	A	A	A	C	D	C
v_6: $A > C > B > D$	A	A	A	C	B	C
Winner of pairwise comparison	?	A	?	C	D	C

Table 2.2: Election without Condorcet winner with results for pairwise comparisons

and the rule uses pairwise comparisons. It respects the Condorcet winner and always has at least one winner. To determine a winner, the Dodgson score of a candidate is determined as the minimal number of candidate swaps in the votes to make this candidate a Condorcet winner. The candidates with the lowest Dodgson score win.

For the election in Table 2.2, candidate A has a Dodgson score of two. It is not possible to make A a Condorcet winner with only one swap, and it is possible to make them a Condorcet winner with two swaps, such as in v_2:

$C > D > \underline{B > A} \rightarrow C > \underline{D > A} > B \rightarrow C > A > D > B.$

Candidate C also has a Dodgson score of two with the following two swaps:

$v_5 : \underline{A > C} > D > B \rightarrow C > A > D > B$

$v_6 : \underline{A > C} > B > D \rightarrow C > A > B > D$

Because the Dodgson scores for B and D are greater than two, A and C are the Dodgson winners in this election.

2.1.1.4 Scoring rules

Next, we explain the concept of scoring rules and provide an example for an election and the results under three well-known scoring rules, closely following [Rothe et al., 2012, p.124f.]. A scoring vector determines how many points a candidate gains depending on its position in the respective vote, which has the form of a complete linear order. As described in [Rothe et al., 2012, p.124, translated], "*[a] scoring vector for m candidates has the form*

$$\vec{\alpha} = (\alpha_1, ..., \alpha_m),$$

where α_i are natural numbers fulfilling the inequation $\alpha_1 \geq ... \geq \alpha_m$." A candidate on position i in a vote receives α_i points from the corresponding voter. The winners of an election are the candidates with the most points over all votes.

We briefly list three well-known scoring rules and the corresponding scoring vectors:

- Plurality: $(1, 0, ..., 0)$
- Veto: $(1, ..., 1, 0)$
- Borda: $(m-1, m-2, ..., 0)$

The following tables are adapted from [Rothe et al., 2012, p.125]. As can be seen in Table 2.3, under Plurality, candidate A wins with a score of two. Table 2.4 shows that under Veto, candidate B wins with a score of three. Under Borda, both candidates A and B win with a score of four (see Table 2.5).

v_1: $A > B > C$	1	0	0
v_2: $B > C > A$	0	1	0
v_3: $A > B > C$	1	0	0
Scores	**2**	1	0

Table 2.3: Election under Plurality

v_1: $A > B > C$	1	1	0
v_2: $B > C > A$	0	1	1
v_3: $A > B > C$	1	1	0
Scores	2	**3**	1

Table 2.4: Election under Veto

v_1: $A > B > C$	2	1	0
v_2: $B > C > A$	0	2	1
v_3: $A > B > C$	2	1	0
Scores	**4**	**4**	1

Table 2.5: Election under Borda

2.1.2 Committee elections

We aim to create consensus on a fixed number of destinations for travel groups. To this end, we will use committee voting rules. Committee elections have been considered in several works, such as the master's thesis by Dennisen[2], where the following definition was used:

A *committee election* (C, V, k) consists of

- A set of candidates C

[2] Messung und Minimierung der Wählerunzufriedenheit in Komiteewahlen. S. Dennisen. Master's thesis. Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany, 2014.

- A list of voters V
- A committee size k

Let $F_k(C)$ denote the set of all subsets of C of size k. Then, we define a *committee voting rule* as a mapping

$$\{(C, V, k) | (C, V, k) \text{ is a committee election}\} \to \mathcal{P}(F_k(C))$$

that assigns to each committee election a set of committees of size k. Via tie-breaking, one can determine a unique winning committee.

Similar to the master's thesis of Dennisen, for a committee K, the vector representation $vec(K)$ can be defined as follows: For a fixed order $(c_1, ...c_m)$ of the candidates in C, we represent in the vector representation a candidate c_i by placing a zero at the $i-th$ position if the candidate was not elected and by placing a one at $i-th$ position otherwise.

We will focus on comparing two well-known committee voting rules, both based on Approval votes, namely k-Minisum-Approval and k-Minimax-Approval[3], as well as a committee voting rule based on complete linear orders, namely k-Minisum-Ranksum[4]. The subsequent formulations follow the master's thesis by Dennisen[5].

We define the total disagreement of the electorate V with committee $K \subseteq C$ as

$$\begin{aligned}&totdis(C, V, K)\\ &= \sum_{v \in V}(disag(v, K)).\end{aligned}$$

The maximum disagreement of the electorate V with committee $K \in C$ is defined as

$$\begin{aligned}&maxdis(C, V, K)\\ &= \max_{v \in V}(disag(v, K)).\end{aligned}$$

For Approval votes, we define the disagreement of voter v with committee K as

$$\begin{aligned}&disag(v, K)\\ &= HD(v, vec(K))\end{aligned}$$

where $HD(vec_i, vec_j)$ is the Hamming distance between two $(0, 1)$-vectors vec_i and vec_j, which counts the number of positions where the two vectors differ and which was originally developed as a means of error detection in binary sequences by [Hamming, 1950].

[3] Both approaches have been considered by [Brams et al., 2004]. In this paper, Brams et al. proposed the minimax approach.

[4] k-Minisum-Ranksum was presented in [Baumeister and Dennisen, 2015].

[5] Messung und Minimierung der Wählerunzufriedenheit in Komiteewahlen. S. Dennisen. Master's thesis. Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany, 2014.

k-Minisum-Approval selects a committee K^* for which the following holds:

$$\begin{aligned}&totdis(C, V, K^*)\\&= \min_{K \in F_k(C)} totdis(C, V, K)\\&= \min_{K \in F_k(C)} \sum_{v \in V} HD(v, vec(K)).\end{aligned}$$

This corresponds to a utilitarian approach and is equivalent to selecting a committee K^* with the highest Approval score, where the Approval score of K is defined as

$$\begin{aligned}&AScore(C, V, K)\\&= \sum_{c \in K} AScore(c, V).\end{aligned}$$

For a candidate $c \in K$, the Approval score is defined as

$$\begin{aligned}&AScore(c, V)\\&= \{v \in V | v \text{ approves of c}\}.\end{aligned}$$

The following example is adapted from the master's thesis by Dennisen (p.22). We consider an example election with $k = 2$ and $C = \{A, B, C, D\}$ (see Table 2.6). The winning committees for this election are $\{A, B\}$, $\{A, C\}$ and $\{B, C\}$ with a minimum sum of five.

Committee \ Vote	1110	0101	1010	Sum
1100	1	2	2	**5**
1010	1	4	0	**5**
1001	3	2	2	7
0110	1	2	2	**5**
0101	3	0	4	7
0011	3	2	2	7

Table 2.6: Example for k-Minisum-Approval with Hamming distances

k-Minimax-Approval selects a committee K^* for which the following holds:

$$\begin{aligned}&maxdis(C, V, K^*)\\&= \min_{K \in F_k(C)} maxdis(C, V, K)\\&= \min_{K \in F_k(C)} \max_{v \in V} HD(v, vec(K)).\end{aligned}$$

This corresponds to an egalitarian approach.

The following example election is again adapted from the master's thesis by Dennisen (p.35). We consider $k = 2$ and $C = \{A, B, C, D\}$ (see Table 2.7). The winning committees are $\{A, B\}$ and $\{B, C\}$ with the minimum maximum Hamming distance of two.

Committee \ Vote	1110	0101	1010	Maximum
1100	1	2	2	**2**
1010	1	4	0	4
1001	3	2	2	3
0110	1	2	2	**2**
0101	3	0	4	4
0011	3	2	2	3

Table 2.7: Example for k-Minimax-Approval with Hamming distances

For complete linear orders, we define the disagreement between voter v and committee K via a normalised ranksum. Let $pos(c, v)$ be the position of candidate c in vote v. Then, the ranksum of voter v for committee K is

$$\begin{aligned} &rs(v, K) \\ &= \sum_{c \in K} pos(c, v) - \sum_{i=1}^{k} i. \end{aligned}$$

Analogue to k-Minisum-Approval, *k-Minisum-Ranksum* selects a committee K^* for which the following holds:

$$\begin{aligned} &totdis(C, V, K^*) \\ &= \min_{K \in F_k(C)} totdis(C, V, K) \\ &= \min_{K \in F_k(C)} \sum_{v \in V} rs(v, K). \end{aligned}$$

The following example is again taken from the master's thesis by Dennisen (p.25). We consider $k = 2$ and $C = \{A, B, C, D, E\}$ (see Table 2.8). Here, we write linear orders of the form $A > B > C > D > E$ in the form $ABCDE$ for better readability. The winning committee in this election is $\{B, E\}$ with a minimum sum of 14.

Committee \ Vote	$ABEDC$	$BECAD$	$DECAB$	$EABCD$	$CDEAB$	$EDCBA$	$EBACD$	Sum
$\{A, B\}$	0	2	6	2	6	6	2	24
$\{A, C\}$	3	4	4	3	2	5	4	25
$\{A, D\}$	2	6	2	4	3	4	5	26
$\{A, E\}$	1	3	3	0	4	3	1	15
$\{B, C\}$	4	1	5	4	3	4	3	24
$\{B, D\}$	3	3	3	5	4	3	4	25
$\{B, E\}$	2	0	4	1	5	2	0	**14**
$\{C, D\}$	6	5	1	6	0	2	6	26
$\{C, E\}$	5	2	2	2	1	1	2	15
$\{D, E\}$	4	4	0	3	2	0	3	17

Table 2.8: Example for k-Minisum-Ranksum with ranksums

For the sake of readability, we will shorten k-Minisum-Approval to Minisum-Approval, k-Minimax-Approval to Minimax-Approval and k-Minisum-Ranksum to Minisum-Ranksum in the further course of the text.

2.1.3 Coalition formation

Another important topic in Computational Social Choice is the formation of coalitions. What happens when agents can form coalitions and each agent has preferences over these coalitions? This relates to our topic insofar as we consider the question of how to form travel groups. There are three subtopics of the topic coalition formation:

- Matching under preferences
- Hedonic games
- Weighted voting games

To explain the three different classes, we closely follow the three corresponding chapters in [Brandt et al., 2016b].

2.1.3.1 Matching under preferences

The assignment of POIs to travellers seems to be similar to matching problems, which are considered by [Klaus et al., 2016] in [Brandt et al., 2016b, p.333-355]. In this section, we closely follow their explanation.

A prominent example for matching under preferences is the Stable Marriage Algorithm by [Gale and Shapley, 1962], as mentioned in Section 2.1. Matching problems can be subdivided into bipartite and non-bipartite. In bipartite matching problems, the agent set consists of two separate sets A and B, where the members of A have only preferences over the members of B. It is possible that the members of B have preferences over the members of A. In non-bipartite matching problems, there is a single set of agents, where each has a ranking over some or all of the other agents. Bipartite problems can be further subdivided into two-sided and one-sided. In the two-sided case, members of both A and B have preferences over one another. In the one-sided case, only the members of one set (A) have preferences over the members of the other one (B).

The Stable Marriage problem is a two-sided bipartite matching problem, because it assumes two sets, A and B, where the agents in each have preferences over only the agents in the other set. It is a special case of the Hospitals/Residents (HR) problem, which is formally defined as follows [Brandt et al., 2016b, p.336f., verbatim]:

Definition 1. *An instance I of* HR *involves a set $R = \{r_1, ...r_{n_1}\}$ of residents and a set $H = \{h_1, ..., h_{n_2}\}$ of hospitals. Each hospital $h_j \in H$ has a positive integer capacity, denoted by c_j, indicating the number of posts for h_j. $E \subseteq R \times H$ is the set of acceptable resident-hospital pairs. Let $m = |E|$. Each resident $r_i \in R$ has an acceptable set of* *hospitals* $A(r_i)$ *where* $A(r_i) = \{h_j \in H : (r_i, h_j) \in E\}$. *Similarly, each hospital* $h_j \in H$ *has an acceptable set of residents* $A(h_j)$ *where* $A(h_j) = \{r_i \in R : (r_i, h_j) \in E\}$.

The agents in I *are residents at hospitals in* $R \cup H$. *Each agent* $a_k \in R \cup H$ *has a preference list in which they rank* $A(a_k)$ *in strict order. Given any resident* $r_i \in R$ *and any hospitals* h_j, $h_k \in H$, r_i *prefers* h_j *to* h_k *if* $\{h_j, h_k\} \subseteq A(r_i)$ *and* h_j *precedes* h_k *on* r_i*'s preference list. The prefers relation is defined similarly for a hospital.*

An assignment M *in* I *is a subset of* E. *If* $(r_i, r_j) \in M$, r_i *is said to be assigned to* h_j, *and* h_j *is assigned* r_i. *For each* $a_k \in R \cup H$, *the set of assignees of* a_k *in* M *is denoted by* $M(a_k)$. *If* $r_i \in R$ *and* $M(r_i) = \emptyset$, *then* r_i *is said to be unassigned. Otherwise,* r_i *is assigned. Similarly, a hospital* $h_j \in H$ *is undersubscribed or full if* $|M(h_j)|$ *is less than or equal to* c_j, *respectively. A matching* M *in* I *is an assignment such that* $|M(r_i)| \leq 1$ *for each* $r_i \in R$ *and* $|M(h_j)| \leq c_j$ *for each* $h_j \in H$. *Given a matching* M *and a resident* $r_i \in R$ *such that* $M(r_i) \neq \emptyset$, *if there is no ambiguity, the notation* $M(r_i)$ *is also used to refer to a single member of the set* $M(r_i)$.

Given an instance I *of* HR *and a matching* M, *a pair* $(r_i, h_j) \in E \setminus M$ *blocks* M *(or is a blocking pair for* M*) if (i)* r_i *is unassigned or prefers* h_j *to* $M(r_i)$ *and (ii)* h_j *is undersubscribed or prefers* r_i *to at least one member of* $M(h_j)$. M *is said to be stable if it admits no blocking pair. If a resident–hospital pair,* (r_j, h_j), *belongs to a stable matching in* I, r_i *is a stable partner of* h_j *and vice versa.*

The Stable Marriage problem with Incomplete lists (SMI) [Gale and Shapley, 1962] is a special case of HR in which $c_j = 1$ for all h_j, and the classical Stable Marriage problem is a restriction of SMI in which $n_1 = n_2$ and $E = R \times H$.

Additionally to describing the different classes of matching problems and discussing the Stable Marriage problem and HR, [Klaus et al., 2016] mention that there are several variants of many-to-many matching problems considered in the literature and that those variants are often considered in the context of assigning workers to firms, meaning that each agent can be assigned multiple times according to a given capacity. Usually, both workers and firms have preferences in form of strict rankings over the agents in the other set (or over subsets of agents). In these problems, the aim is to find stable matchings. [Brandt et al., 2016b, p.344f.]

The problem considered in this dissertation is similar to bipartite matching with one-sided preferences: One can interpret the travellers as members of the set A which have preferences over the members of set B, the POIs. Unlike in this matching problem, we consider a scenario where a traveller chooses several POIs. The problem of assigning travellers to POIs is similar to many-to-many matching problems. However, in this dissertation the focus does not lie on finding stable matchings but rather on comparing the effects of several voting algorithms on user and system goals.

2.1.3.2 Hedonic games

[Aziz et al., 2016] consider hedonic games in [Brandt et al., 2016b, p.356–376]. In this section, we closely follow their explanation.

Hedonic games can be used to model research-team formation ([Alcalde and Revilla, 2004]), scheduling group activities, see [Darmann et al., 2012], forming coalition governments as described by [Le Breton et al., 2008], clustering in social networks as described by [Aziz et al., 2019] and distributed task allocation, see [Saad et al., 2010]. Hedonic games are coalition formation games with hedonic preferences, where the outcome of a coalition formation game is the partitioning of agents into disjoint coalitions. Having hedonic preferences means that an agent only cares about what agents are in its coalition and does not care about the group composition in other coalitions, see [Dreze and Greenberg, 1980].

A hedonic game can be formally defined as follows [Brandt et al., 2016b, p.357, verbatim]:

Definition 2. *Let N be a finite set of agents. A coalition is a non-empty subset of N. Let $\mathcal{N}_i = \{S \subseteq N : i \in S\}$ be the set of all coalitions (subsets of N) that include agent $i \in N$. A coalitional structure is a partition π of agents N into disjoint coalitions. A hedonic coalition formation game is a pair $(N, \succsim)$ where $\succsim$ is a preference profile that specifies for every agent $i \in N$ a reflexive, complete and transitive binary relation, $\succsim$ on $\mathcal{N}_i$. We call $\succsim_i$ a preference relation.*

Agents have an incentive to deviate if there exist deviations to new coalitions they prefer to their old one. These are called profitable deviations. [Bogomolnaia et al., 2005] define a coalition structure as "*stable with respect to a class of allowable single-agent deviations if no agent has a profitable allowable deviation*".

Additionally to the view of [Aziz et al., 2016] in [Brandt et al., 2016b], hedonic games are described by [Darmann and Lang, 2016] in another context, as explained in Subsection 2.1.4.

The situation considered in this dissertation could theoretically be modelled as a hedonic game in which every possible coalition of travellers implies, given a voting rule, a certain set of POIs for this coalition. Each traveller has a certain satisfaction value for each such POI set, implying a ranking over different coalitions. The condition that each agent only cares about which agents are in their coalition is fulfilled implicitly when assuming a fixed voting rule. However, we aim to draw comparisons between several voting algorithms.

2.1.3.3 Weighted voting games

For the sake of completeness, we also briefly describe the last subarea of coalition formation, weighted voting games, which are considered by [Chalkiadakis and Wooldridge, 2016] in [Brandt et al., 2016b, p.377–395]. We closely follow their explanation.

Weighted voting games are a special form of cooperative games that can be used to model decision-making situations in which a set of voters make a binary (yes/no) decision on a particular issue. Each voter has a numeric weight, and the decision is carried out if the number of voters favouring this decision meets or exceeds a given threshold (i.e. the quota).

Next, we describe a short example. Following [Bilbao et al., 2002], [Taylor, 1995] and [Taylor and Zwicker, 1999], both the U.S. federal system and the voting system of the European Union can be modelled as weighted voting games, where, in the first case, the participating agents are the president, vice president, senators and representatives. The senators have zero weight regarding the House of Representatives game and the representatives have zero weight regarding the Senate game, and the president has a non-zero weight in both games. In the case of the European Union voting system, we have a three-weighted voting game: each member state is a player. The law requires the support of 50% of the member countries, 62% of the population and 74% of the European Union commissioners. In the first component of the game, the weights for the member states are assigned according to the number of commissioners of the respective member state. In the second component, we have a simple majority game where every state has one vote. In the third component, the weights for the states are proportional to the population of the respective state. [Brandt et al., 2016b, p.393]

[Brandt et al., 2016b, p.378, verbatim] formally define cooperative games as follows:

Definition 3. *A cooperative game, G, is given by a pair $G = (N, v)$, where $N = \{1, ..., n\}$ is the set of players of the game and $v : 2^N \to \mathbb{R}$ is the characteristic function of the game. A cooperative game $G = (N, v)$ is simple if $v(C) \in [0, 1]$ for all $C \subseteq N$. In this case, we say $C \subseteq N$ are winning if $v(C) = 1$ and losing otherwise. A simple game is nontrivial if $v(N) = 1$.*

[Brandt et al., 2016b, p.379, verbatim] formally define a weighted voting game as follows:

Definition 4. *A weighted voting game G with a set of players $N = \{1, ..., n\}$ is given by a list of weights $w = (w_1, ..., w_n) \in \mathbb{R}^n$ and a quota $q \in \mathbb{R}$. The characteristic function $v : 2^N \to \{0, 1\}$ of the game is defined as follows with coalition $C \subseteq N$:*

$$v(C) = \begin{cases} 1 & \textit{if} \sum\limits_{i \in C} w_i \geq q \\ 0 & \textit{otherwise} \end{cases} .$$

2.1.4 Group activity selection problems

Another topic considered in Computational Social Choice is the selection of activities for a group of agents. Our situation is similar, because we need to select POIs for a group of traffic participants. The explanations regarding group activity selection problems in this section closely follow the formulations by [Darmann and Lang, 2016] in [Endriss, 2017, p.87–101].

Group activity selection means that one or several activities are selected for a set of agents, and the agents are assigned to one of the different selected activities according to their preferences. In group activity selection problems, one considers agents' preferences over both activities and the number of participants for an activity.

The authors give as an example a group activity selection problem known as the "Dagstuhl group activity selection problem" among the Dagstuhl staff. It is described as follows in [Endriss, 2017, p.87]. The organisers of a workshop plan to have a set of group activities during a free afternoon. They have the following conditions:

- The activities are held in parallel, and each participant can take part in at most one activity.
- The possible activities include hiking, a bus trip to a nearby city and a table-tennis competition.
- There can be several hiking groups, and analogously, several buses can be rented for the trip. As for the table-tennis competition, there can only be one group, because there is only one table.
- Because the cost of renting bus(es) must be shared among the participants, a bus trip with more participants will be preferred to a bus trip with fewer participants.
- Considering the table-tennis competition, is is assumed that the participants prefer the number of participants to be neither too small nor too large, because the participants neither will want to wait long to play nor to play without breaks.

Furthermore, the authors explain that there are several natural variations of this problem. For example, [Lee and Shoham, 2015] defined the *stable invitation problem* as follows: There is only one activity for which the organiser looks for a set of invitees. The potential invitees have preferences about the number of invitees; they can also have preferences about the other invitees. [Lee and Shoham, 2014] further defined an extension of this problem where the invitee preferences additionally also depend on the date the event takes place.

Darmann and Lang explain that if there are no distinct activity types, which is equivalent to every group being assigned to the same activity, one gets an *anonymous hedonic game*, meaning that agents have only preferences regarding the number of agents in the group they belong to. If the agents can additionally have preferences about the identity of their group members, one gets the general case of a *hedonic game*. In the following, some problems are formally defined considering a set of agents $N = \{1, ..., n\}$.

Definition 5. ***Activities and Assignments [Endriss, 2017, p.88f., verbatim].*** *We consider a set of activities* $A = A^* \cup \{a_\emptyset\}$, *where* $A^* = \{a_1, ..., a_m\}$. *Activity* $a_\emptyset$ *is called the void activity. An agent being assigned to* $a_\emptyset$ *will not participate in any concrete activity.*

An assignment for (N, A) is a mapping $\pi : N \rightarrow A$. π^0 denotes the set of agents i such that $\pi(i) = a_\emptyset$ and for each $j \leq m, \pi^j$ is the set of agents i such that $\pi(i) = a_j$. The coalition structure induced by π is defined as $CS_\pi = \{\{i\}|i \in \pi^0\} \cup \{\pi^j|1 \leq j \leq m, \pi^j \neq \emptyset\}$.

Definition 6. ***Alternatives and Preference Profiles [Endriss, 2017, p.89, verbatim]*** *Agents have preferences that bear both on the activity to which they will be assigned and on the set of agents who will participate in the same activity. An alternative for agent i is either an $a_\emptyset$ or a pair $(a, S) \in N_i$, where N_i is the set of all subsets of N containing i. X_i is the set of alternatives for i.*

Each agent i has some preferences over X_i. A preference relation for agent $i \succsim_i$ is a reflexive and transitive order over X_i. A preference profile is an n-tuple $P = (\succsim_1, ..., \succsim_n)$ where $\succsim$ is a preference relation for i.

In the following, several classes of Group Activity Selection Problems (GASPs) are described.

Hedonic Games As mentioned in Subsubsection 2.1.3.2, hedonic games cannot only be viewed as coalition formation, as in [Aziz et al., 2016], but also as group activity selection problem. As group activity selection problem, hedonic games can formally be defined as follows:

Definition 7. ***Hedonic Games [Endriss, 2017, p.89, verbatim]*** *We say that $\succsim_i$ is activity-independent if $(a, S) \sim_i (a', S)$ for all activities a, a' and coalitions S. i's preference relation $\succsim_i$ depends only on the set of agents in i's coalition and not on the activity to which i is assigned. When $P = (\succsim_1, ..., \succsim_n)$ and every $\succsim_i$ is activity-independent, (N, A, P) degenerates into a hedonic game, where agents only care about which agents are in their coalition [Dreze and Greenberg, 1980, Banerjee et al., 2001, Bogomolnaia and Jackson, 2002].*

Anonymous Group Activity Selection: GASP [Endriss, 2017, p.90] In anonymous group activity selection, agents care only about the activity to which they belong and the number of participants in that activity. For consistency with existing literature, Darmann and Lang referred to the general model, where agents care about both the assigned activity and the identity of the agents in their group, as *generalised group activity selection* and the anonymous model as *group activity selection* (GASP). If in a GASP, all agents have approval-based preferences, it is called an *approval-based group activity selection problem*, or an a-GASP [Darmann et al., 2012]. If all agents have as preferences complete linear orders, the GASP is called an *ordinal group activity selection problem*, or an o-GASP [Darmann, 2015].

Stable invitations [Endriss, 2017, p.91f.] In the case where only one non-void activity exists, we have a *stable invitation problem.* If the preferences are anonymous, we have an *anonymous stable invitation problem.*

Group Activity Selection on Social Networks: gGASP [Endriss, 2017, p.92] Another problem is the one described by [Igarashi et al., 2017a, Igarashi et al., 2017b]. It is a constrained group activity problem where the agents are linked through an undirected graph G which represents social interactions. A coalition of agents being assigned to an activity is feasible only if the coalition is connected with respect to this graph. GASP corresponds to the special case where G is a complete graph.

Simplified GASP [Endriss, 2017, p.92] Additionally to the above-presented problems, there is an even simpler one. When agents' preferences depend only on the activity to which they are assigned, we have a *simplified GASP*.

Simplified GASP is similar to our considered problem insofar as the agents' preferences only depend on the POIs they are assigned to. Unlike in group activity selection, where each agent is assigned to at most one activity, we consider a scenario where agents can be assigned to several POIs.

Overview of problem classes Figure 2.2 which is taken from [Darmann and Lang, 2016] in [Endriss, 2017, p.92] depicts the relation between the different group selection activity problems.

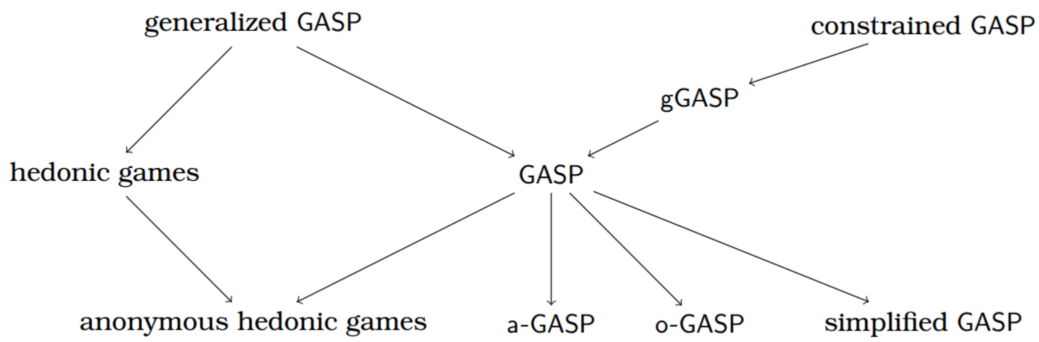

Figure 2.2: GASP: overview

Solution Concepts There are different solution concepts for the above-defined group activity selection problems.

Maximum Individual Rationality and Pareto Optimality[Endriss, 2017, p.94]: A possible solution approach is to find a maximum individually rational assignment. This would be an individually rational assignment that maximises the number of agents assigned to a non-void activity. Another approach is to find a Pareto-optimal partition for a hedonic game. This would require a partition such that it could not be improved for some agents without other agents being worse off.

Stability Notions [Endriss, 2017, p.94ff.]: Another solution concept is to consider stability regarding incentive for agents to deviate from an assignment. For hedonic games and group activity selection, there exist several notions of stability: Nash, individual, contractual individual, core and strict core.

2.1.5 Allocation of indivisible, non-shareable resources

The assignment of POIs to agents is reminiscent of another area considered in Computational Social Choice (ComSoc), namely the allocation of indivisible, non-shareable resources. These resources can be single items or bundles. In this section, we follow the definitions for auctions and multi-agent resource allocation given in [Rothe et al., 2012]. Note that in Multi-agent Resource Allocation (MARA) and auctions, each item or bundle can be allocated exactly once. However, in our scenario, a POI can be assigned to several agents.

2.1.5.1 Single-item auctions

In this section, we explain single-item auctions following [Rothe et al., 2012, p.327-332]. To allocate single objects, one needs an *allocation procedure*, which can be distributed or centralised. While in distributed allocation procedures, the agents follow a negotiation protocol to agree on an allocation of the items amongst themselves, in centralised allocation procedures, a central authority conducts the allocation based on the submitted individual valuations of the agents regarding the objects. *Auctions* are centralised allocation procedures. Some examples of single-item auctions are first-price sealed, English, Dutch, Vickrey and all-pay auctions.

2.1.5.2 Multi-agent resource allocation (MARA)

In this section, we give a brief description of the area MARA, using the definitions and explanations in [Rothe et al., 2012, p.333f.]. MARA investigates the allocation of bundles of resources.

In order to define MARA settings, we assume a set of agents $A = \{a_1, ..., a_n\}$ and a set of resources $R = \{r_1, ..., r_m\}$. As described in [Rothe et al., 2012, p.333f., translated], "*[f]or each bundle $B \subseteq R$ of resources, the utility function $u_i : \mathcal{P}(C) \rightarrow \mathbb{Q}$, defines the utility of agent $a_i \in A$ independently of the utility values of other agents.*" Agents can have preferences over single bundles, either ordinal or cardinal. As described in [Rothe et al., 2012, p.334, translated], "*[a]n ordinal preference over R is based on a binary relation $\succeq$ over R, which is reflexive and transitive and usually, but not necessarily, complete. $B \succeq_i B'$ means that bundle B has at least the same value as B' to agent a_i. In contrast, a cardinal preference over R is defined by the utility function of agent a_i.*" Note that each cardinal preference u_i induces an ordinal preference $\succeq_i$ by $B \succeq_i B' \leftrightarrow u_i(B) \leq u_i(B')$. Based on these assumptions, *allocations* can be formally defined as follows [Rothe et al., 2012, p.334, translated and adapted]:

Definition 8. ***Allocations*** *An allocation for A and R is a mapping $X : A \rightarrow \mathcal{P}(R)$ with $X(a_j) \cap X(a_k) = \emptyset$ for any pair of agents a_j and a_k where $j \neq k$. That is, $X(a_i) = B \subseteq R$ is the bundle of resources allocated to agent a_i, and the bundles different agents receive are disjunct. [Let $U = \{u_1, ..., u_n\}$ denote the set of all utility functions u_i for the agents.] Then, the triple (A, R, U) is called a MARA setting.*

2.2 Transport applications

As depicted in the classification in Figure 2.1, we apply voting approaches in the traffic domain. Thus, we need to consider the relationship to existing transport applications. Here, two traffic research areas are relevant, because they deal with groups of traffic participants: ridesharing and platooning research.

2.2.1 Ridesharing

Because we consider traveller groups, our research is related to ridesharing research. In the following, we describe a selection from a broad range of works regarding ridesharing.

[Furuhata et al., 2013], defines ridesharing as follows: Each traveller has a demand for their trip consisting of the origin and the destination, and ridesharing is a joint-trip of at least two participants sharing a vehicle.

As [Agatz et al., 2012, p.297] outlined, most studies on ridesharing considered one or a combination of the following goals; we closely follow their formulations:

- *Minimise system-wide vehicle miles*: the system-wide vehicle miles are the total vehicle miles driven by all participants travelling to their destinations either by ridesharing or by driving alone. This goal is important at the societal level, because it helps reduce pollution and congestion. It is also compatible with minimising total travel costs, which is an important consideration for the participating drivers and riders and the ride-share provider.

- *Minimise the system-wide travel time*: the travel time is the time spent in the vehicle while travelling between origin and destination. Apart from being an important consideration for the participants, it is also an important measure at the societal level, because vehicle emissions relate to both vehicle miles and vehicle speeds.

- *Maximise the number of participants*: this goal maximises the number of satisfied drivers and riders in the system. This goal is relevant for private ride-share providers whose gain is linked to the number of successful ride-share arrangements. The matching success rate is also an important performance indicator for users of ride-share services, and a high success rate can lead to larger participant pools in the future.

[Furuhata et al., 2013, p.34f.] listed and explained classes of ridesharing; we closely follow their formulations:

Dynamic ridesharing: here, an automated process of ride-matching (routing, scheduling, and pricing) between drivers and passengers is provided on very short notice or even en route [Agatz et al., 2012].

Carpooling: this defines services for commuters who share transportation to work in a private vehicle with other workers, see for example [Ferguson, 1997].

Long-distance ride-match: this defines services for travellers taking long-distance trips. Typically, long-distance travellers have more flexible travel schedules than on-demand travellers and commuters. Some providers in this class offer a list-based search as an alternative search choice. Users of this service specify the departure region and then search for candidates in the list. This means rather than specifying a preferred departure time, they choose their departure time based on availability of rides.

One-shot ride-match: this is a hybrid of carpooling and long-distance ride-matching, where choices for a ride-matching method are offered according to trip types. The matching methods are similar to those used in carpooling and long-distance ridesharing. Matching agencies in this class provide not only the search criteria OD-Pair and Time, but also additional search criteria lists and keywords, as well as routes.

Bulletin-board: here, ridesharing opportunities are provided based on notice boards. Some providers try to keep the ridesharing offers and requests as flexible as possible and base the decision on what type of information should be included in the offers and requests on the users, meaning that most ridesharing conditions are fixed by negotiation among the users. See also [Beroldo, 1991].

Flexible carpooling: this form is semi-organised, where the users meet at predetermined locations to organise shared rides, see for example [Kelley, 2007].

Note that unlike in ridesharing, in our scenario, the destinations or activities are not fixed a priori, but they must be determined based on the preferences of the travellers over all possible destinations. Additionally, rather than considering space-time interdependencies for the determination of travel groups, we consider routing a downstream problem.

Generally, there are two types of ridesharing: trip-based and activity-based, as described in the following subsections.

2.2.1.1 Trip-based vs. activity-based ridesharing

[Wang et al., 2016] considered activity-based ridesharing, as opposed to trip-based ridesharing. In the following paragraphs, we briefly explain the differences and describe the solution approach for activity-based ridesharing by [Wang et al., 2016].

Trip-based ridesharing A common assumption of trip-based ridesharing is that a traveller has a demand for a trip consisting of an origin and a destination. In trip-based ridesharing, the trips are defined a priori, and journeys from one unique location to another are matched. [Wang et al., 2016, p.2]

Activity-based ridesharing Analogue to defining trips a priori in trip-based ridesharing, activity-based travel planning categorises activities as fixed (e.g. workplace-related activities) and flexible activities (e.g. shopping)[Wang et al., 2016, p.1].

[Wang et al., 2016] proposed an activity-based ridesharing algorithm (ABRA), which aims to efficiently increase matching rates by considering alternative destinations for flexible activities while keeping detour costs tolerable. As the authors explained, ABRA includes two steps:

1. Build a pool of alternative destinations and trips for the targeted activities based on the trip's space–time budgets

2. Find feasible matchings considering these alternatives in addition to the original ones.

Matching is conducted as static pre-planning with all daily schedules as input. A daily schedule includes one or multiple trip chains consisting of multiple trips. [Wang et al., 2016, p.2]

In their study, they considered a centralised approach to show the theoretical potential of the activity-based approach. ABRA centrally computes the global maximum of all feasible matches. They pointed out that cheaper (i.e. heuristic or decentralised) solutions may exist.[Wang et al., 2016, p.2]

An important assumption in their paper was that each person has a complete list of full-day activities with a predefined running order. Thus, the activity sequence and planning were outside of their study scope. [Wang et al., 2016, p.3]

2.2.2 Platooning

Another traffic research area relevant in the context of the considered topic is platooning. The Cambridge Dictionary defines platooning as "*a method of connecting vehicles either physically or using computer technology so they can travel close together in a group, as a way of saving space, fuel, or money*" [Cambridge Dictionary, 2021]. In this dissertation, we assume platoons consisting of pods which are physically coupled.

In recent years, the relevance of platooning has increased. [Bergenhem et al., 2012] provided an overview of platooning projects until 2012. The projects aimed at utilising platoons in order to make improvements such as increasing fuel and traffic efficiency, safety and driver comfort. Although the focus was often placed on truck platooning and/or highways, there were also projects that focused on urban scenarios. For example, the GCDC (Grand Cooperative Driving Challenge) of 2011 considered both urban and highway scenarios [Mårtensson et al., 2012]. In the GCDC, one challenge was to increase road throughput by reducing the spacing between vehicles.

[Haas and Friedrich, 2017] gave a short outline of platooning research until 2017. They determined that platoons were mostly used for freight purposes [Ramakers et al., 2011, Alam et al., 2015, Liang et al., 2016] and were considered promising in terms of road utilisation, road safety [Varaiya, 1993, Alam et al., 2015] and reduction of fuel consumption [Tsugawa et al., 2011, Janssen et al., 2015, Van De Hoef et al., 2015]. While early studies regarding platooning techniques focused on Adaptive Cruise Control (ACC) systems, later studies focused on Cooperative Adaptive Cruise Control (CACC) systems, which enable increasing the road throughput even further [Ploeg et al., 2011]. While there have been many studies on the environmental benefits of using platoons, see e.g. [Bonnet and Fritz, 2000, Al Alam et al., 2010, Liang et al., 2013], there have been comparably few studies regarding traffic-related aspects (e.g. travel time). In this context, [Haas and Friedrich, 2017] considered autonomous connected vehicle platoons for city logistics, focusing on the effect of different platoon configurations on travel time.

The target quantities considered in the works reviewed by [Bergenhem et al., 2012] and [Haas and Friedrich, 2017] included fuel and traffic efficiency, safety, driver comfort, road throughput and travel time. In our approach, we consider other target quantities: preference dissatisfaction, group size and organisational effort.

2.3 Collective decision-making in transport

Next, we consider approaches that combine both domains/areas, i.e. investigate collective decision-making in traffic applications.

These approaches include reservation-based approaches in road-traffic management (Subsection 2.3.1), a model for urban mobility social simulation (Subsection 2.3.2), joint decisions in transport (Subsection 2.3.3), the joint-travel problem (Subsection 2.3.4), tactical-level decision-making for platoons of autonomous vehicles using auction mechanisms (Subsection 2.3.5) and existing voting applications in traffic management (Subsection 2.3.6).

2.3.1 Reservation-based approaches in traffic management

As prominent example for collective decision-making in traffic management, we highlight the works by Dresner, Stone, Vasirani and Ossowski on reservation-based approaches. In these approaches, driver agents need to reserve resources by communicating with intersection managers.

[Dresner and Stone, 2008] proposed a multi-agent approach for autonomous intersection management. In this approach, drivers crossing intersections are coordinated with the help of intersection managers. At each intersection, the driver agents call ahead in order to reserve a block of space-time in the intersection. According to an intersection control policy, the respective intersection manager decides whether to grant or reject reservation requests.

Based on Dresner and Stone's approach, [Vasirani and Ossowski, 2009] proposed a market-inspired approach to reservation-based urban road-traffic management. They extended the approach to networks of intersections using market-inspired control methods. In the new approach, several intersection managers can act as teams.

Note that in these reservation-based approaches, there is no clear separation of subgroups of vehicles that need to agree on common plans, as the vehicles at the intersection are constantly replaced by the following vehicles. In contrast, we consider in our scenario clearly defined groups of vehicles that need to agree on common destinations.

2.3.2 Model for urban mobility social simulation

[Grimaldo et al., 2012] developed a market-based model for urban mobility social simulation. In their simulation, they implemented social decisions made by the inhabitants of a town about how to get to work, such as travelling by train, using their own car or via car sharing.

They compared outcomes produced by societies of individualist and egalitarian agents in terms of average travel time, usage of urban transportation and CO2 emission. For their agent-based social simulation of this scenario, they used the Multimodal Agent Decision-making (MADeM) model described in [Grimaldo et al., 2008].

They followed a preference network approach in which the agents expressed their preferences using utility functions so that personal attitudes were represented by the differential utilitarian importance they placed on the utilities of others. They describe the MADeM approach as a market-based mechanism for social decision-making which is capable of simulating different kinds of social welfare, such as elitist or utilitarian as well as different social attitudes, such as egoism or altruism. [Grimaldo et al., 2012, p.150]

In this approach, agents are required to express their preferences with regard to the different solutions for the respective decision problem. Because MADeM is based on the MARA theory, it represents each solution as a set of resource allocations. MADeM can consider both tasks and objects as resources to be allocated and uses first-sealed one-round auctions as the allocation procedure. The winner determination uses a multi-criteria winner determination problem to merge the collected preferences according to the kind of agent or society simulated. [Grimaldo et al., 2012, p.151]

In the considered scenario, each inhabitant is represented by an agent that uses the J-MADeM library as defined in [Grimaldo et al., 2010] to make decisions that balance individual and social preferences, and inhabitants are randomly organised in decision groups comprising family, friends, neighbours, etc. The maximum capacity of cars is set to four people, which is also the size of the decision groups. An allocation is a travel alternative, and J-MADeM collects the preferences of the group for every possible alternative. Every agent computes three utility functions for each allocation: monetary cost, travel time, and ecological impact. For winner determination in this scenario, J-MADeM used the utilitarian collective utility function of social welfare to reflect the aggregate impact of the type of allocations considered. [Grimaldo et al., 2012, p.152f.]

The authors considered three world views. Agents with a *hierarchical* worldview assumed that nature was stable in most cases, but that it can collapse if the capacity is exceeded. *Egalitarian* agents assumed that nature is highly unstable, each human intervention can lead to a collapse, and *individualists* assumed that nature provides plenty of resources and will remain stable. To model these types, utility weights were used. [Grimaldo et al., 2012, p.154]

In [Grimaldo et al., 2010], the bidding phase is described in more detail. Because the auctioneer informs the bidders about both the task-slot allocation and the used utility functions, bidders only need to compute the specified utility functions and return the values for each auction to the auctioneer.

In J-MADeM, each solution to the considered decision problems is represented as a set of resource allocations. In our model, we consider the agreement on a common decision via voting, not resource allocation. We focus on comparing several voting algorithms. Furthermore, we consider defined vote forms, whereas the MADeM model can use arbitrary utility functions. Note that some elections could probably also be modelled using the MADeM model: returning the utility values to the auctioneer is similar to submitting a vote. One could define some vote forms using J-MADeM by only allowing certain utility values to represent different vote forms, for example [0,1] utility functions for Approval votes. However, the winner determination in J-MADeM is conducted via a multi-criteria approach.

2.3.3 Joint decisions in transport

[Dubernet and Axhausen, 2012] and [Dubernet and Axhausen, 2013] developed a multi-agent simulation for modelling joint-travel behaviour. In this section, we closely follow their formulations.

[Dubernet and Axhausen, 2012] modelled behaviour for the case where several individuals may travel in the same private vehicle. As they explained, this is both important for detailed simulations of household behaviour and for the evaluation of policies, including incentives to perform carpooling. [Dubernet and Axhausen, 2012, p.1]

[Dubernet and Axhausen, 2013] presented an approach for simulating joint decision processes, which uses the MATSim framework.

Their model was based on an activity-based approach, where the fact that travel is oriented toward a goal is explicitly taken into account by assigning agents plans which consist of located activities. The agents then travel between those activities in a simulated network. [Dubernet and Axhausen, 2013, p.2]

The authors described different approaches for joint decision modelling, for example the random utility approach. This approach has been widely used in transportation research [Ben-Akiva et al., 1985]. In the random utility approach, each alternative is associated with a numerical utility comprising a systematic part, which is an expectation value, and a random error term representing unobserved variability. The higher the probability that the utility of an alternative is higher than the utility of all other alternatives, the higher the probability is that an agent chooses this alternative. When considering the choice problem for joint decision as a group utility maximisation problem, using random utility is a straightforward approach. For several decades, the random utility approach has been used for group schedule generation for activity-based transport simulation, mainly for household-schedule generation, Note that, because the choice set has high dimensionality with both discrete dimensions (e.g. activity types, joint activity participation, sequence of activities and modes) and continuous dimensions (e.g. activity duration), different choice dimensions have been considered by different researchers. [Dubernet and Axhausen, 2013, p. 4]

In previous approaches for coping with individual coordination, some relied on the actual simulation of bargaining processes, whereas others considered the utility-based optimisation of a joint plan. Dubernet and Axhausen proposed a utility-based approach based on the joint-plan concept, while allowing representation of coordination in arbitrary social structures. [Dubernet and Axhausen, 2013, p. 6f.].

In their works, Dubernet and Axhausen did not focus on comparing given voting algorithms when modelling joint-travel behaviour, whereas in our approach, we focus on comparing the effects of several voting algorithms.

2.3.4 Joint-travel problem

[Liao, 2017] considered the Joint-Travel Problem (JTP), which is an extension of the shortest-path problem. Liao proposed a space-time multistate super-network to tackle JTP for the situation that one joint activity is conducted in a time-dependent context. Starting with a two-person JTP in a unimodal network, it was extended for both multimodal and multiperson transport.

[Liao, 2017, p.2] described the problem as follows: "*JTP [...] aims to find the optimal joint path for a travel group. A joint path involves multiple individual paths with multiple origins and destinations. Some parts of the individual paths may be shared by a subset of the travel group.*"

When a group of individuals decides to travel together, they must agree on where and when to meet and depart. If more than two persons are involved, decisions regarding the meeting/departing sequence are also necessary. [Liao, 2017, p.2f.]

In an example of joint-travel scheduling for one joint activity, [Liao, 2017, p.5] described how, for the two-person case, two persons might agree on a meeting point from two alternatives, an activity location from two alternatives and a departing point from two alternatives, i.e. the algorithm can be used to choose one activity location from several alternatives.

Liao defined the disutility of an individual or a joint path as the sum of the associated link disutilities. To minimise group disutility, JTP aims to find the optimal joint path that includes choices regarding departure time, route, meeting/departing time/point/sequencing, as well as activity location and duration for the travel group. [Liao, 2017, p. 6]

This paper focuses on finding the optimal joint path for different variants, namely, for two or more persons for no joint activity or one joint activity, considering both unimodal and multimodal transport.

Rather than determining the optimal path for joint activities, we focus on comparing the effect of several voting algorithms on system and user goals. Liao does not explicitly consider the case in which the group conducts more than one joint activity, whereas we aim to achieve consensus on several destinations. Furthermore, we assume a fixed assembly point rather than the question of where the agents should meet and depart. Additionally, we do not consider time windows.

2.3.5 Tactical-level decision-making for platoons of autonomous vehicles using auction mechanisms

[Kokkinogenis et al., 2019] proposed the application of market-based mechanisms for establishing cooperative behaviour in traffic scenarios involving autonomous vehicles. This paper aimed to show the suitability of well-known auction rules as a mechanism for tactical-level collective decision-making in platooning applications by comparing two auction rules as proof of concept: first- and second-price sealed bid auctions. They evaluated the effects of the auction rules on the quantities monetary flows, platoon welfare from utilitarian and egalitarian perspectives (measured in average and minimum utility) and time to consensus. As a concrete scenario, they considered an already formed platoon whose members needed to come to an agreement regarding two contexts: cruising speed and route. Each platoon member had a preferred cruising speed and route for the same origin and destination, a certain willingness to pay for each resource and an endowment that reflected the available amount of monetary units. In the context of route choice, each platoon member considered the sequences of vertices which represent the preferred route and an alternative route. The Hamming distance was used to measure the similarity between the two routes. This was then compared with the desired maximum similarity between routes to compute the utility for the route context.

In contrast to the approach of Kokkinogeneis et al., we use voting instead of auction rules to agree on destinations for the platoon. Furthermore, instead of assuming already formed platoons, we take consider several grouping algorithms.

2.3.6 Existing voting applications in traffic management

In this subsection, we provide an overview of existing voting approaches for traffic applications. Those include institutional consensus in vehicular networks, leader elections in platoons, voting-based approaches for autonomous vehicles coordination and group aggregation strategies for tourism.

2.3.6.1 Institutionalised consensus in vehicular networks

An example of the application of voting algorithms in traffic management is found in the work by [Sanderson and Pitt, 2012, Sanderson et al., 2012]. [Sanderson and Pitt, 2012] proposed the management of consensus formation in open, decentralised and resource-constrained systems (e.g. vehicular networks) using self-organising electronic institutions. To this end, they adapted the Paxos algorithm for fault-tolerant consensus in distributed databases [Lamport, 1998] to the Institutionalised Paxos Consensus (IPCon) algorithm for robust collective choice in electronic institutions. They aimed at fault-tolerance in regard to inadvertent failure, non-compliant behaviour and environmental changes. In their paper, they demonstrated that the IPCon algorithm is a viable method for coordination, consensus formation and collective choice in self-organising MAS using electronic institutions.

[Sanderson et al., 2012, p.77] proposed a three-layered framework for providing intelligence at different levels of driving, which they described as follows:

1. The micro level is concerned with the actual execution of driving, which includes safety issues as well as fulfilling drivers' and passengers' goals and preferences such as driving speed or safety distance.

2. The meso level is concerned with collective decision-making within groups or clusters of vehicles, where the clusters are created according to some relationship among the vehicles (e.g. clusters form at every junction) and the vehicles should reach agreements on decisions such as who passes first or which speed a platoon travels at

   The authors assumed that, owing to the high dynamics of traffic, vehicles are constantly leaving and joining clusters. The meso level relies on V2X communication for both information exchange between vehicles regarding their intentions and for carrying out negotiations to reach agreements

3. The macro level deals with infrastructure or system-wide goals, such as reducing congestion and pollution, making efficient use of the road network and managing the interaction of the network.

As an example, the authors demonstrated how vehicles in a cluster could reach a consensus on several institutional facts via IPCon to govern their behaviour.

Among other things, IPCon considers that the current value may not always be the best one. Thus, it may be necessary to revise a chosen value. In the case of platooning, the vehicles may agree on the travel speed, which would then need to be adjusted depending on changes in congestion. [Sanderson et al., 2012, p.80]

The focus of Sanderson et al. is on robust implementation of consensus. In contrast to this, in our scenario, our aim is to compare several voting approaches with regard to user and system goals.

2.3.6.2 Leader election in platoons

There are several papers which use voting algorithms for leader election in platoons; [Teixeira et al., 2019, p.3f.] gave a short overview of some works in this vein:

[Singh et al., 2018] implemented an incentive-based election protocol for agreeing on a platoon. Furthermore, [Ferreira and d'Orey, 2011] used voting mechanisms to elect platoon leaders in the context of vehicle coordination for intersection-management scenarios. [Asplund et al., 2017] used a consensus mechanism for the vehicle leader election to achieve vehicular coordination.

In our approach, we also use voting in platooning, but the application is different: we use voting to determine common destinations for platoons.

2.3.6.3 Autonomous vehicles coordination through voting-based decision-making

[Teixeira et al., 2018] proposed the application of ComSoc mechanisms to establish co-operative behaviour within traffic scenarios involving autonomous vehicles and used an integrated simulation platform comprising the agent-based platform LightJason, the microscopic traffic simulator Simulation of Urban MObility (SUMO) and the network simulator Objective Modular Network Testbed in C++ (OMNeT++). They focused on the usage of single-winner voting rules to reach consensus on a platoon's cruising speed, assuming that platoon formation is given. They investigated the effect of the voting rules on the quantities time to consensus and average platoon utility under unreliable communication.

In contrast to Teixeira et al, we assume an ideal situation with perfect communication. We focus on comparing the inherent effects of several voting algorithms on the target quantities preference dissatisfaction, group size and organisational effort. In our approach, we consider committee elections and several grouping algorithms.

2.3.6.4 Simulating collective decision-making for autonomous vehicle coordination enabled by vehicular networks

Extending their research in [Teixeira et al., 2018], [Teixeira et al., 2019] presented a simulation framework that combined vehicular, communication and agent-based simulators for testing the effects of bargaining and voting mechanisms under realistic conditions with kinematic and communication constraints. The framework was benchmarked using lane-merging and platoon scenarios. In the lane-merging scenario, a bargaining mechanism was used. In the platoon scenario, the members used single-winner voting with an iterative process as in [Teixeira et al., 2018] to agree on speed and single-round committee voting to agree on the route. For the committee election, the authors compared the effects of Minisum- and Minimax-Approval.

Teixeira et al. did not consider the Minisum-Ranksum voting rule. Furthermore, they considered different target quantities for the committee elections than in our approach. They considered the agents' perceived satisfaction with the chosen route and the channel busy ratio (CBR) (i.e. the ratio of time during which the communication channel is busy). The latter is a target quantity that is not relevant to us. Instead, we focused on group size, preference dissatisfaction and organisational effort. Furthermore, [Teixeira et al., 2019] did not consider algorithms for group formation.

2.3.6.5 Group aggregation strategies for tourism

[Najafian et al., 2020] evaluated several group aggregation strategies for tourism in order to recommend POIs to groups of tourists. They considered four different aggregation strategies, focusing on the target quantities perceived individual satisfaction, perceived group satisfaction, perceived fairness and user acceptance.

In our approach, we focused on other target quantities: preference dissatisfaction, group sizes and organisational effort. Furthermore, Najafian et al. did not consider group formation.

2.4 Multi-agent systems (MAS)

In this dissertation, we model travellers/voters and other components as agents. Next, we describe agent definitions/paradigms, agent-based modelling/simulation, the LightJason framework and some frameworks that combine microscopic traffic simulation with multi-agent simulation.

2.4.1 Agent definitions and paradigms

Adapted from [Wooldridge and Jennings, 1995], [Wooldridge, 2009] defined an agent as "*a computer system that is situated in some environment and that is capable of autonomous action in this environment to meet its design objectives*". According to this model, an agent uses sensors to perceive its environment and effectors to affect its environment based on observations. [Wooldridge, 2009] illustrated this using Figure 2.3 (which is adapted from [Russell and Norvig, 1995, p.32]).

[Jennings, 2000, p.280] described agents as

1. *"clearly identifiable problem-solving entities with well-defined boundaries and interfaces*

2. *situated (embedded) in a particular environment - they receive inputs related to the state of their environment through sensors, acting on the environment using effectors*

3. *designed to fulfil a specific purpose - they have particular objectives (goals) to achieve*

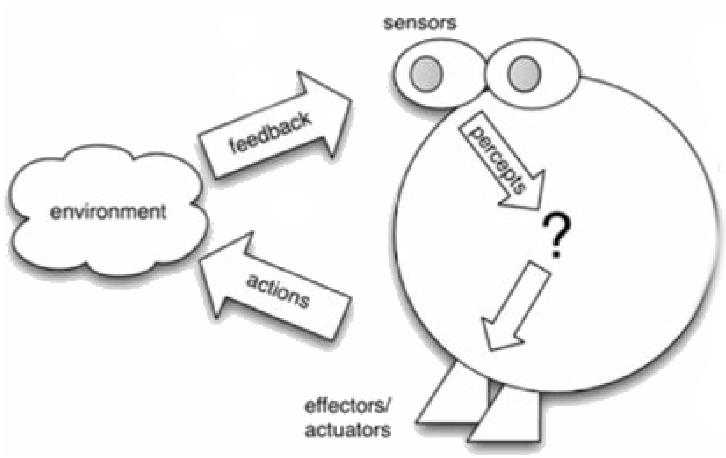

Figure 2.3: Agent situated in an environment

4. *autonomous - they have control over both their internal state and their own behaviour*

5. *capable of exhibiting flexible problem-solving behaviour in pursuit of their design objectives - they need to be both reactive (able to respond in a timely fashion to changes that occur in their environment) and pro-active (able to act in anticipation of future goals)"*

Thus, a MAS is a system consisting of autonomous, flexible problem-solving entities embedded in a particular environment which are designed to fulfil a specific purpose.

2.4.2 Agent-based modelling and simulation

According to Bordini et al., the predominant approach for implementing intelligent or rational agents is the Belief–Desire–Intention (BDI) approach [Bordini et al., 2005]. The BDI model was originally developed in 1987 by [Bratman, 1987] as a psychological-philosophical theory considering the relation of intentions, plans and practical reasoning. As [Georgeff et al., 1998] put it, *Beliefs* represent an agent's knowledge of the world, and *Desires* (or goals) represent some desired end state. *Intentions* are particular courses of actions to which agents commit to handle certain events, see [Bordini et al., 2005].

An early BDI architecture was the Intelligent Resource-bounded Machine Architecture (IRMA), as described by [Bratman et al., 1988]. [Georgeff and Lansky, 1987] developed a system called Procedural Reasoning System (PRS) for BDI agents, and Figure 2.4 (taken from [Georgeff and Lansky, 1987, p.679]) shows its basic structure.

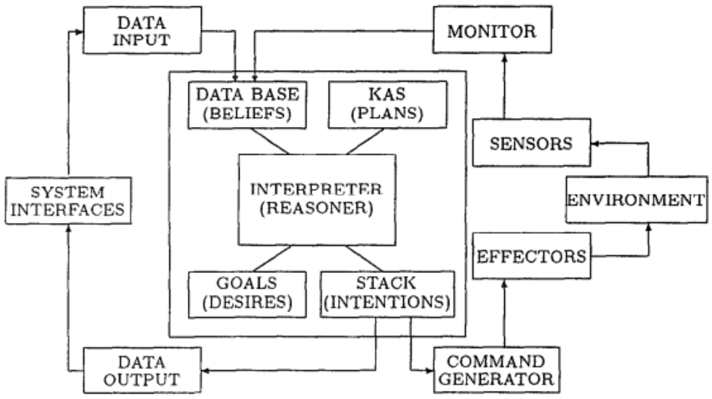

Figure 2.4: Procedural Reasoning System

PRS has several components, namely a database with current beliefs (or facts about the world), a set of current goals (or desires), a set of procedures (or knowledge areas (KAs)) describing how to achieve goals or to react to certain situations by conducting sequences of actions and tests and an interpreter (or inference mechanism) to manipulate the other components. The process stack containing all currently active procedures represents the current intentions of the system. [Georgeff and Lansky, 1987, p.678]

[Weiss, 1999] describes some properties of PRS as follows. In PRS, each agent performs mean-ends reasoning by using plans from its library and achieves deliberation by using meta-level plans. Meta-level plans can modify intention structure in order to shift the focus of practical reasoning in an agent. Furthermore, PRS uses Prolog-like facts, i.e. atoms of first-order logic for representing beliefs.

There are several implementations of PRS, for example the AgentSpeak language (AgentSpeak(L)). This language is used in the commonly used BDI interpreter *Jason*, which has been developed by Bordini et al. [Bordini and Hübner, 2005], [Bordini et al., 2005] and [Bordini and Hübner, 2009].

2.4.2.1 LightJason framework

Another BDI multi-agent framework is LightJason as described by Aschermann et al. in [Aschermann et al., 2016], which is used to implement the multi-agent simulation tool for this dissertation. As Aschermann et al. explain, in LightJason, the logic of each agent is described using the language AgentSpeak(L++), which is a modification and extension of AgentSpeak(L).

AgentSpeak(L++) is a logic programming language, which means that all elements can be reduced to terms and literals, representing behaviour and environment data in a symbolic way. In an agent cycle, all plans with true plan conditions are triggered in parallel. Note that for beliefs, plans and actions, hierarchical naming structures can be defined using slashes (/) or minuses (-). [Aschermann et al., 2016, p.7f.]

The following example code is directly taken from [Aschermann et al., 2016, p.19f.], and the explanation before and after the code closely follows [Aschermann et al., 2016, p.17, p.19f.]. In the example, we consider the AgentSpeak(L++) code for a walking agent in an evacuation scenario. The grid environment contains rectangular obstacles the agents need to pass in order to reach the defined exit destination. In the code below, all plans relating to movements are grouped together by naming them "movement/...".

```
// initial-goal
!main.

// initial plan (triggered by the initial-goal)
// calculates the initial route
+!main<-
  route/set/start( 140, 140 );
  !movement/walk/forward.

// walk straight forward into the direction of the goal-position
+!movement/walk/forward<-
  move/forward();
  !movement/walk/forward.

// walk straight forward fails then go left
-!movement/walk/forward<-
  !movement/walk/left.

// walk left - direction 90 degree to the goal position
+!movement/walk/left<-
  move/left();
  !movement/walk/forward.

// walk left fails then go right
-!movement/walk/left<-
  !movement/walk/right.

// walk right - direction 90 degree to the goal position
+!movement/walk/right<-
  move/right();
  !movement/walk/forward.

// walk right fails then sleep and hope everything will be
// fine later, wakeup plan will be triggered after sleeping
-!movement/walk/right<-
  T = math/statistic/randomsimple() * 10 + 1;
  T = generic/type/toint( T );
  T = math/min( 5, T );
  generic/sleep( T ).

// if the agent is not walking because speed is
// low the agent increments the current speed
+!movement/standstill<-
  >>attribute/speed( S );
  S = generic/type/toint( S ) + 1;
  +attribute/speed( S );
  !movement/walk/forward.

+!position/achieve( P, D ) <-
  route/next;
  !movement/walk/forward.

// if the agent woke up the speed is set to 1 and the agent
// starts walking to the next goal-position
+!wakeup<-
  +attribute/speed( 1 );
  !movement/walk/forward.
```

The initial plan (which is triggered by the initial goal) is defined in line 4-8 and is responsible for calculating a route for the agent from its initial position to the goal position. The routing action in line 7 (route/set/start) calculates a list of landmarks as subgoals for the agent. Subsequently, the agent tries to follow up on each landmark in order to reach the goal position. After each landmark, the agent uses the plan structure for calculating the next position. After the completion of the routing, the agent starts a new cycle and begins walking, see line 8 (!movement/-/walk/forward). The backend triggers the plan in line 49-51 if the agent is close to (i.e. within a defined radius of) the next landmark. In this plan, the agent slows down if it is close to the landmark. Finally, a wake-up plan is defined in line 53-57, which is automatically triggered by the end of the sleeping time. Waking up, the agent resets its speed to 1 (+attribute/speed(1)) and begins to walk forward (!movement/walk/forward). [Aschermann et al., 2016, p.20]

2.4.2.2 MAS and microscopic traffic simulation

Agent-based simulation is a useful approach for researching complex situations. MAS can solve problems that are difficult or impossible to solve for individual agents or monolithic systems [Alkhateeb et al., 2010]. In the past few decades, MAS have been increasingly used in traffic research, see for example [Bazzan and Klügl, 2009]. [Teixeira et al., 2019, p.3] listed some frameworks that combined microscopic traffic simulation and MAS:

[Rossetti et al., 2000] combined a microscopic traffic simulator with a BDI MAS to model the uncertainty and variability of human behaviour, such as in commuting scenarios. They used agent-based techniques to model the driver's decision-making processes to enable a more accurate prediction of actual departure times, arrival times and routes.

[Soares et al., 2013] presented the integration of a multi-agent development framework, Java Agent DEvelopment framework (JADE) and the traffic simulator SUMO to build a simulation framework where drivers and traffic control could be designed as MAS. They aimed to improve traffic flow analyses by incorporating the cognitive and behavioural aspects of the drivers.

[Görmer and Müller, 2012] combined JADE and a traffic simulation platform, Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks (AIMSUN), to investigate a decentralised dynamic vehicle-grouping algorithm.

A traffic-management solution for an adaptive green-wave protocol was evaluated using Jason and SUMO in [de Abreu Batista and Coutinho, 2013]. They proposed a multi-agent system for urban traffic control that mimics a social organisation. The MAS in this work includes traffic-light agents and focuses on creating green waves in a dynamic way that improves local control and coordination of successive intersections.

Chapter 3

Research Gap and Research Question

3.1 Research gap

As explained in Chapter 1, the main focus of this dissertation lies on the situation where several agents want to visit a city and are requested to form travel groups in the form of platoons consisting of autonomous vehicles. This scenario leads to two questions:

1. How can the travel groups be determined?

2. How can the travellers agree on destinations to be visited?

We decided to focus on the approach of using voting to tackle these issues. Note that we view routing as a downstream problem. We are interested in how different voting algorithms (voting rules, grouping algorithms and voting protocols) compare regarding user goals (low organisational effort and high preference fulfilment) and system goals (large groups). Next, we compare the contribution of this dissertation with the related work described in Chapter 2.

3.1.1 Collective decision-making

In this subsection, we compare the contribution of this dissertation with the related work on collective decision-making.

- Coalition formation: the problem considered in this dissertation seems similar to coalition formation as described in Subsection 2.1.3.

  - Matching problems [Klaus et al., 2016]: For example, as mentioned in Subsubsection 2.1.3.1, it bears a resemblance to matching problems, but other than in bipartite matching with one-sided preferences, in our scenario a traveller chooses several POIs. Rather than focusing on finding stable matchings like in many-to-many matchings, we aim at comparing the effects of several voting algorithms on user and system goals.

  - Hedonic games [Aziz et al., 2016]: As mentioned in Subsubsection 2.1.3.2, the situation considered in this dissertation could theoretically be modelled as a hedonic game if assuming a fixed voting rule. Other than in hedonic games, we aim at drawing comparisons between several voting algorithms.

- GASP [Darmann and Lang, 2016]: As described in Subsection 2.1.4, simplified GASP is similar to our considered problem, with the difference that we consider a scenario where agents can be assigned to several POIs.

- MARA [Rothe et al., 2012]: as described in Subsection 2.1.5, the assignment of POIs to agents is reminiscent of the allocation of objects to agents, with the difference that in our situation, a POI can be assigned to several agents.

3.1.2 Transport applications

In this subsection, we compare the contribution of this dissertation with related works on transport applications.

- Ridesharing [Furuhata et al., 2013]: as described in Subsection 2.2.1, other than in ridesharing, in our approach, the destinations or activities are determined based on the preferences of the travellers over all possible destinations, and we consider routing to be a downstream problem.

- Platooning [Bergenhem et al., 2012, Haas and Friedrich, 2017]: as described in Subsection 2.2.2, we focus on other target quantities than in other works on platooning, namely preference dissatisfaction, group size and organisational effort.

3.1.3 Collective decision-making in transport

In this subsection, we compare the contribution of this dissertation with related works on collective decision-making in transport.

- Reservation-based approaches for traffic management: As described in Subsection 2.3.1, other than in the reservation-based approaches presented by the researchers [Dresner and Stone, 2008] and [Vasirani and Ossowski, 2009] where intersection managers assign space-time slots to vehicles crossing intersections, we consider a scenario with clearly defined groups of vehicles which need to agree on common destinations.

- Model for urban mobility simulation [Grimaldo et al., 2012]: As described in Subsection 2.3.2, Grimaldo et al. developed a market-based approach for urban mobility simulation. Other than in this approach, we focus on common decisions via voting and compare several voting algorithms.

- Modelling joint-travel behaviour [Dubernet and Axhausen, 2012]: As described in Subsection 2.3.3, Dubernet et al. modelled joint-travel behaviour. Other than in this approach, we focus on comparing the effects of several voting algorithms.

- Joint-travel problem [Liao, 2017]: As described in Subsection 2.3.4, rather than finding an optimal path for joint activities, we aim to achieve consensus on several destinations and compare several voting algorithms in regard to system and user goals. Furthermore, we assume a scenario without time windows and with fixed assembly point.

- Tactical-level decision-making in platoons [Kokkinogenis et al., 2019]: As described in Subsection 2.3.5, other than in the approach of Kokkinogeneis et al., we use voting instead of auction rules to agree on destinations for the platoon. We also consider several grouping algorithms.

3.1.3.1 Existing voting applications in traffic management

- Institutionalised consensus in vehicular networks [Sanderson and Pitt, 2012]: As described in Subsubsection 2.3.6.1, rather than focusing on a robust implementation of consensus, we compare several voting approaches with regard to user and system goals.

- Leader election in platoons (e.g. [Singh et al., 2018]): As described in Subsubsection 2.3.6.2, voting is often used to determine platoon leaders. In our approach, we also consider voting for platooning, but for agreeing on common destinations.

- Autonomous vehicle coordination through voting-based decision-making: As described in Subsubsection 2.3.6.3, other than in the approach by [Teixeira et al., 2018], we assume an ideal situation with perfect communication. Our focus is on comparing the inherent effects of several voting algorithms on the target quantities preference dissatisfaction, group size and organisational effort. This also includes several grouping algorithms. We consider committee voting rules.

- Collective decision making for autonomous vehicle coordination: As described in Subsubsection 2.3.6.4, other than in the approach by [Teixeira et al., 2019], we additionally considered the Minisum-Ranksum voting rule. We focused on the target quantities preference dissatisfaction, group size and organisational effort. We also considered several grouping algorithms.

- Group aggregation strategied for tourism [Najafian et al., 2020]: As described in Subsubsection 2.3.6.5, other than in the approach by Najafian et al., we consider several grouping algorithms and focus on the target quantities preference dissatisfaction, group size and organisational effort.

3.1.4 Overview research gap

As discussed, there has been a wide range of works in the areas Computational Social Choice, ridesharing, platooning and collective decision-making in transport.

Regarding the related works on Computational Social Choice, there is a range of works considering coalition formation, group activity selection and resource allocation problems. Unlike in coalition formation, our aim is to compare several voting algorithms in regard to their effects on system and user goals. Other than in group activity selection, we consider a situation where a traveller can be assigned to several POIs. Unlike in resource allocation, POIs can be assigned to several agents.

Regarding ridesharing, situations in which destinations or actions are fixed a priori are well-researched. In our research application, the destinations or actions are not fixed a priori but have to be agreed upon by the travellers. We focus on other target quantities than on the existing range of works on platooning.

Related works on collective decision-making in traffic focused on other target quantities, did not consider group formation, did not compare the effects of several voting algorithms, used other voting algorithms, did not consider clearly defined groups of vehicles, used voting for other applications, or used collective decision-making algorithms other than voting. It is important to note that we focus on comparing the inherent effects of several grouping algorithms, voting rules and voting protocols on system and user goals.

In the following table, we provide a summary of the comparisons between related works and our approach.

SotA	**Considered approach**	**Reference**
Collective decision-making: Matching under preferences	Other than in bipartite matching with one-sided preferences, in our scenario a traveller chooses several POIs. Rather than focusing on finding stable matchings like in many-to-many matchings, we aim at comparing the effects of several voting algorithms on user and system goals	[Klaus et al., 2016]
Collective decision-making: Hedonic games	We aim to draw comparisons between several voting algorithms	[Aziz et al., 2016]
Collective decision-making: GASP	Agents can be assigned to several POIs	[Darmann and Lang, 2016]
Collective decision-making: MARA and auctions	POIs can be assigned to several travellers	[Rothe et al., 2012]
Transport applications: Ridesharing	Destinations are determined based on the preferences of the travellers over all possible destinations. We consider routing to be a downstream problem	[Furuhata et al., 2013, Wang et al., 2016]
Transport applications: Platooning	We consider other target quantities	[Bergenhem et al., 2012, Haas and Friedrich, 2017]
Collective decision-making in transport: Reservation-based approaches for traffic management	We consider clearly defined groups of vehicles which need to agree on common destinations	[Dresner and Stone, 2008, Vasirani and Ossowski, 2009]
Collective decision-making in transport: Model for urban mobility simulation	We focus on common decisions via voting and compare several voting algorithms	[Grimaldo et al., 2008, Grimaldo et al., 2012, Grimaldo et al., 2010]
Collective decision-making in transport: Modelling joint-travel behaviour	We compare several voting algorithms	[Dubernet and Axhausen, 2012, Dubernet and Axhausen, 2013]

Collective decision-making in transport: Joint-travel problem	We aim to achieve consensus on several destinations for a scenario without time windows and with a fixed assembly point. We compare several voting algorithms in regard to system and user goals	[Liao, 2017]
Collective decision-making in transport: Tactical-level decision-making in platoons	We use voting rules to agree on destinations for a platoon, and we consider several grouping algorithms	[Kokkinogenis et al., 2019]
Existing voting applications in traffic management: Institutionalised consensus	We compare several voting approaches with regard to user and system goals	[Sanderson and Pitt, 2012]
Existing voting applications in traffic management: Leader election in platoons	We use voting for agreeing on common destinations	[Singh et al., 2018]
Existing voting applications in traffic management: Autonomous vehicle coordination through voting-based decision-making	We assume perfect communication and compare the inherent effects of several voting algorithms on preference fulfilment, group size and organisational effort. We consider committee voting rules and several grouping algorithms.	[Teixeira et al., 2018]
Existing voting applications in traffic management: Collective decision-making for autonomous vehicle coordination	We focus on the target quantities preference dissatisfaction, group sizes and organisational effort. We consider Minisum-Ranksum as well as several grouping algorithms.	[Teixeira et al., 2019]
Existing voting applications in traffic management: Group Aggregation Strategies for Tourism	We consider group formation and focus on the target quantities group sizes, preference dissatisfaction and organisational effort. We consider several grouping algorithms.	[Najafian et al., 2020]

3.2 Research questions

We aim to investigate the applicability of voting in traffic management for scenarios with travel groups, focusing on situations where the groups need to agree on common destinations.

In this dissertation, we investigate several voting algorithms for the purpose of creating travel groups and a consensus on destinations for the groups. We compare the inherent effects of several grouping algorithms, voting rules and voting protocols.

We consider the following research question, which we then split into several subquestions for each type of algorithm.

RQ *How do different algorithms used for creating travel groups and for determining common destinations compare regarding system and user goals?*

RQ1 *How do different voting rules used for determining common destinations compare regarding system and user goals?*

RQ2 *How do different voting protocols used for voting in travel groups compare regarding system and user goals?*

RQ3 *How do different grouping algorithms used for creating travel groups compare regarding system and user goals?*

RQ4 *How do different combinations of voting protocols and grouping algorithms used for creating travel groups and for determining common destinations compare regarding system and user goals?*

As mentioned in Chapter 1, we aim at few large groups from the system perspective, which leads us to group size as a metric. Furthermore, from the user perspective, we aim at high preference fulfilment and low organisational effort. The preference fulfilment measures how satisfied the respective traveller is with the destinations chosen for the traveller's group. As mentioned in Chapter 1, organisational effort measures both the cognitive effort for the traveller and the phases in which the traveller has to wait for results from other agents.

Considering a traveller who enters the city and is immediately allowed to drive alone using an autonomous vehicle, preference fulfilment is high and organisational effort is low. However, from the system side, this is unfavourable because the group has a size of exactly one, which makes it harder to optimise traffic flow. From the traffic management perspective, coordinating fewer and larger groups is more favourable.

By contrast, considering a group of travellers that travels together and has to agree on common destinations using voting, the group size is larger than in the single-driver case. However, from the user perspective, we have lower preference fulfilment and higher organisational effort.

We aim to investigate how these system and user metrics compare for different grouping algorithms, voting rules and voting protocols. To this end, we conduct agent-based simulations. We choose to use agent-based simulation because we consider a dynamic problem. Also, we want to be able to model human behaviour, in this case preference-based decision-making. In a similar vein to the approach described by [Carley, 1999], we use the simulation results to generate hypotheses regarding the effect of the different algorithms on the quantities preference dissatisfaction, organisational effort and group size. We use the generated hypotheses both for recommendations for system designers of real-world applications and as basis for further research.

Chapter 4

Model

In this chapter, we depict our approach for solving the problem described in Chapter 1. First, we describe assumptions for our approach, the preferences model and the vote types we use. Subsequently, we describe the considered algorithms: grouping algorithms, voting rules and voting protocols. Following this, we explain the models used for preference fulfilment / preference satisfaction and organisational effort. Lastly, we demonstrate with an example how the algorithms can be used to solve the problem of assigning travellers to groups and to find common destinations for these groups.

4.1 Assumptions

We have the following assumptions for the considered scenario. Let $P = \{p_1, ..., p_m\}$ be the available POIs in the urban area. We consider a time span with travellers $T = \{t_1, ..., t_n\}$, where each traveller has preferences over all POIs in P. In reality, travellers might not have clear preferences over all available POIs. For our investigations, however, we stick to this assumption in order to keep the mathematical model simple. Furthermore, we assume that the travellers arrive on different points in time at a pre-defined assembly point where they are grouped together in groups with maximum size / capacity c. Each group can visit exactly k POIs. We make this last assumption in order to keep the mathematical model simple and to make the results of different simulation runs better comparable.

4.2 Traveller preferences

In the following, we describe our preference model. Each traveller t_i has a preference vector $pref_i$ with values from [0,1], where $pref_{ij}$ is the preference of traveller t_i for POI p_j. We consider different possibilities of preference generation and conversion. Note that we use randomly generated preferences because it would go beyond the scope of this dissertation to survey real-world preferences.

The simplest way to generate preferences is to create the preference values according to the uniform distribution on interval [0,1]. We use these *uniform preferences* as a baseline. As an extension, we use a second possibility to generate more realistic preferences for comparison. In order to create realistic preferences, we need an evaluation system. We decided to use the platform *Foursquare* for this. Other researchers used Yelp for this purpose, see [Wang, 2018]. Foursquare yields properties of POIs such as overall rating and categories they belong to. To derive realistic preferences, the preference generation is conducted in a two-stage process as follows:

1. Each agent specifies their preference for the categories ($k_j \in [0, 1]$ for each category j).[1]

2. For each POI p_k, its weight $weight(p_k) \in [0, 1]$ is the tenth of its Foursquare rating value, which leads to [0,1]-normalised weights since the maximum value for Foursquare rating values is 10.

Based on this, the preference of traveller t_i for POI p_k of category j is computed as

$$pref(t_i, p_k) = pref_{ik} = f(k_j) * weight(p_k), \text{ where } f(x) = \frac{x+1}{2} \in [0.5, 1]$$

The preferences over all categories sum up to 1.

4.3 Vote types

In this dissertation, we consider two different types of votes which are widely used in ComSoc research, namely *Approval votes* and *complete linear orders*, as described in Section 2.1.1. The private preference vector $pref_i$ of traveller i is converted into an Approval vector av_i as follows: If $pref_{ij} \leq 0.5$, av_{ij} is 0, otherwise av_{ij} is 1. As for complete linear orders, $pref_i$ is converted into a linear order lv_i as follows: $pref_{ij} > pref_{ik}$ implies $p_j \succ p_k$. In the case of ties, lexicographic tiebreaking is applied.

4.4 Voting rules

In the following, we describe the voting rules we investigate in this dissertation. Note that we slightly misuse the term voting rule here, as we use it from now on for committee voting rules; in this dissertation, we focus on committee voting rules and disregard single-winner voting rules, except for the excursus in Chapter 9 and for one aspect of the extended model described in Chapter 8.

We decided to compare the effects of two well known voting rules based on Approval votes, namely Minisum-Approval and Minimax-Approval. As explained in Subsection 2.1.2, Minisum-Approval corresponds to a utilitarian approach, whereas Minimax-Approval corresponds to an egalitarian approach. Additionally to comparing the effect of utilitarian

[1] For the main simulation, we assume different agent types, where each agent type has a basis preference regarding the categories. We use normal distribution to create preferences diverging from the basis preferences, as described in Chapter 5.

and egalitarian approaches, we were also interested in comparing the effects of approval-based vs. linear-order-based voting rules. Hence, we decided to also investigate Minisum-Ranksum as introduced in Subsection 2.1.2, which is based on complete linear orders. For determining unique winner committees, lexicographic tie-breaking is used if needed.

Consider as small example for Minisum-Approval an election with three travellers t_1, t_2 and t_3 which hold an election with the votes v_1, v_2, v_3 to choose two out of four POIs p_1, p_2, p_3 and p_4. The POIs p_2 and p_3 are each approved by two travellers and constitute the winning committee as the other POIs are each approved by only one traveller.

Vote / POI	p_1	p_2	p_3	p_4
v_1	0	1	0	1
v_2	0	1	1	0
v_3	1	0	1	0
Sum	1	2	2	1

4.5 Group formation

We consider two different possibilities for group formation. The simplest way to group the travellers together is to assign them to groups according to the order in which they arrive at the assembly point. We call this possibility *sequential grouping*. As second possibility, we consider *coordinated grouping*: Each newly arriving traveller chooses the group with the smallest disagreement between their vote and the current election result of the group, unless the disagreement with all election results in all groups exceeds a certain join threshold; in this case, the traveller creates a new group. Following [Baumeister and Dennisen, 2015], for approval-based voting rules (i.e. Minisum- and Minimax-Approval), we use the Hamming distance to measure the disagreement and for voting rules based on complete linear orders (i.e. Minisum-Ranksum), we use the normalised ranksum as described in Chapter 2. The traveller groups accept new agents up to their defined capacity or until a deadline is reached.

As a small example for coordinated grouping under Minisum-Approval consider three travellers t_1, t_2 and t_3 with the following votes: v_1, v_2 and v_3.

v_1	0	1	0	1
v_2	0	1	0	0
v_3	1	0	1	0

Assuming that the travellers arrive in the order their indices represent and we have a join threshold of 2 with capacity 3, we get the following groups.

Traveller t_1 arrives first and is assigned to a new group G_1 which contains only this traveller.

v_1	0	1	0	1
Result	0	1	0	1

Next, traveller t_2 arrives. The Hamming distance between v_2 and the current election result of Group G_1 is $HD((0,1,0,0),(0,1,0,1)) = 1 < 2$, which is the minimal disagreement of t_2 with any group, so t_2 is assigned to group G_1.

v_1	0	1	0	1
v_2	0	1	0	0
Result	0	1	0	1

Traveller t_3 arrives last. The Hamming distance between v_3 and the current election result of Group G_1 is $HD((1,0,1,0),(0,1,0,1)) = 4 > 2$, and there are no other groups, so t_3 is assigned to a new group G_2 which contains only this traveller.

v_3	1	0	1	0
Result	1	0	1	0

4.6 Voting protocols

We consider two different voting protocols which describe how voters interact with their chair once they have joined the group. The first protocol we consider is the following one: In the *basic protocol*, the group departs as soon as the election result for the filled group (i.e. either the maximal capacity or a deadline has been reached) is known. The second protocol, the *iterative protocol*, aims at reorganising the travellers based on the election result of the filled group in order to create groups where the travellers' preference satisfaction with the election result is higher. The concept for this protocol was also investigated by [Dennisen and Müller, 2015] and [Dennisen and Müller, 2016].

Figure 4.1 and Figure 4.2 give an overview of the communication models for the both protocols. These figures have been adapted from [Dennisen and Müller, 2015, p.6] and [Dennisen and Müller, 2016, p.3].

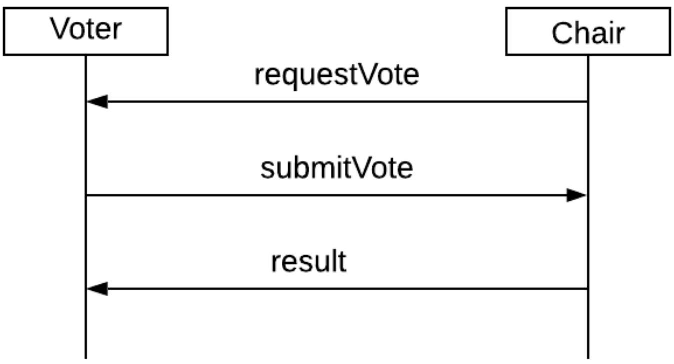

Figure 4.1: Communication model for basic protocol

Note that each group has a chair who is responsible for collecting the votes, computing the result of the election and for sending the (current) result to the voters. Once the final

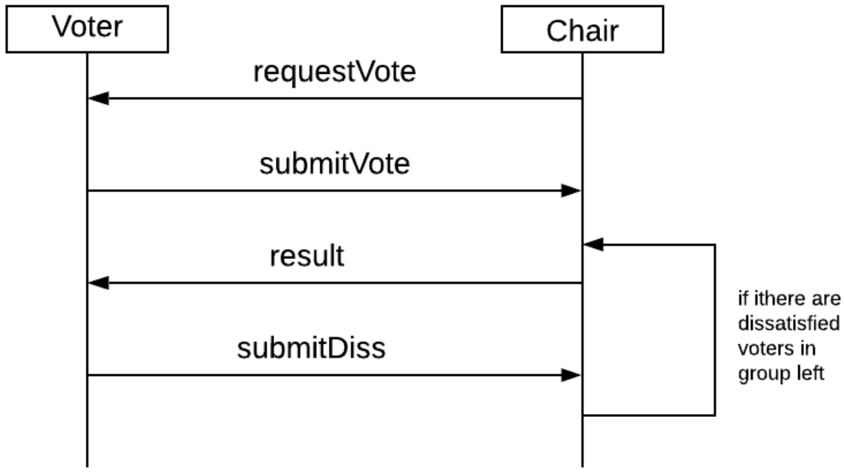

Figure 4.2: Communication model for iterative protocol

result for a group has been sent to all voters (and is accepted), the group can depart.

Consider for both protocols the election result for the filled group. While in the basic voting protocol the announcement of this election result means that the group can leave, there are additional steps in the iterative voting protocol. If there are travellers who are dissatisfied with the result of the election of the filled group, the most dissatisfied one leaves the group and the winner determination is immediately repeated. Which travellers are dissatisfied can be determined via a dissatisfaction threshold. This process is repeated until there are only satisfied travellers left, and then the group departs, i.e. the winner determination, the announcement of the election result, the computation of the dissatisfaction with the result and the removal of voters is iterative[2]. Travellers who leave their group are considered similarly to newly arrived travellers; they either join one of the existing groups they have not belonged to before or are assigned to a new group.

Note that, under the basic protocol, in the case of coordinated grouping, each time a new traveller joins the group, the chair requests the new voter to submit their vote and recomputes the winner determination. Once the group is filled, the chair sends the final election result to the voters.

4.7 Preference fulfilment/satisfaction model

Our preference fulfilment/satisfaction model is defined as follows: For the preference satisfaction, we sum up the real-valued preferences for the selected candidates, i.e. the preference satisfaction of traveller t_i with commitee K would be

[2]Note that the well-known term *iterative voting* in the literature refers to something else, as described in [Airiau et al., 2017, p.3]: The voters start from a voting situation; a winner is announced based on the original votes. Then, the voters change their votes, one at time, resulting in several iterations with the respective preference profiles and winners.

$sat(i,K) = \sum_{POI_j \in K} pref_{ij}$.

Analogously, the preference dissatisfaction of traveller t_i with commitee K is defined as

$$diss(i,K) = \sum_{POI_j \in K} (1 - pref_{ij})$$

We define a normalised preference dissatisfaction as follows: Let $sat(i,K)$ be defined as above. Additionally, let $maxsat(i,C,k)$ be the maximum possible satisfaction of traveller t_i. It is defined as

$$maxsat(i,C,k) = \sum_{POI_j \in C_{(i,k)}} pref_{ij}$$

where $C_{(i,k)}$ is the set of k POIs with highest preference value of traveller t_i.

The normalised preference satisfaction of traveller t_i for K is

$$nsat(i,K) = sat(i,K)/maxsat(i,C,k)$$

and analogously, the normalised preference dissatisfaction of traveller t_i for K is

$$ndiss(i,K) = 1 - sat(i,K)/maxsat(i,C,k)$$

In our main simulation series in Chapter 6, we use normalised dissatisfaction.

4.8 Organisational effort

As described in Subsection 2.4.1, we use LightJason agents for the agent-based evaluation simulation. In our approach, we use the number of agent cycles (as defined in [LightJason, 2019b]) which elapse for each traveller agent to measure their organisational effort. This includes both the phases where they actively take part (for example by submitting votes) and the phases where they have to wait for results from other agents.

4.9 Overall voting process

To summarise the overall voting process, we determine the election result for each election group with a given voting rule. The travellers are either sequentially assigned to groups or via coordinated grouping. When using the basic protocol, the travellers depart either when their group has reached its capacity or when a deadline is reached. Under the iterative protocol, dissatisfied travellers can be reassigned to other groups. The overall process is schematically presented in Figure 4.3.

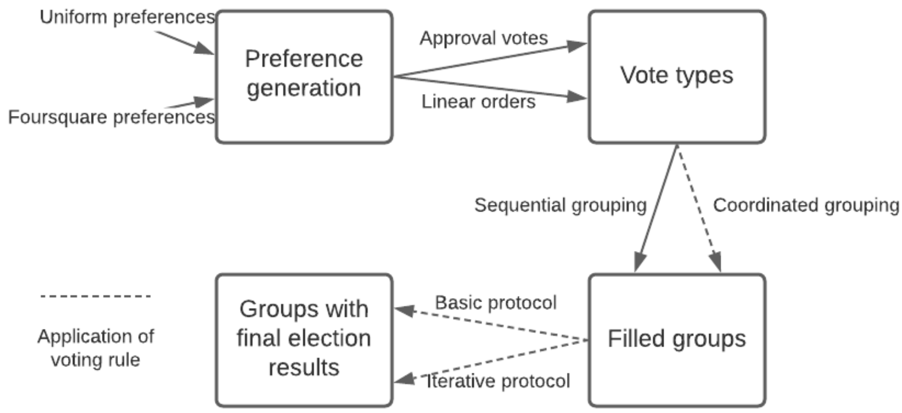

Figure 4.3: Overall voting process

4.10 Example

In the following, an example for the steps of the overall voting process is described. We assume three travellers t_1, t_2, t_3 who want to visit Manhattan with the available POIs $P = \{p_B, p_C, p_H, p_M, p_N, p_P\}$. For group formation, we assume sequential grouping, travellers t_1, t_2, t_3 being assigned to group G_1. Furthermore, we assume that the travellers decide on common destinations with Minisum-Approval using the basic protocol and visit exactly $k = 3$ POIs. In this example, we use Foursquare data to generate the preferences, with the Foursquare ratings/categories being

- p_B: Bryant Park/9.7/zoo park
- p_C: Central Park/9.8/zoo park
- p_H: Carnegie Hall/9.5/historic site
- p_M: MoMA/9.5/museum
- p_N: American Museum of Natural History/9.2/museum
- p_P: St. Patrick's Cathedral/9.6/historic site

Next, for each traveller, we give the preferences for each of the six POIs and compute the respective preference values given the function $pref_{ik}$ introduced in Section 4.2

The Approval vote of t_1 is constructed as follows, see Figure 4.4: In the first stage, we specify that their preference for the categories historic site and zoo park is 0.5, respectively. In the second stage, we compute the weights for each POI from the Foursquare ratings. For example, the weight for St. Patrick's Cathedral (p_P) is 0.96 since its Foursquare rating is 9.6. The final preference values are greater than 0.5 for the POIs p_B, p_C, p_H and p_P and lower than 0.5 for the other POIs, which leads to the Approval vector $(1, 1, 1, 0, 0, 1)$.

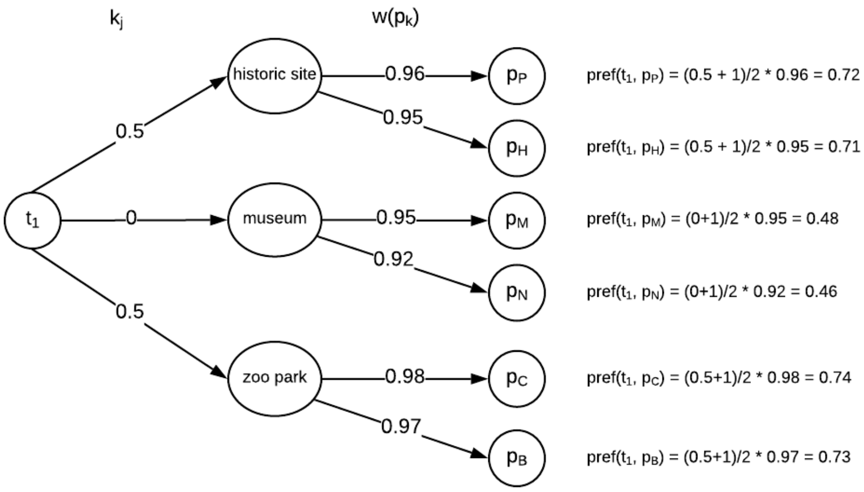

Figure 4.4: Preference generation for t_1

Analogously, the Approval vote of t_2 is $(1, 1, 1, 1, 1, 1)$, see Figure 4.5

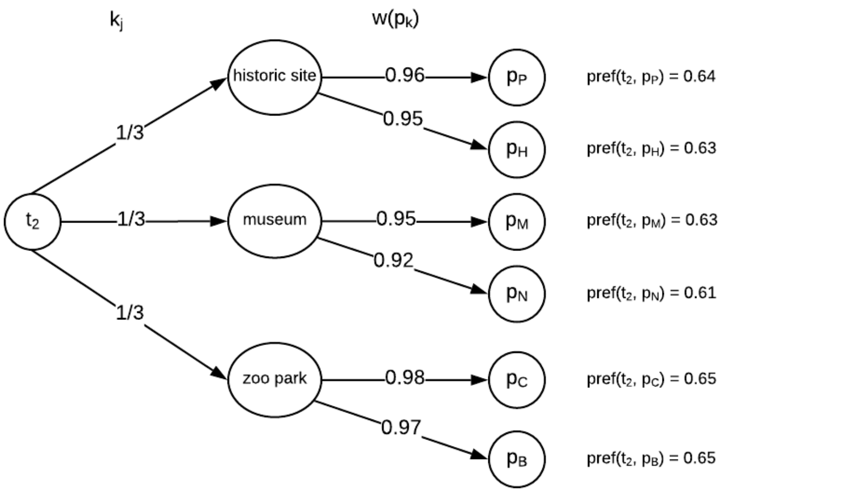

Figure 4.5: Preference generation for t_2

Analogously, the Approval vote of t_3 is $(0, 0, 1, 1, 1, 1)$, see Figure 4.6

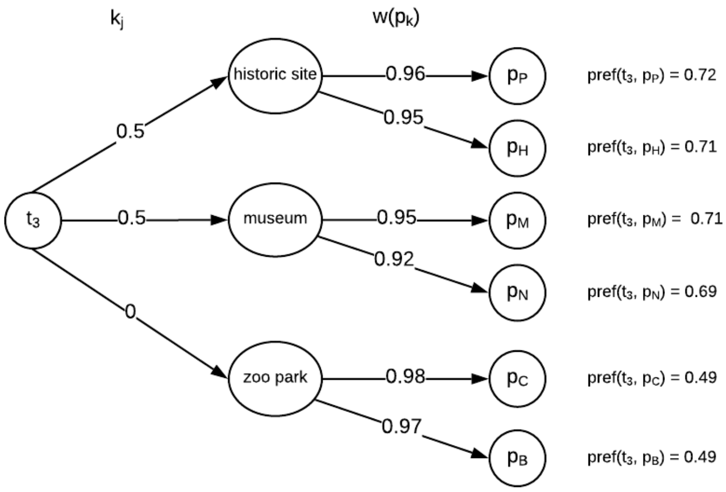

Figure 4.6: Preference generation for t_3

In the final step, we compute the result of Minisum-Approval election based on the three traveller votes, see Table 4.1. The members of the winning committee are p_H and p_P with Approval score 3 and p_B with Approval score 2 via tie-breaking. This means that the group will visit the POIs Bryant Park, Carnegie Hall and St. Patrick's Cathedral.

POI	p_B	p_C	p_H	p_M	p_N	p_P
t_1	1	1	1	0	0	1
t_2	1	1	1	1	1	1
t_3	0	0	1	1	1	1
Result	**2**	2	**3**	2	2	**3**

Table 4.1: Minisum-Approval election for real-world example

Chapter 5

Implementation

In the following, we give an overview of the implementation details for the simulation tool LightVoting which was newly developed to evaluate the algorithms described in Chapter 4. First, we describe the simulation architecture. Subsequently, we give information on the LightJason aspect, the execution model, the preferences generation and the statistics component of the simulation.

5.1 Simulation architecture for LightVoting

For the simulations, the agent-based simulation tool LightVoting was developed, see [Dennisen, 2021b]. We decided to develop an own tool because existing tools at this time did not provide a scalable, domain-independent way to simulate elections. The architecture is based on the framework LightJason and has the following components: runtime, input, voting and statistics. Note that in the implementation for our main simulation, we decided, to keep the implementation simple, to use a broker agent which is responsible for creating voter and chair agents and assigning the voter agents to groups. The simulation architecture is schematically depicted in Figure 5.1.

We briefly describe the input parameters, which are explained in more detail in Section 5.3.

- Scenario definition: Number of POIs, number of travellers
- Voting algorithms: Grouping algorithm, voting rule and voting protocol
- Other settings, for example dissatisfaction threshold if applicable
- Preferences of travellers

The input parameters are passed to the voting component which includes voters/travellers, chairs and broker, which are differently defined according to the respective voting protocol and grouping algorithm. The agent part of the voting component is built based on the LightJason framework and has both AgentSpeak(L++) and Java parts: The agents are defined via AgentSpeak(L++) agent definitions and corresponding Java classes. The voting rules are defined via Java classes. The output parameters - preference dissatisfaction and organisational effort values from voters/travellers and group size values from

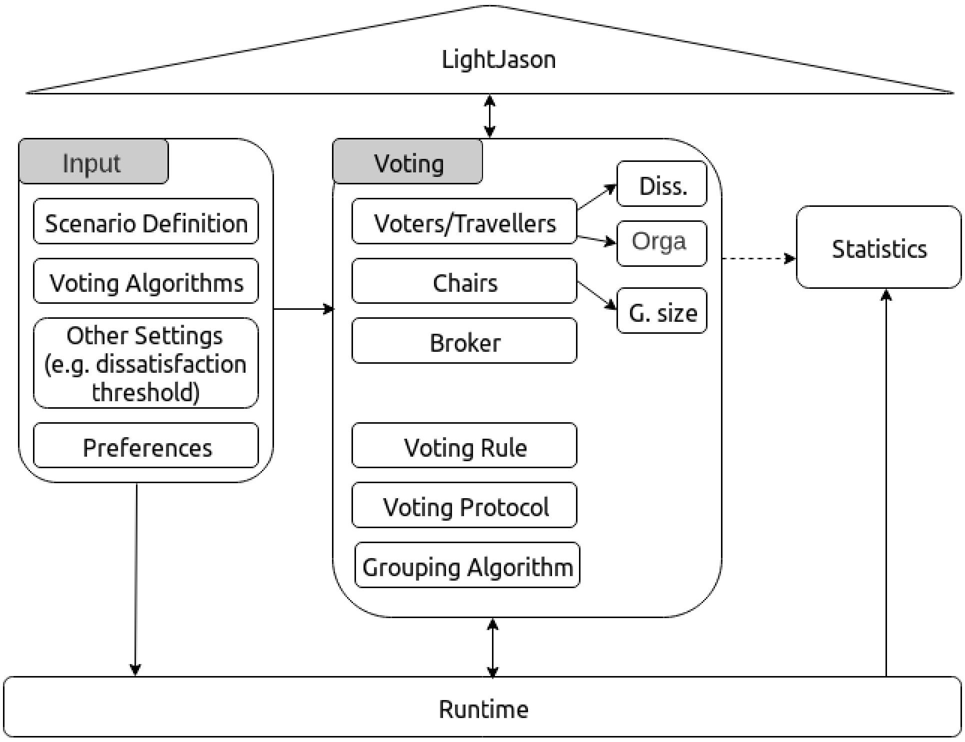

Figure 5.1: LightVoting Simulation Architecture

chairs - are evaluated in the statistics component. The simulation is controlled by a Java runtime class. The Java runtime class reads the input parameters from the input component, controls the agents and the creation of the statistics component.

5.2 LightJason aspect

LightVoting was implemented based on LightJason. LightJason is described in detail in the technical report by [Aschermann et al., 2016]. As explained on the project website, LightJason is a concurrent BDI multi-agent framework inspired by Jason, fine-tuned for concurrent plan execution suitable for distributed computing environments and aiming at efficiency and scalability. [LightJason, 2019a]

In the following, we briefly explain the definition of LightJason agents as finite state machines and describe selected Deterministic Finite Automaton (DFA) models for agents in our simulation.

5.2.1 Finite state machines and logic programming

As described on the LightJason project website [LightJason, 2019c], LightJason agents are defined as *"[f]inite-[s]tate [m]achine[s] in a [l]ogical [p]rogramming language with the following definition:*

- *the initial state is optionally defined with the initial goal*
- *a state is a set of beliefs if a cycle is not running*
- *a transition is the execution of a plan (with instantiation of a goal) and is limited by the plan condition"*

5.2.2 DFA models for selected agents in simulation

In this subsection, we explain some Deterministic Finite Automaton models for agents in our simulation. Each DFA model has the following parts:

- Set Q of states with some end state, marked with double outline
- Transition function: Every transition d corresponds to a plan which leads to the state change

In the following, we briefly recapitulate the agent types used in the simulation and subsequently present selected, simplified, DFA models for agents under the combination sequential_basic.

Agent types In our simulation, we have three different agent types: The broker agent creates the voter and chair agents and assigns the voters to groups. Each chair agent is responsible for the elections in a group, which means that the chair agent collects the votes, computes the election result(s) and - in the case of the iterative voting protocol - removes dissatisfied voter agents from the group. The voter agents submit their preferences in the form of votes to the chair agent so that it can aggregate the votes to a consensus. In the following, we focus on how voter and chair agents behave under the combination sequential_basic.

DFA voting agent model for sequential_basic Figure 5.2 depicts the behaviour of a voter agent under the combination sequential_basic. The voter agent starts with the !start plan d_0 to reach the state q_0. From this state, it can reach the group/joined state q_1 with the corresponding plan d_1, i.e. in state q_1 it has joined a group. To submit a vote to the chair and attain the state q_2, it conducts the plan d_2. Subsequently, it attains the diss/submitted state q_3 by conducting plan d_2, i.e. submitting its preference dissatisfaction. Finally, it reaches the done state q_4 by the plan d_4.

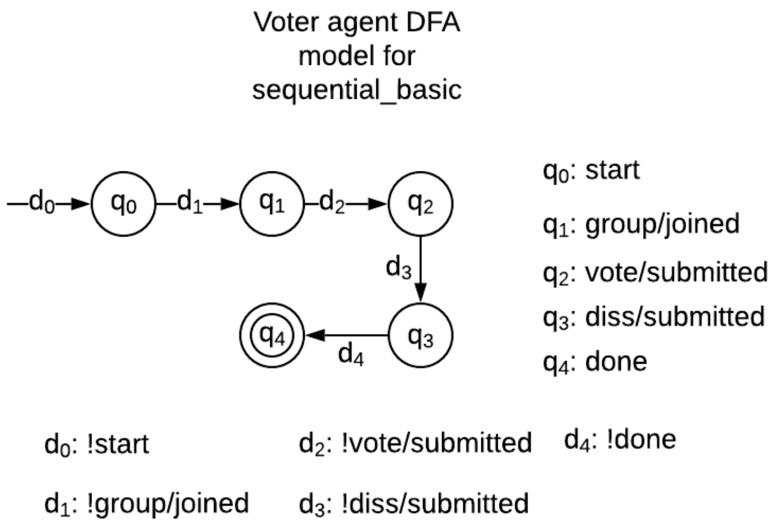

Figure 5.2: DFA model for voter, sequential_basic

DFA chair agent model for sequential_basic Figure 5.3 depicts the behaviour of a chair agent under the combination sequential_basic. The chair agent starts with the !start plan d_0 to reach the state q_0. From this state, it can reach the state stored/vote q_1 via collecting votes. From this state, it can reach the started/voting state q_2 when all agents in the group have submitted their vote. From this state, it can reach the state stored/diss via collecting dissatisfaction values. Finally, it reaches the done state q_4.

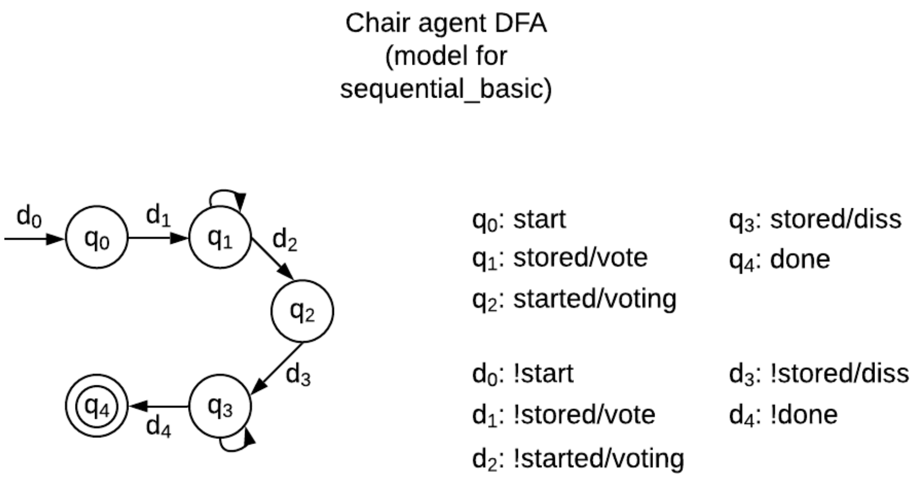

Figure 5.3: DFA model for chair, sequential_basic

5.3 Execution model for LightVoting

The execution model of LightVoting is defined as depicted in Figure 5.4.

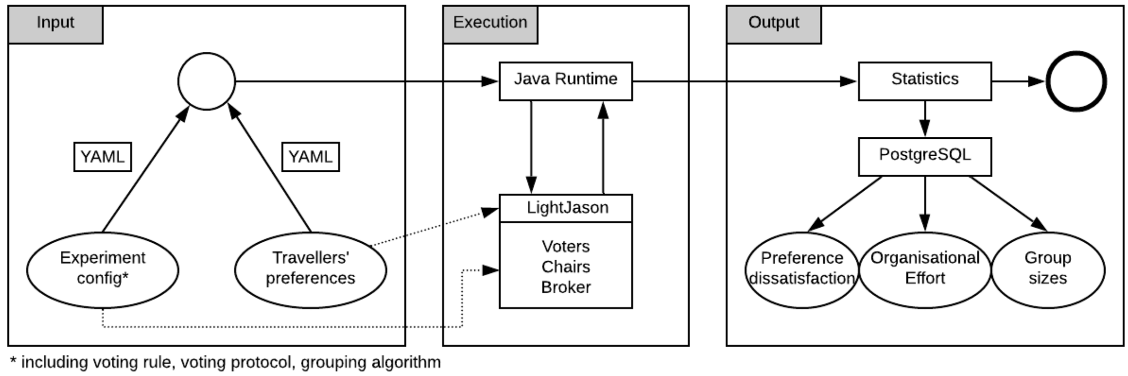

Figure 5.4: Execution model for LightVoting

The input for LightVoting consists of two YAML [1]. files, one for the experiment configuration and the other for the travellers' preferences. The experiment configuration contains the variables as described in Table 5.1.

runs	number of conducted runs
setting	used combination of grouping algorithm and voting protocol, for example COORDINATED_BASIC
rule	used voting rule, for example MINISUM_APPROVAL
agnum	number of voters/travellers
altnum	number of candidates/POIs
comsize	size of the committee to be elected
capacity	maximum size of traveller groups
dissthr	dissatisfaction threshold for the iterative protocol
jointhr	join threshold for coordinated grouping
preferences	specifies whether the preferences are uniform or Foursquare-based

Table 5.1: Configuration variables

On the execution level, the input data are processed by the Java runtime and the LightJason agents, where we have several voter and chair agents and one broker agent for each run. The statistical data are stored in a PostgreSQL database. Here, we store the following data: Preference dissatisfaction, organisational effort for each traveller and the group size.

[1]The name is a recursive acronym standing for YAML Ain't Markup Language. YAML "*is a human friendly data serialization standard for all programming languages*" (`https://yaml.org/`, retrieved 06 April 2021).

5.4 Generation of preferences

As described in Chapter 4, we consider two types of preferences. For the first one, the uniform preferences, we simply draw preference values from the uniform distribution on the interval [0,1].

For the generation of Foursquare-based preferences, we use a Python script which generates a YAML preference file based on a YAML configuration file. This configuration file specifies the following input parameters:

- Number of POIs
- Foursquare categories
  - In each category, specify the names and the popularity of the POIs
- Distribution for generation of preferences, based on agent types with different preferences for the categories
- Number of simulation runs

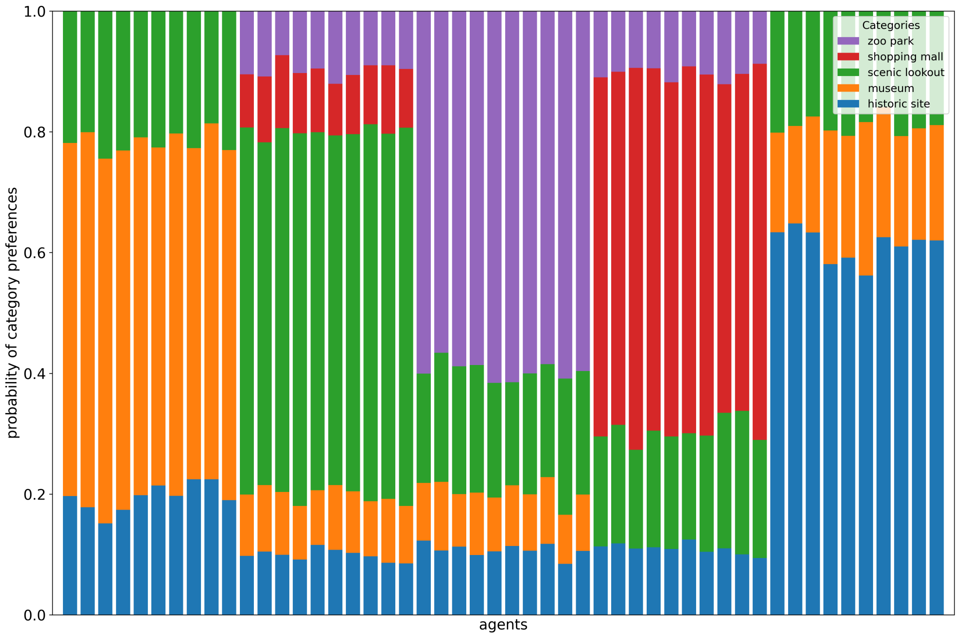

Figure 5.5: Preferences example

An example for the generation of the travellers' preferences over the categories can be seen in Figure 5.5. The bars represent for each agent the probability for choosing the respective category. These probabilities are used to draw preferences over the categories from the [0,1] interval and are created as follows.

We have five agent types with different basis preferences. For each agent type, normal distributed values are drawn for each agent and multiplied with the basis preferences for the categories in this type. After drawing the preference values, the preferences are normalised such that the preferences for an agent over the categories sum up to 1.

If we consider the outermost ten bars on the left side, they represent agents for which the preference for historic sites and scenic lookouts range around 0.2 and the preference for museums ranges around 0.6. The other bars represent 40 other agents which are subdivided in four agent types with other basis preferences.

5.5 Statistics

In the following, we explain the statistics component of the simulation. We needed statistics for evaluating the results from the main simulation series (Chapter 6). We decided to store the simulation data in a PostgreSQL database and to analyse the simulation data using R, a popular language for statistical modelling and analysis. In the following subsections, we describe the use case for the statistics component as well as its PostgreSQL and R elements.

5.5.1 Use case

We have the following use case for the statistics component, **UC1**, which is depicted in Figure 5.6.

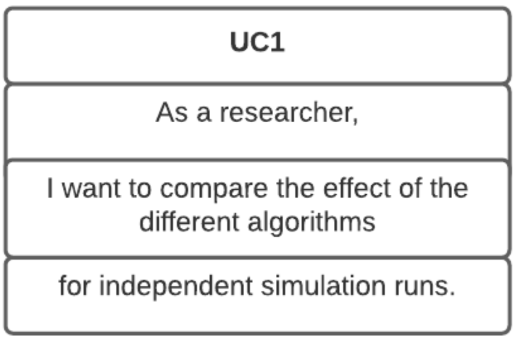

Figure 5.6: Use Case 1

UC1 relates to the main simulation series. The aim here is to compare the different algorithms described in Chapter 4 regarding their effects on the quantities preference dissatisfaction, organisational effort and group size. We need to be able to compare results from independent simulation runs.

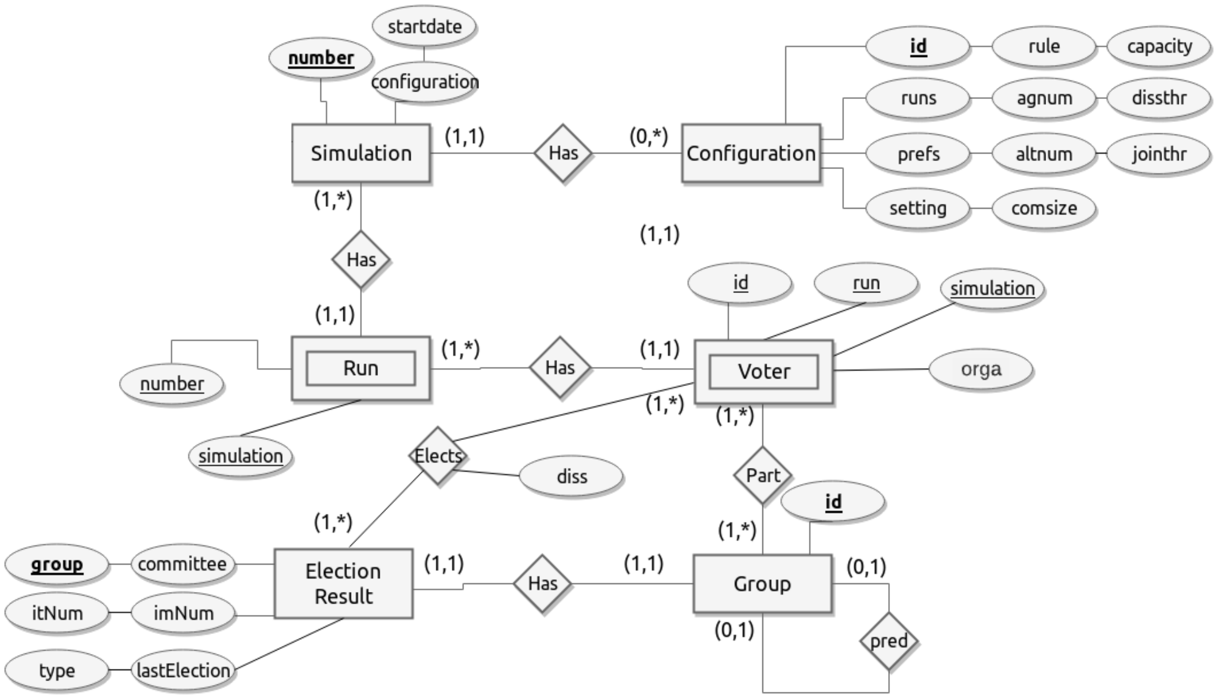

Figure 5.7: Database model

5.5.2 PostgreSQL

We decided to store the relevant data from the simulation in a PostgreSQL database since it is straightforward to use PostgreSQL in Java and also easily possible to connect PostgreSQL to R. The model of this database is depicted in Figure 5.7. Using a PostgreSQL database enables the analysis of the results from independent simulation runs.

Table 5.2 gives an overview over the different entities in the database.

Entity	Notes
Simulation Table	Has the attribute number as primary key, has exactly one configuration and can have any number of runs
Configuration Table	Contains the simulation parameters as attributes, has an id attribute as primary key.
Run Table	Belongs to exactly one simulation, has a composite primary key: number + simulation. Has at least one voter
Voter Table	Has a composite primary key: id + run + simulation. A voter elects at least one election result and is part of at least one group
Group Table	Each group has a unique id and either one or no predecessor group
Election Result Table	Each election result belongs to exactly one group, and the primary key is defined by the according group

Table 5.2: Database entities

5.5.3 Analysis using R

We decided to analyse the simulation data using R, a popular language for statistical modelling and analysis, because it provided the required statistical functions and integrated well into the simulation toolchain. In the following, we describe an example R evaluation file (The code is adapted from [Zimmermann, 2015]).

First, we create a connection to the database containing the simulation results.

```
# install package
require("RPostgreSQL")

# save the password
pw <- {
"..."
}

# load the PostgreSQL driver
drv <- dbDriver("PostgreSQL")
# create connection to the database
# "con" will be used later in each connection to the database
con <- dbConnect(drv, dbname = "lightvoting", host = "...", port = ...,
user = "...", password = pw)
```

The following R code determines minimum, 1st quartal, median, 3rd quartal and maximum values for the output variable preference dissatisfaction.

```
# determine preference dissatisfaction values for last elections in last
# simulation

df_diss <- dbGetQuery(con,
"SELECT elects.voter AS voterid, elects.run AS voterrun,
elects.simulation as votersim,
elects.diss as diss
FROM election_result, elects
WHERE elects.simulation =
(SELECT max(elects.simulation) from elects)
AND
election_result.group_column = elects.electionresult
AND election_result.lastelection = TRUE
GROUP BY elects.voter, elects.simulation, elects.run,
elects.diss")

dissVals <- df_diss['diss']

summary(dissVals)
```

Analogously, the following R code determines minimum, 1st quartal, median, 3rd quartal and maximum values for the output variable group size.

```
# determine group size values for last simulation

df_size <- dbGetQuery(con,
"SELECT simulation.number, election_result.group_column,
count(distinct (voter_group.voter, voter_group.run,
voter_group.simulation)) AS size
```

```
FROM voter_group, simulation, election_result
WHERE
simulation.number=(SELECT max(simulation.number)
FROM simulation)
AND
election_result.group_column=voter_group.group_column
AND election_result.lastelection=TRUE AND simulation.number=
voter_group.simulation
GROUP BY simulation.number, election_result.group_column;")

sizeVals <- df_size['size']

summary(sizeVals)
```

Analogously, the following R code determines minimum, 1st quartal, median, 3rd quartal and maximum values for the output variable organisational effort.

```
# determine organisational effort values for last simulation

df_orga <- dbGetQuery(con,
"SELECT simulation.number, voter.id AS voterid, voter.run as voterrun,
voter.simulation as votersim, voter.orga as voterorga
FROM simulation, voter
WHERE simulation.number=(SELECT max(simulation.number)
from simulation)
AND voter.simulation=simulation.number;")

orgaVals <- df_orga['voterorga']

summary(orgVals)
```

Finally, we close the connection.

```
# close the connection
dbDisconnect(con)
dbUnloadDriver(drv)
```

Chapter 6

Experiments

In this chapter, we aim at answering the following questions:

Q1 *How do the voting rules compare regarding preference dissatisfaction, organisational effort and group size?*

Q2 *How do the voting protocols compare regarding preference dissatisfaction, organisational effort and group size?*

Q3 *How do the grouping algorithms compare regarding preference dissatisfaction, organisational effort and group size?*

Q4 *How do the combinations of grouping algorithms and voting protocols compare regarding preference dissatisfaction, organisational effort and group size?*

In order to answer these questions, we conducted several experiments, the results for which can be found in the appendix. For both uniform and Foursquare-based preferences, we conducted simulations for each voting rule, varying the voting protocol, the grouping algorithm, the dissatisfaction threshold and the join threshold. As explained in Chapter 3, we follow an approach similar to the one described in [Carley, 1999], i.e. we use the simulation results to generate hypotheses. Before we present and discuss the simulation results, we describe the simulation settings.

6.1 Simulation settings for main simulation series

In this section, we describe the simulation settings, which encompass the study data, the considered parameters, the output variables and some technical information.

Study Data We considered the following POIs in Manhattan, NY and vicinity. The POIs and their properties are defined via a YAML file. The categories the POIs belong to and their popularity values were determined manually based on Foursquare data, which are listed in Table 6.1

Category	POI	Popularity
historic site		
	St. Patrick's Cathedral	9.6
	Carnegie Hall	9.5
	Federal Hall National Memorial	8.8
	Former Location of Edward Hopper's Studio (1913 - 1967)	8.6
museum		
	Metropolitan Museum of Art	9.7
	MoMA	9.5
	Whitney Museum of American Art	9.4
	American Museum of Natural History	9.2
zoo park		
	Central Park	9.8
	Bryant Park	9.7
	Madison Square Park	9.6
	Bronx Zoo	9.2
shopping mall		
	Brookfield Place	9.2
	City Place	8.8
	Aritzia	8.6
	Chelsea Market	9.5
scenic lookout		
	Top of the Rock Observation Deck	9.6
	High Line 10th Ave Amphitheatre	9.5
	One World Observatory	9.2
	Under The Brooklyn Bridge	9.1

Table 6.1: Foursquare data for considered sites

Figure 6.1 depicts the POIs located in Manhattan. (The POIs City Place and Bronx Zoo in the vicinity are omitted.) Based on these information, the preferences of the agents are generated as described in Section 5.4.

Considered parameters In Table 6.2, we depict the considered parameters. Note that the term "setting" in this table refers to the different combinations of grouping algorithm and voting protocol. For example, "sequential_basic" refers to the combination of sequential grouping and basic voting protocol. Note that for the considered simulations, the maximum number of cycles agents spend looking for suitable groups is capped to 500 to ensure the termination of the simulation.

Figure 6.1: Considered sites

Map data © OpenStreetMap contributors (`https://www.openstreetmap.org/copyright`)

Name	Explanation
runs	The number of simulation runs
setting	The name of the used combination, e.g. sequential_iterative
rule	The name of the used voting rule, e.g. Minisum-Approval
agnum	The number of traveller/voter agents
altnum	The number of POIs/alternatives
comsize	The size of the to-be-elected POI committee
capacity	The capacity, i.e. maximum size of the to-be-created traveller groups
dissthr	The (preference) dissatisfaction threshold of the travellers, if applicable
jointhr	The join threshold of the travellers, if applicable
preferences	The preference type, i.e. uniform or Foursquare-based preferences

Table 6.2: Description of input parameters

We ran simulations with the parameter settings as described in Table 6.3.

Series	Settings
1	Minisum-Approval, capacity 20, uniform preferences, agnum 41, altnum 20, comsize 5
2	Minimax-Approval, capacity 20, uniform preferences, agnum 41, altnum 20, comsize 5
3	Minisum-Ranksum, capacity 20, uniform preferences, agnum 41, altnum 20, comsize 5
4	Minisum-Approval, capacity 20, Foursquare-based preferences, agnum 41, altnum 20, comsize 5
5	Minimax-Approval, capacity 20, Foursquare-based preferences, agnum 41, altnum 20, comsize 5
6	Minisum-Ranksum, capacity 20, Foursquare-based preferences, agnum 41, altnum 20, comsize 5

Table 6.3: Considered parameter settings

Output variables As output variables for our main simulation series, we considered the three quantities preference dissatisfaction, organisational effort and group size.

Technical information The simulations were conducted using LightVoting as described in Chapter 5, on a Dell Latitude E5440 (Ubuntu Linux).

6.2 Simulation results for main simulation series

In this section, we present the results for the main simulation series. For the different voting rules, voting protocols, grouping algorithms and combinations of grouping algorithms and voting protocols, we compare their effects on preferences dissatisfaction, organisational effort and group size. Note that in the terms of [Carley, 1999], we consider the following input parameters to be key input parameters for generating hypotheses: Voting rule, voting protocol and grouping algorithm.

6.2.1 Voting rules

In this section, we compare the three voting rules regarding their effect on preference dissatisfaction, organisational effort and group size.

6.2.1.1 Preference dissatisfaction

In this section, we consider the question how the voting rules compare regarding preference dissatisfaction. We consider this separately for both preference forms.

Uniform preferences We first consider the preference dissatisfaction values under the combination sequential_basic, see Figure 6.2. As the Wilcoxon tests in the appendix show, the differences between the three rules are significant. We have the relation $diss(MS-RS) < diss(MS-AV) < diss(MM-AV)$: The preference dissatisfaction values for Minisum-Ranksum are lower than for Minisum-Approval, and they are lower for Minisum-Approval than for Minimax-Approval.

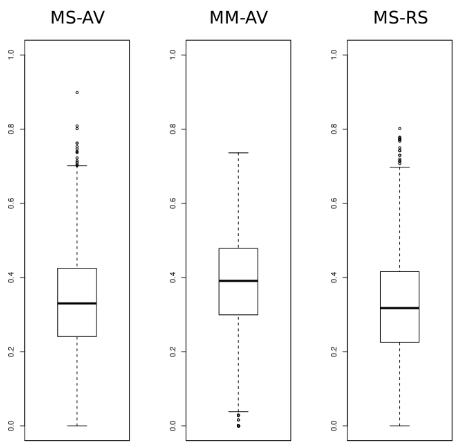

Figure 6.2: Dissatisfaction values for sequential_basic, uniform preferences

We also compared the preference dissatisfaction values under the combination sequential_iterative for median group size 7, see Figure 6.3. As the Wilcoxon tests in the appendix show, the difference between the three rules are significant. We have the relation $diss(MS-RS) < diss(MS-AV) < diss(MM-AV)$: The preference dissatisfaction values for Minisum-Ranksum are lower than for Minisum-Approval, and they are lower for Minisum-Approval than for Minimax-Approval.

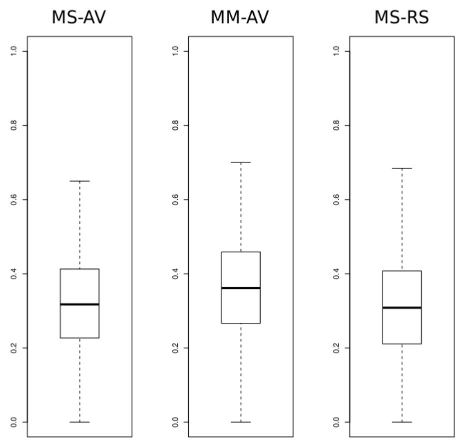

Figure 6.3: Dissatisfaction values for sequential_iterative, uniform preferences

Lastly, we compared the preference dissatisfaction values under the combination coordinated_basic for the three different rules with median group size 7, see Figure 6.4. As the Wilcoxon tests in the appendix show, the differences between the three rules are significant. We have the relation $diss(MS - RS) < diss(MS - AV) < diss(MM - AV)$: The preference dissatisfaction values for Minisum-Ranksum are lower than for Minisum-Approval, and they are lower for Minisum-Approval than for Minimax-Approval.

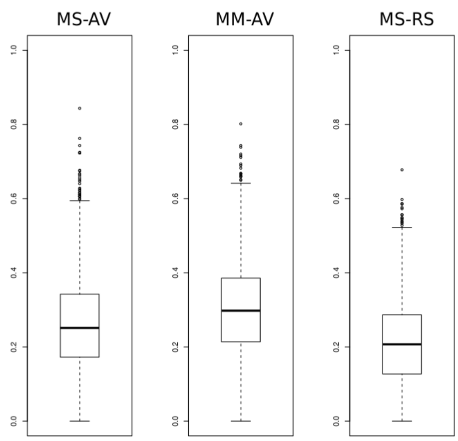

Figure 6.4: Dissatisfaction values for coordinated_basic, uniform preferences

These results indicate that for uniform preferences, Minisum-Ranksum yields better results regarding the preference dissatisfaction than Minisum-Approval and Minimax-Approval, and Minisum-Approval yields better results than Minimax-Approval.

Foursquare-based preferences For Foursquare-based preferences, the results regarding dissatisfaction values are similar, with one exception, namely the comparison between Minisum-Approval and Minimax-Approval for sequential_basic.

Like for uniform preferences, we first consider the preference dissatisfaction values under the combination sequential_basic, see Figure 6.5. The Wilcoxon test results in the appendix show that the difference between Minisum-Approval and Minisum-Ranksum is significant, but the difference between Minisum-Approval and Minimax-Approval is not significant. We have the relation $diss(MS-RS) < diss(MS-AV) = diss(MM-AV)$.

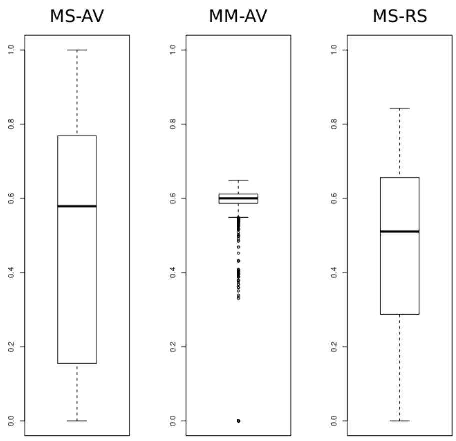

Figure 6.5: Dissatisfaction values for sequential_basic, Foursquare-based preferences

We also consider the preference dissatisfaction values under the combination sequential_iterative for median size 2, see Figure 6.6. The Wilcoxon test results in the appendix show that the differences between the three rules are significant. We have the relation $diss(MS-RS) < diss(MS-AV) < diss(MM-AV)$.

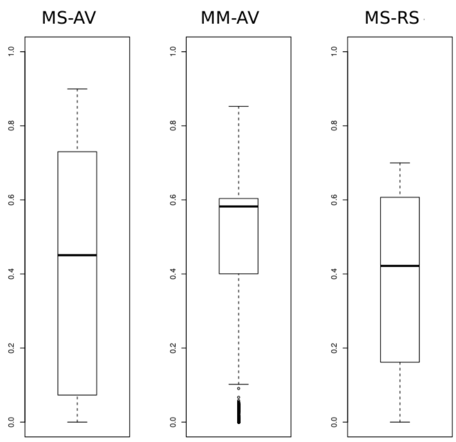

Figure 6.6: Dissatisfaction values for sequential_iterative, Foursquare-based preferences

Lastly, we consider the preference dissatisfaction values under the combination coordinated_basic for median size 5, see Figure 6.7. The Wilcoxon test results in the appendix show that the differences between the three rules are significant. We have the relation $diss(MS-RS) < diss(MS-AV) < diss(MM-AV)$.

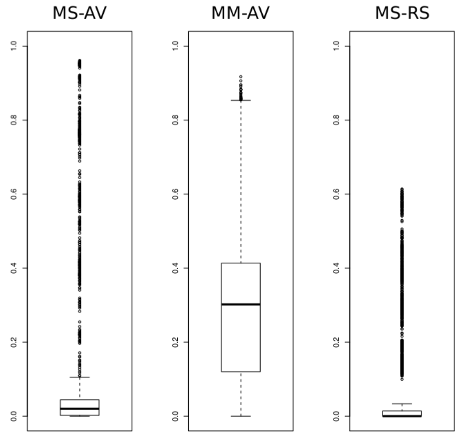

Figure 6.7: Dissatisfaction values for coordinated_basic, Foursquare-based preferences

These results indicate that for Foursquare-based preferences, Minisum-Ranksum always yields better results regarding the preference dissatisfaction than Minisum-Approval and Minimax-Approval. For sequential_basic, there is no significant difference between Minisum-Approval and Minimax-Approval. For coordinated_basic and sequential_iterative, Minisum-Approval yields better results than Minimax-Approval.

This means that for both preference types, Minisum-Ranksum yields better results than Minisum-Approval, and Minisum-Approval yields at least as good results as Minimax-Approval when considering preference dissatisfaction.

6.2.1.2 Group size

In this section we consider the question how the voting rules compare regarding group size. With group size, we mean the sizes of the groups after the respective last election, directly before they depart. We consider the target quantity separately for both preference forms. We omit the combination sequential_basic, because the group sizes are, predictably, the same under this combination for all three voting rules: because we directly fill up all groups and no voter is ever removed from a group, the groups usually achieve maximum capacity (unless there are too few travellers left for filling them up).

Uniform preferences First, we compare the group size under sequential_iterative, see Figure 6.8. We compare simulations with comparable median dissatisfaction value. According to the Wilcoxon tests in the appendix, Minisum-Approval yields larger groups than Minimax-Approval, but there is no significant difference between Minisum-Approval and Minisum-Ranksum. We have the relation $size(MM-AV) < size(MS-AV) = size(MS-RS)$.

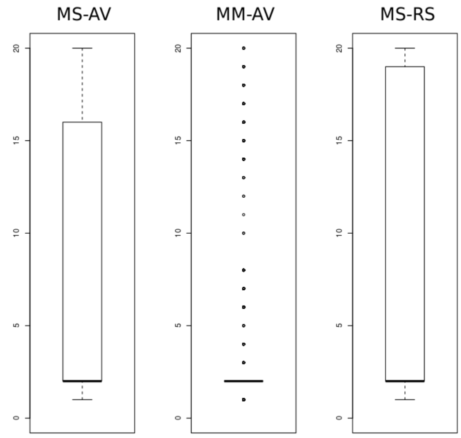

Figure 6.8: Size values for sequential_iterative, uniform preferences, comparable dissatisfaction

We also compared the group size for comparable dissatisfaction under coordinated_basic, see Figure 6.9. According to the Wilcoxon tests in the appendix, the difference between Minisum-Approval and Minimax-Approval is significant, whereas there is no significant difference between Minisum-Approval and Minisum-Ranksum. We have the relation $size(MM - AV) < size(MS - AV) = size(MS - RS)$.

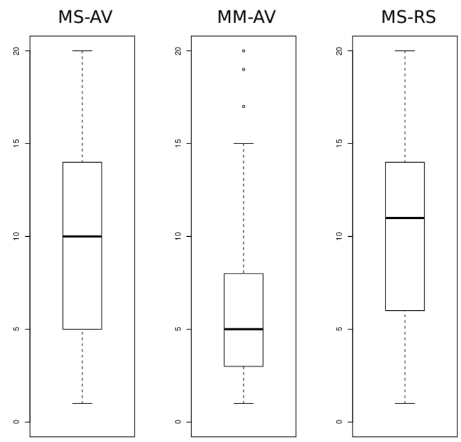

Figure 6.9: Size values for coordinated_basic, uniform preferences, comparable dissatisfaction

These results indicate that for uniform preferences under both sequential_iterative and coordinated_basic, Minisum-Approval yields better results than Minimax-Approval and Minisum-Ranksum yields as good results as Minisum-Approval regarding the group size.

Foursquare-based preferences First, we compared the size values for the three rules under sequential_iterative with comparable dissatisfaction, see Figure 6.10. According to the Wilcoxon test in the appendix, the difference between Minisum-Approval and

Minisum-Ranksum is significant. We have the relation $size(MS-RS) = size(MM-AV) < size(MS-AV)$.

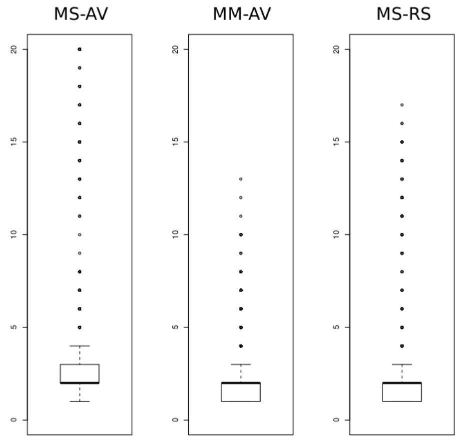

Figure 6.10: Size values for sequential_iterative, Foursquare-based preferences, comparable dissatisfaction

We also compared the size values for the three rules under coordinated_basic with comparable dissatisfaction values, see Figure 6.11.

According to the Wilcoxon test result in the appendix, the difference between Minisum-Approval and Minimax-Approval is significant. We have the relation $size(MS-AV) = size(MS-RS) > size(MM-AV)$.

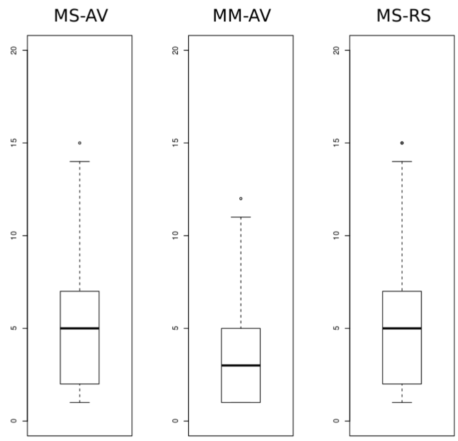

Figure 6.11: Size values for coordinated_basic, Foursquare-based preferences, comparable dissatisfaction

These results indicate that for Foursquare-based preferences, Minisum-Approval yields better results than Minimax-Approval regarding the group size. Minisum-Ranksum yields at least as good results as Minimax-Approval. For coordinated_basic, Minisum-Ranksum yields as good results as Minisum-Approval. For sequential_iterative, Minisum-Approval yields better results than Minisum-Ranksum.

When you consider both preference types, Minisum-Approval yields better results than Minimax-Approval. Minisum-Ranksum yields at least as good results as Minimax-Approval. Minisum-Ranksum yields as good results as Minisum-Approval regarding the group size for coordinated_basic. For sequential_iterative under uniform preferences, Minisum-Ranksum yields as good results as Minisum-Approval. For sequential_iterative under Foursquare-based preferences, Minisum-Approval yields better results than Minisum-Ranksum.

6.2.1.3 Organisational effort

In this section, we consider the question how the voting rules compare regarding organisational effort. Again, we consider the quantity separately for both preference forms. We omit the combination sequential_basic, because the organisational effort is the same under this combination for all three voting rules.

Uniform preferences First, we compared the organisational effort under the combination sequential_iterative for the three different rules with comparable dissatisfaction, see Figure 6.12. We compared three simulations for which the median dissatisfaction value was similar. According to the Wilcoxon tests in the appendix, the differences between the three rules are significant. We have the relation $orga(MS-RS) < orga(MS-AV) < orga(MM-AV)$.

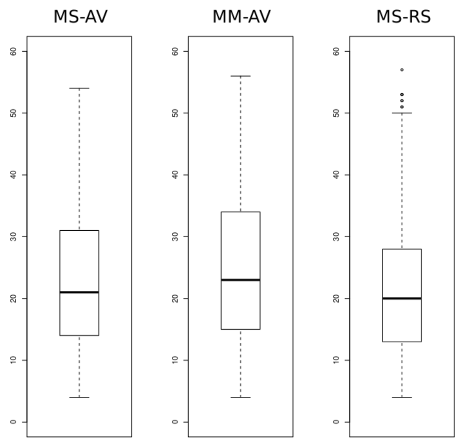

Figure 6.12: Organisational effort values for sequential_iterative, uniform preferences, comparable dissatisfaction

We also compared the organisational effort under coordinated_basic for the three different rules with comparable dissatisfaction, see Figure 6.13. According to the Wilcoxon tests in the appendix, the difference between Minimax-Approval and Minisum-Ranksum is significant, whereas the difference between Minisum-Approval and Minisum-Ranksum is not significant. We have the relation $orga(MS-RS) = orga(MS-AV) < orga(MM-AV)$.

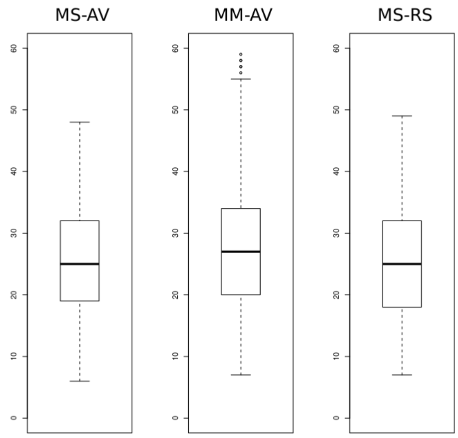

Figure 6.13: Organisational effort values for coordinated_basic, uniform preferences, comparable dissatisfaction

These results for uniform preferences indicate that for both sequential_iterative and coordinated_basic, Minisum-Approval yields better results than Minimax-Approval. For sequential_iterative, Minisum-Ranksum yields better results than Minisum-Approval. Under coordinated_basic, Minisum-Ranksum yields as good results as Minisum-Approval.

Foursquare-based preferences First, we compare the organisational effort values under sequential_iterative for comparable dissatisfaction under Minisum-Approval, Minimax-Approval and Minisum-Ranksum. In Figure 6.14, Minimax-Approval clearly yields higher values than Minisum-Approval and Minisum-Ranksum. Minisum-Ranksum yields higher values than Minisum-Approval. According to the Wilcoxon tests in the appendix, the differences between the three rules are significant. We have the relation $orga(MS-AV) < orga(MS-RS) < orga(MM-AV)$.

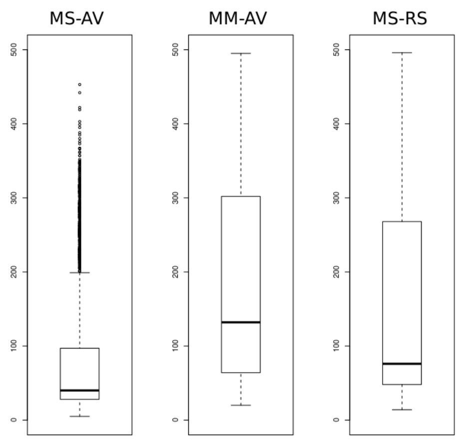

Figure 6.14: Organisational effort values for sequential_iterative, Foursquare-based preferences, comparable dissatisfaction

We also compared the organisational effort values under coordinated_basic for comparable dissatisfaction under Minisum-Approval, Minimax-Approval and Minisum-Ranksum, see Figure 6.15. According to the Wilcoxon test results in the appendix, the differences between the three rules are significant. We have the relation $orga(MS - RS) < orga(MS - AV) < orga(MM - AV)$.

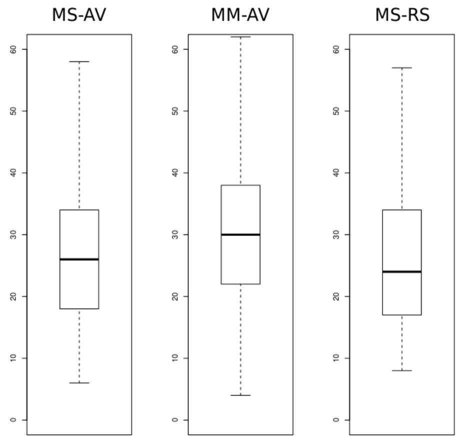

Figure 6.15: Organisational effort values for coordinated_basic, Foursquare-based preferences, comparable dissatisfaction

These results indicate that for Foursquare-based preferences under both sequential_iterative and coordinated_basic, Minisum-Ranksum and Minisum-Approval yield better results than Minimax-Approval regarding the organisational effort values. For sequential_iterative, Minisum-Approval yields better results than Minisum-Ranksum. For coordinated_basic, Minisum-Ranksum yields better results than Minisum-Approval.

When you consider both preference types, Minisum-Approval and Minisum-Ranksum yield better results than Minimax-Approval regarding organisational effort. Except for the case of sequential_iterative under Foursquare-based preferences, Minisum-Ranksum yields at least as good results as Minisum-Approval. For sequential_iterative under Foursquare-based preferences, Minisum-Approval yields better results.

6.2.2 Voting protocols

In this section, we compare the two voting protocols regarding their effect on preference dissatisfaction, organisational effort and group size. Like in the previous comparisons, we consider uniform and Foursquare-based preferences separately.

6.2.2.1 Uniform preferences

First, we compare some results for the basic and the iterative protocol under Minisum-Approval, see Table 6.4, which shows selected results for simulations under the combinations sequential_basic (sb) and sequential_iterative (si) (with different dissatisfaction thresholds).

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.33	14.00
si, dissThr 0.7	20.00	0.33	15.00
si, dissThr 0.65	7.00	0.32	16.00
si, dissThr 0.6	2.00	0.29	21.00

Table 6.4: Selected results for MS-AV, uniform, sb vs. si

For all dissatisfaction thresholds, it holds

- The iterative protocol yields equal or better results than the basic protocol in terms of traveller dissatisfaction
- The iterative protocol yields equal or smaller groups than the basic protocol
- The iterative protocol yields higher organisational effort values than the basic protocol

It can be noted that even for very small median group size, the median dissatisfaction under the iterative protocol is only slightly smaller than for the basic protocol.

Furthermore, we compare some results for the basic and the iterative protocol under Minimax-Approval, see Table 6.5.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.39	14.00
si, dissThr 0.8	20.00	0.37	15.00
si, dissThr 0.7	7.00	0.36	16.00
si, dissThr 0.5	2.00	0.28	41.00

Table 6.5: Selected results for MM-AV, uniform, sb vs. si

For all dissatisfaction thresholds, it holds

- The iterative protocol yields equal or better results than the basic protocol in terms of traveller dissatisfaction
- The iterative protocol yields equal or smaller groups than the basic protocol
- The iterative protocol yields higher organisational effort values than the basic protocol

It can be seen that larger changes in group size under the iterative protocol do not translate to a larger improvement in dissatisfaction in comparison to the basic protocol.

Lastly, we compare some results for the basic and the iterative protocol under Minisum-Ranksum, see Table 6.6.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.32	14.00
si, dissThr 0.8	20.00	0.32	15.00
si, dissThr 0.7	19.00	0.31	16.00
si, dissThr 0.685	7.00	0.30	16.00
si, dissThr 0.5	2.00	0.23	32.00

Table 6.6: Selected results for MS-RS, uniform, sb vs. si

For all dissatisfaction thresholds, it holds

- The iterative protocol yields equal or better results than the basic protocol in terms of traveller dissatisfaction
- The iterative protocol yields equal or smaller groups than the basic protocol
- The iterative protocol yields higher organisational effort values than the basic protocol

Again, it can be seen that larger changes in group size under the iterative protocol do not translate to a larger improvement in dissatisfaction in comparison to the basic protocol.

These results indicate that for uniform preferences, using the iterative protocol leads to improvement in preference dissatisfaction in comparison to the basic protocol, but this is gained by severely deteriorating the group size.

6.2.2.2 Foursquare-based preferences

For Foursquare-based preferences, the results of the comparisons between the basic and iterative protocol under the three voting rules are similar.

First, we compare some results for the basic and the iterative protocol under Minisum-Approval, see Table 6.7.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.58	14.00
si, dissThr 0.9	2.00	0.45	23.00
si, dissThr 0.5	2.00	0.03	106.50

Table 6.7: Selected results for MS-AV, Foursquare-based, sb vs. si

For all dissatisfaction thresholds, it holds

- The iterative protocol yields equal or better results than the basic protocol in terms of traveller dissatisfaction

- The iterative protocol yields equal or smaller groups than the basic protocol
- The iterative protocol yields higher organisational effort values than the basic protocol

Furthermore, we compare some results for the basic and the iterative protocol under Minimax-Approval, see Table 6.8.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.60	14.00
si, dissThr 0.65	19.00	0.60	16.00
si, dissThr 0.625	2.00	0.58	28.00
si, dissThr 0.5	2.00	0.31	132.00

Table 6.8: Selected results for MM-AV, Foursquare-based, sb vs. si

For all dissatisfaction thresholds, it holds

- The iterative protocol yields equal or better results than the basic protocol in terms of traveller dissatisfaction
- The iterative protocol yields equal or smaller groups than the basic protocol
- The iterative protocol yields higher organisational effort values than the basic protocol

It can be seen that even for very small group sizes, the dissatisfaction for the iterative protocol can be almost as high as for the basic protocol.

Lastly, we compare some results for the basic and the iterative protocol under Minisum-Ranksum, see Table 6.9.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.51	14.00
si, dissThr 0.8	20.00	0.51	15.00
si, dissThr 0.7	2.00	0.42	33.00

Table 6.9: Selected results for MS-RS, Foursquare-based, sb vs. si

For all dissatisfaction thresholds, it holds

- The iterative protocol yields equal or better results than the basic protocol in terms of traveller dissatisfaction
- The iterative protocol yields equal or smaller groups than the basic protocol

- The iterative protocol yields higher organisational effort values than the basic protocol

These results indicate that for Foursquare-based preferences, using the iterative protocol leads to improvement in preference dissatisfaction in comparison to the basic protocol, which is often gained by severely deteriorating the group size.

6.2.3 Grouping algorithms

In this section, we compare the grouping algorithms regarding their effect on preference dissatisfaction, organisational effort and group size. Again, we consider uniform and Foursquare-based preferences separately.

6.2.3.1 Uniform preferences

First, we compare some results for the sequential and the coordinated grouping approach under Minisum-Approval, see Table 6.10, which shows selected results for simulations under the combinations sequential_basic (sb) and coordinated_basic (cb) (with different join thresholds)

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.33	14.00
cb, joinThr 13	10.00	0.28	25.00
cb, joinThr 12	7.00	0.25	26.00

Table 6.10: Selected results for MS-AV, uniform, sb vs. cb

For all considered join thresholds, it holds

- Coordinated grouping yields better results than sequential grouping in terms of traveller dissatisfaction
- Coordinated grouping yields smaller groups than sequential grouping
- Coordinated grouping yields higher organisational effort values than sequential grouping

Note here that coordinated grouping fares comparatively well in terms of median dissatisfaction in relation to median group size.

Furthermore, we compare some results for the sequential and the coordinated grouping approach under Minimax-Approval, see Table 6.11,

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.39	14.00
cb, joinThr 11	5.00	0.27	27.00
cb, joinThr 10	3.00	0.23	27.00
cb, joinThr 9	2.00	0.20	30.00

Table 6.11: Selected results for MM-AV, uniform, sb vs. cb

For all considered join thresholds, it holds

- Coordinated grouping yields better results than sequential grouping in terms of traveller dissatisfaction
- Coordinated grouping yields smaller groups than sequential grouping
- Coordinated grouping yields higher organisational effort values than sequential grouping

Lastly, we compare some results for the sequential and the coordinated grouping approach under Minisum-Ranksum, see Table 6.12.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.32	14.00
cb, joinThr 50	11.00	0.26	25.00
cb, joinThr 40	7.00	0.21	28.00
cb, joinThr 30	4.00	0.13	34.00

Table 6.12: Selected results for MS-RS, uniform, sb vs. cb

For all considered join thresholds, it holds

- Coordinated grouping yields better results than sequential grouping in terms of preference dissatisfaction
- Coordinated grouping yields smaller groups than sequential grouping
- Coordinated grouping yields higher organisational effort than sequential grouping

Again, it can be noted that coordinated grouping fares comparatively well in terms of median dissatisfaction in relation to median group size.

These results indicate that for uniform preferences, coordinated grouping tends to fare comparatively well in terms of median dissatisfaction in relation to median group size.

6.2.3.2 Foursquare-based preferences

First, we compare some results for the sequential and the coordinated grouping approach, see Table 6.13.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.58	14.00
cb, joinThr 7	5.00	0.02	26.00
cb, joinThr 5	3.00	0.01	28.00

Table 6.13: Selected results for MS-AV, Foursquare-based, sb vs. cb

For all considered join thresholds, it holds

- Coordinated grouping yields better results than sequential grouping in terms of preference dissatisfaction
- Coordinated grouping yields smaller groups than sequential grouping
- Coordinated grouping yields higher organisational effort values than sequential grouping

Furthermore, we compare some results for the sequential and the coordinated grouping approach under Minimax-Approval, see Table 6.14.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.60	14.00
cb, joinThr 9	5.00	0.30	25.00
cb, joinThr 6	5.00	0.14	25.00
cb, joinThr 5	3.00	0.03	30.00

Table 6.14: Selected results for MM-AV, Foursquare-based, sb vs. cb

For all considered join thresholds, it holds

- Coordinated grouping yields better results than sequential grouping in terms of preference dissatisfaction
- Coordinated grouping yields smaller groups than sequential grouping
- Coordinated grouping yields higher organisational effort values than sequential grouping

Lastly, we compare some results for the sequential and the coordinated grouping approach, see Table 6.15.

Considered	Median size	Median dissatisfaction	Median organisational effort
sb	20.00	0.51	14.00
cb, joinThr 50	10.00	0.33	19.00
cb, joinThr 40	7.00	0.20	21.00
cb, joinThr 30	6.00	0.13	29.00
cb, joinThr 20	5.00	0.00	24.00

Table 6.15: Selected results for MS-RS, Foursquare-based, sb vs. cb

For all considered join thresholds, it holds

- Coordinated grouping yields better results than sequential grouping in terms of preference dissatisfaction
- Coordinated grouping yields smaller groups than sequential grouping
- Coordinated grouping yields higher organisational effort than sequential grouping

It can be noted here again that the coordinated grouping fares comparatively well: Even for simulations with median size $\geq 1/2$ median size under sequential grouping, the dissatisfaction is clearly lower than under sequential grouping.

These results indicate that for Foursquare-based preferences, as for uniform preferences, the coordinated grouping tends to fare comparatively well in terms of median dissatisfaction in relation to median group size.

6.2.4 Combinations

In this section we investigate how different combinations of grouping algorithms and voting protocols compare regarding their effects on preference dissatisfaction, organisational effort and group size.

6.2.4.1 Preference dissatisfaction

In the following, we consider the quantity preference dissatisfaction. We again consider it separately for uniform and Foursquare-based preferences.

Uniform preferences First, we compared the preference dissatisfaction values under sequential_iterative and coordinated_basic for Minisum-Approval with median group size 7. As can be seen in Figure 6.16, coordinated_basic yields lower dissatisfaction values than sequential_iterative. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

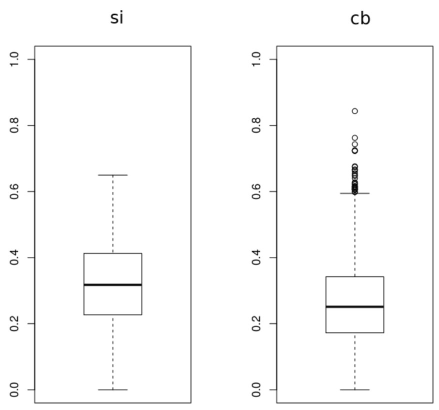

Figure 6.16: Dissatisfaction values for sequential_iterative and coordinated_basic for MS-AV

Furthermore, we compared the preference dissatisfaction values under sequential_iterative and coordinated_basic for Minimax-Approval with median group size 7. As can be seen in Figure 6.17, coordinated_basic again yields lower dissatisfaction values than sequential_iterative. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

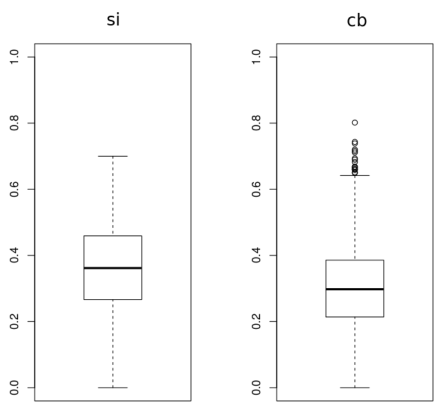

Figure 6.17: Dissatisfaction values for sequential_iterative and coordinated_basic for MM-AV

Lastly, we compared the dissatisfaction values under sequential_iterative and coordinated_basic for Minisum-Ranksum with median group size 7. As can be seen in Figure 6.18, coordinated_basic again yields lower dissatisfaction values than sequential_iterative. As the Wilcoxon test in Section 10.3.4 shows, the difference between sequential_iterative and coordinated_basic is significant.

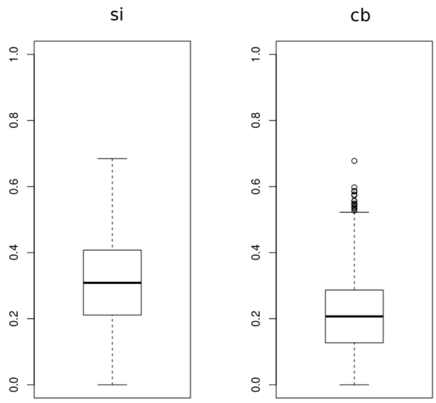

Figure 6.18: Dissatisfaction values for sequential_iterative and coordinated_basic for MS-RS

These results indicate that under uniform preferences, for comparable group size, coordinated_basic yields better results than sequential_iterative regarding preference dissatisfaction for all three voting rules.

Foursquare-based preferences First, we considered the preference dissatisfaction values for Minisum-Approval. For the considered simulations for Minisum-Approval with Foursquare-based preferences, coordinated_basic is better regarding preference dissatisfaction than sequential_iterative: The maximum median dissatisfaction for simulations under coordinated_basic with median group size 5 is smaller than the minimum median dissatisfaction for simulations under sequential_iterative with median group size 2 (0.02 vs. 0.03).

Furthermore, we considered the preference dissatisfaction values for Minimax-Approval. For the considered simulations for Minimax-Approval with Foursquare-based preferences, coordinated_basic is better regarding preference dissatisfaction than sequential_iterative: The maximum median dissatisfaction for simulations under coordinated_basic with median group size 3 is smaller than the minimum median dissatisfaction for simulations under sequential_iterative with median group size 2 (0.03 vs. 0.31).

Lastly, we considered the preference dissatisfaction values for Minisum-Ranksum. For the considered simulations for Minisum-Ranksum with Foursquare-based preferences, coordinated_basic is better regarding preference dissatisfaction than sequential_iterative: The maximum median dissatisfaction for simulations under coordinated_basic with median group size 5 is smaller than the minimum median dissatisfaction for simulations under sequential_iterative with median group size 2 (0.00 vs. 0.25).

These results indicate that, as under uniform preferences, under Foursquare-based preferences, for comparable group size, coordinated_basic yields better results than sequential_iterative regarding preference dissatisfaction for all three voting rules.

6.2.4.2 Group size

In the following, we consider the quantity group size, again separately for uniform and Foursquare-based preferences.

Uniform preferences First, we compared the group size values under sequential_iterative and coordinated_basic for Minisum-Approval for comparable dissatisfaction. Here, sequential_iterative yields smaller groups than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

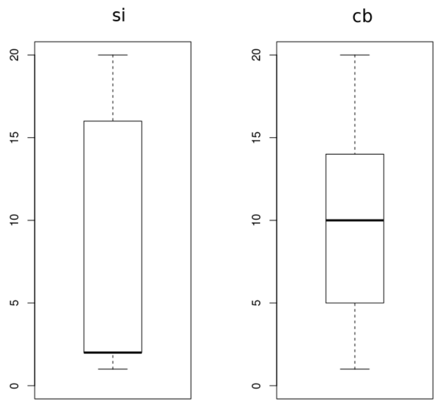

Figure 6.19: Size values for sequential_iterative and coordinated_basic for MS-AV, comparable dissatisfaction

Furthermore, we compared the size values under sequential_iterative and coordinated_basic for Minimax-Approval for comparable dissatisfaction. Here, sequential_iterative yields smaller group sizes than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

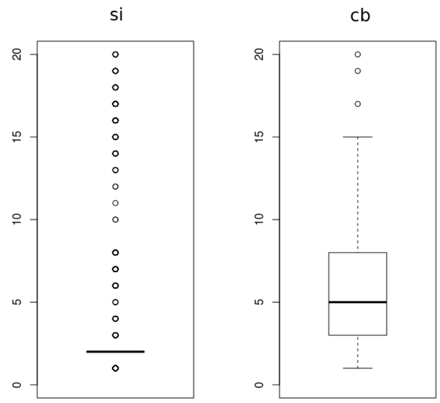

Figure 6.20: Size values for sequential_iterative and coordinated_basic for MM-AV, comparable dissatisfaction

Lastly, we compared the size values under sequential_iterative and coordinated_basic for Minisum-Ranksum for comparable dissatisfaction. Here, sequential_iterative yields smaller group sizes than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

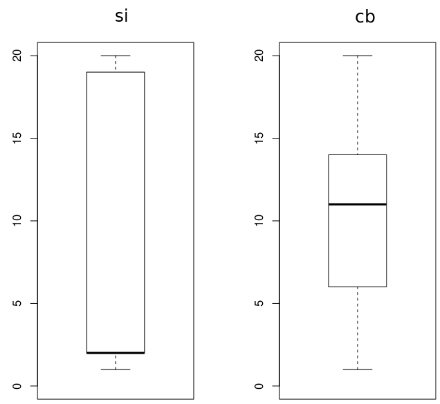

Figure 6.21: Size values for sequential_iterative and coordinated_basic for MS-RS, comparable dissatisfaction

These results indicate that for uniform preferences, coordinated_basic yields better results than sequential_iterative regarding group size.

Foursquare-based preferences For Foursquare-based preferences, the results regarding group size are similar.

First, we compared the group size for Minisum-Approval under sequential_iterative and coordinated_basic for comparable dissatisfaction. Clearly, sequential_iterative yields smaller group sizes than coordinated_basic. Acccording to the Wilcoxon test in the appendix, the difference between sequential_iterative and coordinated_basic is significant.

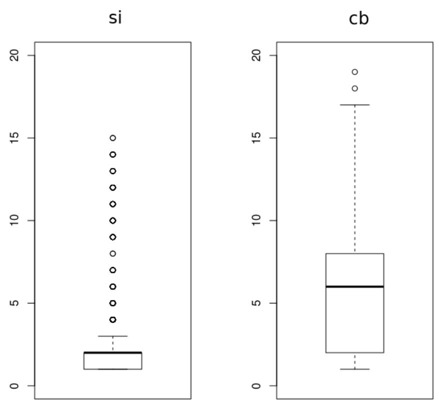

Figure 6.22: Size values for sequential_iterative and coordinated_basic for MS-AV, comparable dissatisfaction

Furthermore, we compared the group size for Minimax-Approval under sequential_iterative and coordinated_basic for comparable dissatisfaction. As Figure 6.23 shows, the group sizes for sequential_iterative are smaller than for coordinated_basic. According to the Wilcoxon test in the appendix, the difference between sequential_iterative and coordinated_basic is significant.

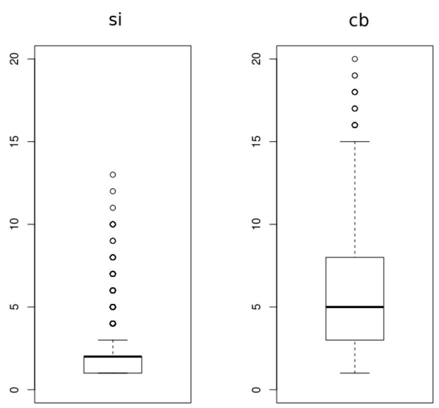

Figure 6.23: Size values for sequential_iterative and coordinated_basic for MM-AV, comparable dissatisfaction

Lastly, we compared the size values for Minisum-Ranksum under sequential_iterative and coordinated_basic for comparable dissatisfaction. As Figure 6.24 shows, sequential_iterative yields smaller groups than coordinated_basic. According to the Wilcoxon test in the appendix, the difference between sequential_iterative and coordinated_basic is significant.

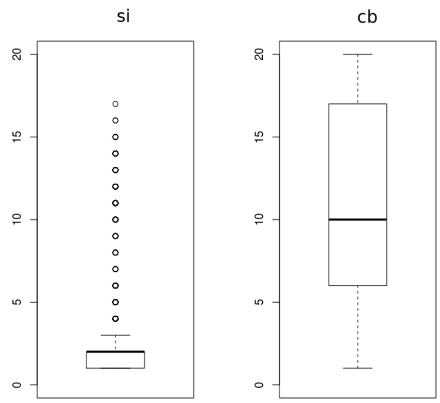

Figure 6.24: Size values for sequential_iterative and coordinated_basic for MS-RS, comparable dissatisfaction

These results indicate that, as for uniform preferences, for Foursquare-based preferences, coordinated_basic yields better results than sequential_iterative regarding group size. This also means that for both preference types, coordinated_basic yields better results than sequential_iterative regarding group size.

6.2.4.3 Organisational effort

In the following, we consider the quantity organisational effort, again separately for uniform and Foursquare-based preferences.

Uniform preferences First, we compared the organisational effort values under sequential_iterative and coordinated_basic for Minisum-Approval for comparable dissatisfaction. Here, sequential_iterative yields lower organisational effort values than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

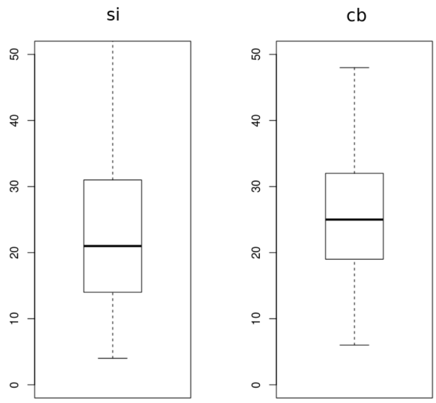

Figure 6.25: Organisational effort values for sequential_iterative and coordinated_basic for MS-AV, comparable dissatisfaction

Furthermore, we compared the organisational effort values under sequential_iterative and coordinated_basic for Minimax-Approval for comparable dissatisfaction. Here, sequential_iterative yields higher organisational effort values than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

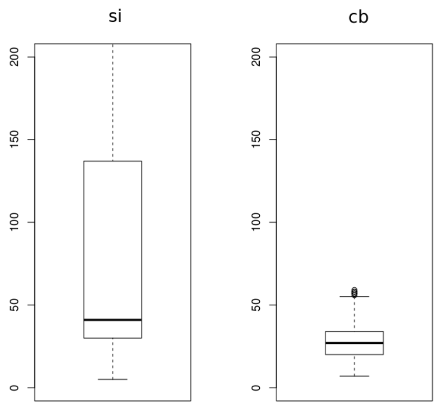

Figure 6.26: Organisational effort values for sequential_iterative and coordinated_basic for MM-AV, comparable dissatisfaction

Lastly, we compared the organisational effort values under sequential_iterative and coordinated_basic for Minisum-Ranksum for comparable dissatisfaction. Here, sequential_iterative yields higher organisational effort values than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

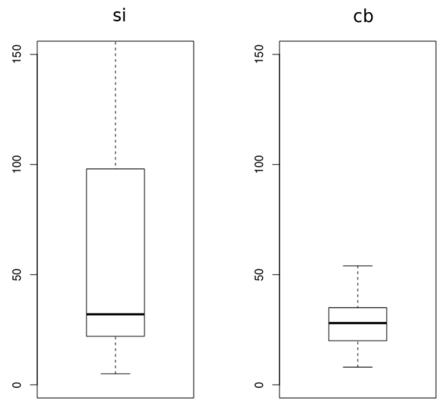

Figure 6.27: Organisational effort values for sequential_iterative and coordinated_basic for MS-RS, comparable dissatisfaction

These results indicate that for uniform preferences, coordinated_basic yields better results than sequential_iterative regarding organisational effort values, except for Minisum-Approval. For Minisum-Approval, sequential_iterative yields better results.

Foursquare-based preferences First, we compared the organisational effort under sequential_iterative and coordinated_basic for Minisum-Approval for comparable dissatisfaction. Here, sequential_iterative yields higher organisational effort than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

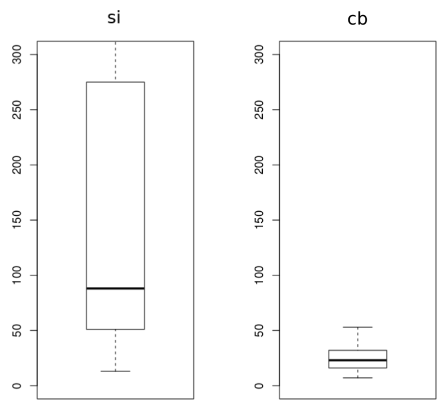

Figure 6.28: Organisational effort values for sequential_iterative and coordinated_basic for MS-AV, comparable dissatisfaction

Furthermore, we compared the organisational effort under sequential_iterative and coordinated_basic for Minimax-Approval for comparable dissatisfaction. Here, sequential_iterative yields higher organisational effort than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

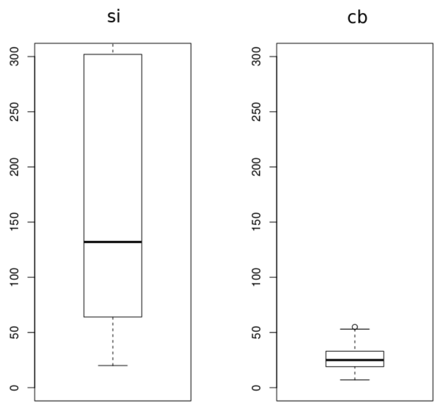

Figure 6.29: Organisational effort values for sequential_iterative and coordinated_basic for MM-AV, comparable dissatisfaction

Lastly, we compared the organisational effort under sequential_iterative and coordinated_basic for Minisum-Ranksum for comparable dissatisfaction. Here, sequential_iterative yields higher organisational effort than coordinated_basic. As the Wilcoxon test in the appendix shows, the difference between sequential_iterative and coordinated_basic is significant.

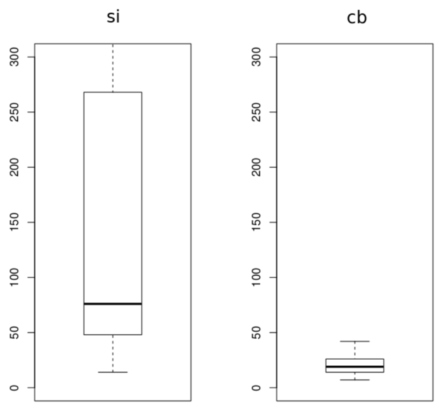

Figure 6.30: Organisational effort values for sequential_iterative and coordinated_basic for MS-RS, comparable dissatisfaction

These results indicate that for Foursquare-based preferences, coordinated_basic always yields better results than sequential_iterative regarding organisational effort.

This means that for both preference types, coordinated_basic yields better results than sequential_iterative regarding organisational effort, except for Minisum-Approval under uniform preferences. For Minisum-Approval under uniform preferences, sequential_iterative yields better results.

6.2.5 Summary of results for main series

The following table and the following paragraphs summarise the results for the main series.

Comparing	Uniform preferences	Foursquare-based preferences
Voting rules, dissatisfaction under sequential_basic	$diss_{sb}(\textit{MS-RS}) < diss_{sb}(\textit{MS-AV}) < diss_{sb}(\textit{MM-AV})$	$diss_{sb}(\textit{MS-RS}) < diss_{sb}(\textit{MS-AV}) = diss_{sb}(\textit{MM-AV})$
Voting rules, dissatisfaction under sequential_iterative	$diss_{si}(\textit{MS-RS}) < diss_{si}(\textit{MS-AV}) < diss_{si}(\textit{MM-AV})$	$diss_{si}(\textit{MS-RS}) < diss_{si}(\textit{MS-AV}) < diss_{si}(\textit{MM-AV})$
Voting rules, dissatisfaction under coordinated_basic	$diss_{cb}(\textit{MS-RS}) < diss_{cb}(\textit{MS-AV}) < diss_{cb}(\textit{MM-AV})$	$diss_{cb}(\textit{MS-RS}) < diss_{cb}(\textit{MS-AV}) < diss_{cb}(\textit{MM-AV})$
Voting rules, size under sequential_iterative	$size_{si}(\textit{MS-RS}) = size_{si}(\textit{MS-AV}) > size_{si}(\textit{MM-AV})$	$size_{si}(\textit{MS-RS}) = size_{si}(\textit{MM-AV}) < size_{si}(\textit{MS-AV})$
Voting rules, size under coordinated_basic	$size_{cb}(\textit{MS-RS}) = size_{cb}(\textit{MS-AV}) > size_{cb}(\textit{MM-AV})$	$size_{cb}(\textit{MS-RS}) = size_{cb}(\textit{MS-AV}) > size_{cb}(\textit{MM-AV})$
Voting rules, org. effort under sequential_iterative	$orga_{si}(\textit{MS-RS}) < orga_{si}(\textit{MS-AV}) < orga_{si}(\textit{MM-AV})$	$orga_{si}(\textit{MS-AV}) < orga_{si}(\textit{MS-RS}) < orga_{si}(\textit{MM-AV})$
Voting rules, org. effort under coordinated_basic	$orga_{cb}(\textit{MS-RS}) = orga_{cb}(\textit{MS-AV}) < orga_{cb}(\textit{MM-AV})$	$orga_{cb}(\textit{MS-RS}) < orga_{cb}(\textit{MS-AV}) < orga_{cb}(\textit{MM-AV})$
Voting protocols, dissatisfaction for all rules	$diss(si) < diss(sb)$	$diss(si) < diss(sb)$
Voting protocols, size for all rules	$size(si) < size(sb)$	$size(si) < size(sb)$
Voting protocols, org. effort for all rules	$orga(si) > orga(sb)$	$orga(si) > orga(sb)$
Grouping algorithms, dissatisfaction for all rules	$diss(cb) < diss(sb)$	$diss(cb) < diss(sb)$
Grouping algorithms, size for all rules	$size(cb) < size(sb)$	$size(cb) < size(sb)$
Grouping algorithms, org. effort for all rules	$orga(cb) > orga(sb)$	$orga(cb) > orga(sb)$
Combinations, dissatisfaction for all rules	$diss(cb) < diss(si)$	$diss(cb) < diss(si)$
Combinations, size for all rules	$size(cb) > size(si)$	$size(cb) > size(si)$
Combinations, org. effort under rule Minisum-Approval	$orga_{\textit{MS-AV}}(cb) > orga_{\textit{MS-AV}}(si)$	$orga_{\textit{MS-AV}}(cb) < orga_{\textit{MS-AV}}(si)$
Combinations, org. effort under rule Minimax-Approval	$orga_{\textit{MM-AV}}(cb) < orga_{\textit{MM-AV}}(si)$	$orga_{\textit{MM-AV}}(cb) < orga_{\textit{MM-AV}}(si)$
Combinations, org. effort under rule Minisum-Ranksum	$orga_{\textit{MS-RS}}(cb) < orga_{\textit{MS-RS}}(si)$	$orga_{MS-RS}(cb) < orga_{MS-RS}(si)$

Voting rules For the considered simulations, Minisum-Approval and Minisum-Ranksum yielded at least as good results as Minimax-Approval. Minisum-Ranksum yielded at least as good results as Minisum-Approval, except for effort and size for sequential_iterative under Foursquare-based preferences. This means that we have the following observations.

O1.1: *Minisum-Approval and Minisum-Ranksum yielded better or as good results as Minimax-Approval.*

O1.2: *Minisum-Ranksum yielded at least as good results as Minisum-Approval, except for effort and size for sequential_iterative under Foursquare-based preferences.*

Voting protocols We compared the basic protocol with the iterative protocol under all three rules for both uniform and Foursquare-based preferences. The iterative protocol always yielded at most as high dissatisfaction values as the basic protocol, at most as large groups as the basic protocol and at least as high organisational effort values as the basic protocol.

Note that under the iterative voting protocol, lower dissatisfaction thresholds lead to lower preference dissatisfaction, but to smaller groups and higher organisational effort. In many cases, the improvement in dissatisfaction values came at the cost of very small groups. This means that we have the following observation:

O2: *The iterative protocol often led to an improvement in preference dissatisfaction at the price of a strong deterioration in group size.*

Grouping algorithms We compared the sequential grouping with the coordinated grouping for all three voting rules for both uniform and Foursquare-based preferences. The coordinated grouping always yielded at most as high dissatisfaction values as the basic protocol, at most as large groups as the basic protocol and at least as high organisational effort as the basic protocol.

When using coordinated grouping, lower join thresholds lead to lower preference dissatisfaction, but to smaller groups and higher organisational effort. It can be noted that in many cases, there is an improvement in dissatisfaction values for the coordinated grouping in comparison to the sequential grouping, for relatively large groups. This means that we have the following observation:

O3: *The coordinated grouping often led to an improvement in preference dissatisfaction, while there was no strong deterioration in group size.*

Combinations Lastly, we compared the combinations sequential_iterative and coordinated_basic for all three voting rules for both uniform and Foursquare-based preferences. For the considered simulations we found that coordinated_basic always yielded smaller dissatisfaction values than sequential_iterative for comparable group size values, coordinated_basic always yielded larger groups than sequential_iterative for comparable dissatisfaction values and coordinated_basic yielded in most cases lower organisational effort values than sequential_iterative for comparable dissatisfaction, except for the case uniform preferences, Minisum-Approval. This means that we have the following observations:

O4: *In most cases, coordinated_basic yielded better results than sequential_iterative.*

O4.1: *coordinated_basic always yielded better results than sequential_iterative regarding the quantities preference dissatisfaction and group size.*

O4.2: *coordinated_basic yielded better results than sequential_iterative regarding the quantity organisational effort, except for the case uniform preferences, Minisum-Approval.*

6.2.6 Discussion for main simulation series

We compared the impact of different voting algorithms (voting rules, voting protocols, grouping algorithms and combinations) on the preference dissatisfaction of the travellers, the sizes of the resulting travel groups and the organisational effort for the travellers. From our observations regarding the results, we generate the following hypotheses:

H1: *Minisum-Approval and Minisum-Ranksum yield better than or as good results as Minimax-Approval, and except for sequential_iterative under Foursquare-based preferences, Minisum-Ranksum yields at least as good results as Minisum-Approval.*

H2: *The iterative protocol tends to lead to an improvement in preference dissatisfaction which comes at the price of a strong deterioration in group size.*

H3: *The coordinated grouping tends to lead to an improvement in preference dissatisfaction, while there is no strong deterioration in group size.*

H4: *The combination coordinated_basic tends to yield better results than sequential_iterative.*

H4.1: *The combination coordinated_basic yields better results than sequential_iterative regarding the quantities preference dissatisfaction and group size.*

H4.2: *The combination coordinated_basic tends to yield better results than sequential_iterative regarding the quantity organisational effort.*

In the terms of [Carley, 1999], we considered the key input parameters voting rule, voting protocol and grouping algorithm for the outputs preference dissatisfaction, group size and organisational effort.

As mentioned in Chapter 3, we suggest to use the hypotheses formulated here for both generating recommendations for designers of real-world applications and as basis for further research. Based on the results on hand, we formulate the following recommendations:

For already defined groups, we recommend using Minisum-Ranksum to reduce dissatisfaction. If the aim were to create large groups with low dissatisfaction, one could use the combination sequential_basic with Minisum-Ranksum. However, if one aims at a compromise between user and system goals, i.e. at improving preference satisfaction without reducing the group sizes too much, it would be recommended to use Minisum-Ranksum with the basic protocol and the coordinated grouping algorithm[1]. This means that, assuming that the goal is to achieve a compromise between the goals of the traffic management and the traffic participants, it would make sense in a real-world implementation to use the voting rule Minisum-Ranksum in combination with coordinated grouping and the basic voting protocol.

Based on the hypotheses, we formulate some recommendations for designers of real-world voting-based applications for building groups of vehicles and agreeing on common destinations. The last recommendation is of particular interest for us because it yields a compromise between system and user goals.

We formulate one recommendation for the case that we consider already defined groups and two recommendations for the case that we consider how the groups are obtained.

> IF the requirement is *low dissatisfaction for defined groups*,
> THEN the recommended configuration is *the voting rule Minisum-Ranksum*

> IF the requirement is *as large groups as possible with avoiding high dissatisfaction*,
> THEN the recommended configuration is *the voting rule Minisum-Ranksum with the combination sequential_basic*

> IF the requirement is *low dissatisfaction with avoiding small groups*,
> THEN the recommended configuration is *the voting rule Minisum-Ranksum with the combination coordinated_basic*

Regarding further research, the results suggest that it makes sense to conduct additional experiments for Minisum-Ranksum under the combination coordinated_basic, because the results for Minisum-Ranksum under coordinated_basic seem promising in regard to a compromise between system and user goals. In particular, it would be useful to compare the effects of Minisum-Ranksum under coordinated_basic with other algorithms, conducting simulations with different numbers of travellers, different numbers of alternatives and different group capacities to investigate whether the hypotheses generated here hold true for other settings and under which conditions they hold true. Further research also should include a more thorough investigation of the effects of different thresholds.

A limitation of our investigations is that only some selected voting algorithms were considered. For further research, additionally to further exploring Minisum-Ranksum under coordinated_basic, it might be reasonable to investigate additional voting algorithms, for example further voting rules, more refined iterative voting protocols or more refined coordinated grouping algorithms.

[1]Note that according to observation O4.2, this configuration is also expected to lead to lower organisational effort than when applying sequential_iterative.

Chapter 7

Travel costs under different combinations

One advantage of the voting-based approach is that grouping travellers together in groups leads to lower travel costs. In the main simulation series, we did not investigate travel costs and disregarded the actual routing of the travellers. In this section, we demonstrate by means of some examples how the travel costs (based on travel time) differ for single traveller agents and travel groups under different combinations of grouping algorithms and voting protocols. In order to do so, we use TSP approximations for computing the routes.

OSM node	POI name
6018	POI_1 American Museum of Natural History
6895	POI_2 Aritzia Mall 600 5th Ave
6115	POI_3 Brookfield Place
5559	POI_4 Bryant Park
2702	POI_5 Carnegie Hall
4986	POI_6 Central Park
4476	POI_7 Chelsea Market
5461	POI_8 Federal Hall National Memorial
2145	POI_9 Former Location of Edward Hopper's Studio
6963	POI_10 High Line 10th Ave Amphitheatre
3290	POI_11 Madison Square Park
5276	POI_12 Metropolitan Museum of Art
4657	POI_13 Museum of Modern Art (MoMA)
6119	POI_14 One World Observatory
6157	POI_15 St. Patrick's Cathedral
4729	POI_16 Top of the Rock Observation Deck
6361	POI_17 TurnStyle Underground Market
3505	POI_18 Under The Brooklyn Bridge
3291	POI_19 Union Square Park
3321	POI_20 Whitney Museum of American Art
7090	Start and End - Grand Central Terminal

Table 7.1: Considered sites as table

To this end, we use a Quantum Geographic Information System (QGIS)[1] database con-

[1] https://www.qgis.org/en/site/

taining the Manhattan network of OpenStreetMap (OSM)[2] nodes. In particular, we consider the POIs\OSM nodes in Table 7.1, which are also depicted in Figure 7.1. Note that we assume the Grand Central Terminal as start and end point for the tour and that the travel groups are formed there.

Figure 7.1: Considered sites
Map data © OpenStreetMap contributors (https://www.openstreetmap.org/copyright)

In QGIS, it is possible to conduct TSP approximations using pgrouting[3]. We give an example for a TSP approximation using all 21 OSM nodes:

```
SELECT * FROM pgr_TSP(
    $$
    SELECT * FROM pgr_dijkstraCostMatrix(
        'SELECT osm_id as id, source, target, to_time_cost as cost,
        reverse_time_cost as reverse_cost, x1, y1, x2, y2
        FROM manhattan_ways',
        ARRAY [6018, 6895, 3291, 6115, 5559, 2702, 4986, 4476, 6361,
        5461, 2145, 6963, 3290, 5276, 4657, 6119, 6157, 4729, 3505,
        3321, 7090],
```

[2] https://www.openstreetmap.org
[3] https://pgrouting.org/

```
        directed := false)
    $$,
    start_id := 7090,
    randomize := false
);
```

We compare the travel costs for the case that travellers drive alone with the case that travellers drive in groups. For better comparability of the results we only consider Minisum-k-Approval, a variant where each voter/traveller accepts exactly k candidates/POIs, so that each traveller will always visit exactly k POIs, even if driving alone.

From the travel times and group sizes, we derive the costs: For the sake of simplicity, we assume that a time unit corresponds to a cost unit. The travel cost for a visitor is computed as the sum of cost units for the route divided by group size. The larger the groups, the smaller the travel costs will be.

7.1 Example simulation - driving alone

The first simulation aims at yielding an example how travel costs behave for the case that the travellers travel alone. To this end, we run a simulation series with 1 run, 41 travellers, 20 POIs, group capacity of 1, committee size of 5 and Foursquare-based preferences. In the following table, we present the resulting committees for the different travellers, the corresponding OSM nodes and the travel cost for each committee. For this example simulation with all agents driving alone, the median individual travel cost is 1,351.

agent	committee	corresponding OSM nodes	cost
0	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
1	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
2	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
3	{4, 5, 6, 7, 8}	[2702, 4986, 4476, 5461, 2145]	1,598.73
4	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
5	{8, 16, 17, 18, 19}	[2145, 6361, 3505, 3291, 3321]	1,299.26
6	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
7	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
8	{0, 1, 2, 3, 8}	[6018, 6895, 6115, 5559, 2145]	1,362.42
9	{0, 4, 5, 6, 7}	[6018, 2702, 4986, 4476, 5461]	1,595.61
10	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
11	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
12	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
13	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
14	{8, 16, 17, 18, 19}	[2145, 6361, 3505, 3291, 3321]	1,299.26
15	{0, 1, 2, 3, 8}	[6018, 6895, 6115, 5559, 2145]	1,362.42
16	{0, 1, 2, 3, 8}	[6018, 6895, 6115, 5559, 2145]	1,362.42
17	{0, 1, 2, 3, 8}	[6018, 6895, 6115, 5559, 2145]	1,362.42
18	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
19	{0, 4, 5, 6, 7}	[6018, 2702, 4986, 4476, 5461]	1,595.61

20	{0, 1, 2, 3, 8}	[6018, 6895, 6115, 5559, 2145]	1,362.42
21	{8, 9, 10, 11, 16}	[2145, 6963, 3290, 5276, 6361]	1,081.41
22	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
23	{0, 4, 5, 6, 7}	[6018, 2702, 4986, 4476, 5461]	1,595.61
24	{0, 4, 5, 6, 7}	[6018, 2702, 4986, 4476, 5461]	1,595.61
25	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
26	{8, 9, 10, 11, 16}	[2145, 6963, 3290, 5276, 6361]	1,081.41
27	{4, 8, 9, 10, 11}	[2702, 2145, 6963, 3290, 5276]	1,039.84
28	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
29	{4, 5, 6, 7, 8}	[2702, 4986, 4476, 5461, 2145]	1,598.73
30	{4, 8, 9, 10, 11}	[2702, 2145, 6963, 3290, 5276]	1,083.44
31	{8, 16, 17, 18, 19}	[2145, 6361, 3505, 3291, 3321]	1,299.26
32	{4, 5, 6, 7, 8}	[2702, 4986, 4476, 5461, 2145]	1,598.73
33	{0, 8, 9, 10, 11}	[6018, 2145, 6963, 3290, 5276]	1,104,21
34	{4, 5, 6, 7, 8}	[2702, 4986, 4476, 5461, 2145]	1,598.73
35	{8, 9, 10, 11, 16}	[2145, 6963, 3290, 5276, 6361]	1,081.41
36	{0, 4, 5, 6, 7}	[6018, 2702, 4986, 4476, 5461]	1,595.61
37	{4, 5, 6, 7, 8}	[2702, 4986, 4476, 5461, 2145]	1,598.73
38	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
39	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
40	{8, 16, 17, 18, 19}	[2145, 6361, 3505, 3291, 3321]	1,299.26

7.2 Example simulation - driving in groups, sequential_basic

The second simulation series aims at yielding an example how travel costs behave for the combination of sequential grouping and the basic voting protocol. We run a simulation series with 1 run, 41 travellers, 20 POIs, group capacity of 20, committee size k of 5 and Foursquare-based preferences.

In the following, we present the resulting committees for the different groups.

Group 0 with 20 agents elects the committee {0, 1, 2, 3, 8}, which corresponds to the POIs as shown in Table 7.3.

OSM node	POI name
6018	POI_1 American Museum of Natural History
6115	POI_3 Brookfield Place
5559	POI_4 Bryant Park
2145	POI_9 Former Location of Edward Hopper's Studio

Table 7.3: POIs for sequential_basic, group 0

Group 1 with 20 agents elects the committee {0, 4, 5, 6, 8}, which corresponds to thePOIs as shown in Table 7.4.

OSM node	POI name
6018	POI_1 American Museum of Natural History
2702	POI_5 Carnegie Hall
4986	POI_6 Central Park
4476	POI_7 Chelsea Market
2145	POI_9 Former Location of Edward Hopper's Studio

Table 7.4: POIs for sequential_basic, group 1

Finally, Group 2 with one agent elects the committee {8, 16, 17, 18, 19}, which corresponds to the POIs as shown in Table 7.5.

OSM node	POI name
2145	POI_9 Former Location of Edward Hopper's Studio
6963	POI_10 High Line 10th Ave Amphitheatre
6361	POI_17 TurnStyle Underground Market
3505	POI_18 Under The Brooklyn Bridge
3291	POI_19 Union Square Park
3321	POI_20 Whitney Museum of American Art

Table 7.5: POIs for sequential_basic, group 2

Based on the selected POIs, we compute for each group the travel costs. Here, we define the travel time based on the travel time and use TSP approximations. In each computation, we use the POI Grand Central Terminal (with OSM ID 7090) as start and end point.

For example, for group 0, the costs are computed in QGIS as follows:

```
SELECT * FROM pgr_TSP(
$$
SELECT * FROM pgr_dijkstraCostMatrix(
'SELECT osm_id as id, source, target, to_time_cost as cost,
reverse_time_cost as reverse_cost, x1, y1, x2, y2
FROM manhattan_ways',
ARRAY [6018, 6895, 6115, 5559, 2145, 7090],
directed := false)
$$,
start_id := 7090,
randomize := false
);
```

This leads us to the following costs for all three groups.

Group	Overall cost	Cost per traveller
Group 0	1,362.42	68.12
Group 1	1,144.12	57.21
Group 2	1,299.26	1,299.26

Table 7.6: Costs for simulation series with sequential_basic

For this example simulation, the median individual travel cost is 68.12.

7.3 Example simulation - driving in groups, sequential_iterative

The third simulation aims at yielding an example how travel costs behave for the combination of sequential grouping and iterative voting protocol. We run a simulation series with 1 run, 41 agents, 20 POIs, group capacity of 20, committee size of 5, a preference dissatisfaction threshold of 0.7 and Foursquare-based preferences. In Table 7.7, we present the resulting committees for the different groups, the corresponding OSM nodes and the travel cost for each committee. For this example, the median individual travel cost is 443.40.

group id (size)	committee	corresponding OSM nodes	aggregated cost
0 (12)	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
1 (4)	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
2 (3)	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
3 (3)	{0, 1, 2, 4, 8}	[6018, 6895, 6115, 2702, 2145]	1,330.14
4 (2)	{4, 8, 9, 10, 11}	[2702, 2145, 6963, 3290, 5276]	1,083.44
5 (2)	{8, 9, 10, 11, 12}	[2145, 6963, 3290, 5276, 4657]	1,081.37
6 (2)	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
7 (1)	{4, 5, 6, 7, 8}	[2702, 4986,4476, 5461, 2145]	1,598.73
8 (2)	{4, 5, 6, 7, 8}	[2702, 4986, 4476, 5461, 2145]	1,598.73
9 (2)	{4, 8, 9, 10, 11}	[2702, 2145, 6963, 3290, 5276]	1,083.44
10 (1)	{0, 4, 5, 6, 7}	[6018, 2702, 4986, 4476, 5461]	1,595.61
11 (2)	{0, 8, 9, 10, 11}	[6018, 2145, 6963, 3290, 5276]	1,104,21
12 (2)	{8, 16, 17, 18, 19}	[2145, 6361, 3505, 3291, 3321]	1,299.26
13 (1)	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
14 (1)	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
15 (1)	{8, 16, 17, 18, 19}	[2145, 6361, 3505, 3291, 3321]	1,299.26

Table 7.7: Resulting committees for sequential_iterative with OSM nodes and costs

7.4 Example simulation - driving in groups, coordinated_basic

The last simulation aims at aims at yielding an example how travel costs behave for the combination of coordinated grouping and basic voting protocol. We run a simulation series with 1 run, 41 agents, 20 POIs, group capacity of 20, committee size of 5, a join threshold of 8 and Foursquare-based preferences. In Table 7.8, we present the resulting committees for the different groups, the corresponding OSM nodes and the travel cost for each committee. For this example, the median individual travel cost is 157.70.

group id (size)	committee	array	aggregated cost
0 (11)	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
1 (6)	{0, 4, 5, 6, 7}	[6018, 2702, 4986, 4476, 5461]	1,595.61
2 (5)	{8, 16, 17, 18, 19}	[2145, 6361, 3505, 3291, 3321]	1,299.26
3 (9)	{8, 12, 13, 14, 15}	[2145, 4657, 6119, 6157, 4729]	1,020.96
4 (7)	{4, 8, 9, 10, 11}	[6018, 2145, 6963, 3290, 5276]	1,104,21
5 (2)	{0, 1, 2, 3, 4}	[6018, 6895, 6115, 5559, 2702]	1,351.31
6 (1)	{8, 16, 17, 18, 19}	[2145, 6361, 3505, 3291, 3321]	1,299.26

Table 7.8: Resulting committees for coordinated_basic with OSM nodes and costs

7.5 Summary for travel costs under different combinations

In this section, we wanted to demonstrate how the inherent differences between the combinations result in differences regarding travel costs. In the main series we got the result that both sequential_iterative and coordinated_basic yield smaller groups than sequential_basic, and coordinated_basic tends to yield larger groups than sequential_iterative. Consistently with this, we can rank the combinations according to the median travel costs in the example simulations as follows: sequential_basic yields lower travel costs than coordinated_basic, and coordinated_basic yields lower travel costs than sequential_iterative. The travel costs when driving alone are higher than for all those combinations. Note that, according to the results from the main simulation series, the preference dissatisfaction for sequential_basic is higher than for coordinated_basic and sequential_iterative, i.e. if you look for a compromise between preference satisfaction and travel costs, one would use coordinated_basic.

Because our aim in this section was to only briefly demonstrate the differences regarding travel costs, we stick to four example simulations. In further work, one could consider simulations with different dissatisfaction and join thresholds to create a more detailed picture.

Chapter 8

Extended model with distance costs

We did not include travel costs in the original model as described in Chapter 4. In this chapter, we consider the incorporation of distance costs (based on the distances between the POIs) into an agent's utility function as a promising extension of our model to make it more realistic. We demonstrate several possibilities for the extension and their effects qualitatively using several example scenarios. Note that this extension has not been implemented in our simulation environment. In the following sections, we describe the components of the model, explain the different possibilities of computing the distance costs, outline an incremental approach and briefly summarise the results of this chapter.

8.1 Model components and considered POIs

In order to take distance costs into account for computing the travellers' preferences, we interpret the old preference function as "attractiveness" function and combine it with a cost function in order to create the new preference/utility function. We also consider three different possibilities for defining the cost function, as detailed below.

Attractiveness values For each traveller t_i and POIs p_j we define an attractiveness value a_{ij} which corresponds to the preference $pref_{ij}$ of traveller t_i for p_j as defined in Chapter 4.

Cost/distance values For each POI p_i, we define an inverse cost/distance value d_i. We present three possibilities for computing the cost values.

- Distance to centroid: Here, we compute for each POI the distance to the centroid.
- Average distance: Here, we compute for each POI the average distance to all other POIs.
- Distance to start/end: Here, we use as cost value for each POI the distance to the start/end point.

Overall function We combine the attractiveness values and inverse cost values by adding them, using weights. The utility or preference value u_{ij} of POI p_j for traveller t_i is defined as

$$u_{ij} = \alpha_i * a_{ij} + (1 - \alpha_i) * d_j.$$

Note that the higher the inverse cost value d_j is, the higher the preference for POI p_j is. You can use different α-values for different agents to model that they have more or less money: Consider two agents a_1 and a_2. Agent a_1 has more money, with budgetary weight $\alpha_1 = 0.8$. Agent a_2 has less money, with budgetary weight $\alpha_2 = 0.2$. The rationale behind using budgetary weights is that the lower the budget of an agent is, the higher the influence of the cost value will be on its preference. The higher the budget of an agent is, the higher the influence of the attractiveness value will be on its preference: Agents with high budget will not pay so much attention to distance costs as agents with lower budget.

Considered POIs For the following examples, we consider some concrete POIs from Manhattan, see Table 8.1. We list the OSM ids in the overview table so that the routing examples below will be comprehensible. We will conduct tie-breaking based on the abbreviations, not based on the full names.

OSM node	POI name	Abbreviation
5559	Bryant Park	BP
4476	Chelsea Market	CM
4986	Central Park	CP
6119	One World Observatory	OW
7090	Grand Central Terminal (start/end)	GC

Table 8.1: Considered POIs

In Section 8.2, we give examples for computing the distance costs, including voting examples for the third possibility. In Section 8.3, we explain how to use an incremental voting process, as opposed to using committee elections.

8.2 Computation of distance costs

In the following subsections, we explain the three possibilities for computing the distance costs, including examples. For the third possibility, we also describe two example committee elections based on the computed distance costs.

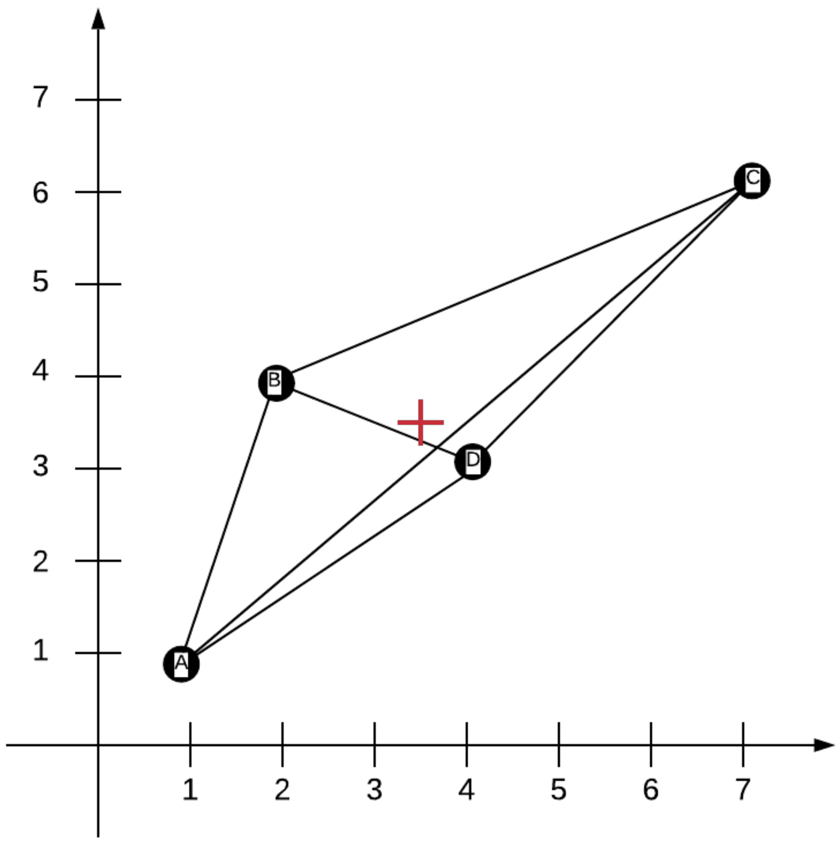

Figure 8.1: Example for computation of distance to centroid

8.2.1 Distance to centroid

A possibility for determining the cost values is to compute the distance of each POI to the centroid over all POIs. For a simple polygon, the centroid is computed by summing up the coordinates, and dividing by number of POIs. Then, we take for each POI the distance to the centroid and normalise this value using the maximum distance between the centroid and any POI. This approach is more suitable for networks with homogenously distributed nodes. Consider as an example the graph depicted in Figure 8.1. We have the following nodes and coordinates:

A with $x_A = 1$, $y_A = 1$

B with $x_B = 2$, $y_B = 4$

C with $x_C = 7$, $y_C = 6$

D with $x_D = 4$, $y_D = 3$

The coordinates $(\overline{x}, \overline{y})$ for the centroid X (marked in red) are computed as

$$\overline{x} = \frac{1+2+7+4}{4} = 3.5$$

and

$$\overline{y} = \frac{1+4+6+3}{4} = 3.5$$

The distances between the nodes and the centroid are

$dist(A, X) = \sqrt{(3.5-1)^2 + (3.5-1)^2} = \sqrt{12.5} = 3.54$

$dist(B, X) = \sqrt{(3.5-2)^2 + (3.5-4)^2} = \sqrt{2.5} = 1.58$

$dist(C, X) = \sqrt{(3.5-7)^2 + (3.5-6)^2} = \sqrt{18.5} = 4.30$

$dist(D, X) = \sqrt{(3.5-4)^2 + (3.5-3)^2} = \sqrt{0.5} = 0.71$

The maximum distance between any node and the centroid is $dist(C, X) = 4.30$. This leads to the normalised distance values

$d_N(A) = 3.54/4.30 = 0.82$

$d_N(B) = 1.58/4.30 = 0.37$

$d_N(C) = 4.30/4.30 = 1.00$

$d_N(D) = 0.71/4.30 = 0.17$

and the inverse distance values

$d(A) = 1 - 0.82 = 0.18$

$d(B) = 1 - 0.37 = 0.63$

$d(C) = 1 - 1.00 = 0.00$

$d(D) = 1 - 0.17 = 0.83$

For an OSM network, we could compute the centroid, determine the shortest path from each POI to the centroid (or the OSM node nearest to the centroid).

8.2.2 Average distance

Another way to determine the cost/distance values is to compute the average distance of each node to all other nodes. This can easily be computed based on a distance matrix. For networks where the nodes are less homogeneously distributed, this approach is more suitable.

Consider as example the graph as specified in Figure 8.2 and the corresponding distance matrix in Table 8.2.

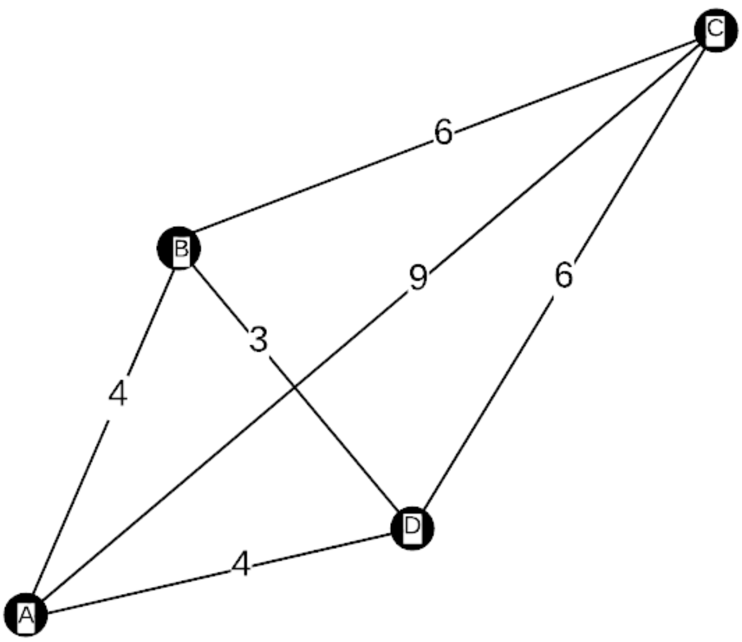

Figure 8.2: Example for computation of average distance

	A	B	C	D
A	–	4	9	4
B	4	–	6	3
C	9	6	–	6
D	4	3	6	–

Table 8.2: Exemplary distance matrix

For this example, we have the average distance values

$\overline{d}(A) = \frac{4+9+4}{3} = 5.67$

$\overline{d}(B) = \frac{4+6+3}{3} = 4.33$

$\overline{d}(C) = \frac{9+6+6}{3} = 7.00$

$\overline{d}(D) = \frac{4+3+6}{3} = 4.33$

We normalise using the maximum average distance $\overline{d}(C) = 7.00$, leading to the normalised cost/distance values

$d_N(A) = 5.67/7.00 = 0.81$

$d_N(B) = 4.33/7.00 = 0.62$

$d_N(C) = 7.00/7.00 = 1.00$

$d_N(D) = 4.33/7.00 = 0.62$

and the inverse distance values

$d(A) = 1 - 0.81 = 0.19$

$d(B) = 1 - 0.62 = 0.38$

$d(C) = 1 - 1.00 = 0.00$

$d(D) = 1 - 0.62 = 0.38$

8.2.2.1 Real-world example

As small real-world example, consider the following 4 POIs in New York City:

- Bryant Park (BP) - OSM id 5559
- Chelsea Market (CM) - OSM id 4476
- Central Park (CP) - OSM id 4986
- One World Observatory (OW) - OSM id 6119

The distances specified in Table 8.3 were computed with pgrouting (Measurements in km).

For example, the distance from Bryant Park to Chelsea Market was computed with the following command.

```
SELECT * FROM pgr_dijkstraCost(
'select osm_id as id, source, target, to_length_cost as cost,
reverse_length_cost as reverse_cost from manhattan_ways',
5559, 4476, true);
```

	BP	CM	CP	OW
BP	–	3.04	4.37	6.75
CM	3.03	–	6.88	4.40
CP	4.31	6.68	–	10.36
OW	5.65	3.81	9.96	–

Table 8.3: Distance matrix for POIs in Manhattan

In the next step, we "correct" the matrix in order to render it symmetrical, see Table 8.4

	BP	CM	CP	OW
BP	–	3.04	4.34	6.20
CM	3.04	–	6.78	4.11
CP	4.34	6.78	–	10.16
OW	6.20	4.11	10.16	–

Table 8.4: Symmetrical distance matrix for POIs in Manhattan

This leads to the average distance values

$\overline{d}(BP) = 4.53$

$\overline{d}(CM) = 4.64$

$\overline{d}(CP) = 7.09$

$\overline{d}(OW) = 6.82$

and the normalised cost/distance values

$d_N(BP) = 4.53/7.09 = 0.64$

$d_N(CM) = 4.64/7.09 = 0.65$

$d_N(CP) = 7.09/7.09 = 1.00$

$d_N(OW) = 6.82/7.09 = 0.96$

as well as the inverse distance values

$d(BP) = 1 - 0.64 = 0.36$

$d(CM) = 1 - 0.65 = 0.35$

$d(CP) = 1 - 1.00 = 0.00$

$d(OW) = 1 - 0.96 = 0.04$

8.2.3 Distances to start/end point

Another possible approach is to compute the distances between the POIs and the start/end point of the tour. In this case, the start/end point is the Grand Central Terminal (GC) with the OSM id 7090. All measurements are given in km. For this approach, we also give an example for an election, using a committee voting rule.

8.2.3.1 Original distances

We compute the distance $d(BP, GC)$ with

```
SELECT * FROM pgr_dijkstraCost(
'select osm_id as id, source, target, to_length_cost as cost,
reverse_length_cost as reverse_cost from manhattan_ways',
5559, 7090, true
);
```

which yields $d(BP, GC) = 0.72$

We compute the reverse distance $d(GC, BP)$ with

```
SELECT * FROM pgr_dijkstraCost(
'select osm_id as id, source, target, to_length_cost as cost,
reverse_length_cost as reverse_cost from manhattan_ways',
7090, 5559, true
);
```

which yields $d(GC, BP) = 1.03$

The average distance is $\overline{d}(BP, GC) = 0.88$.

Analogue, we compute the other distances

$d(CM, GC) = 3.74$, $d(GC, CM) = 3.58$, $\overline{d}(CM, GC) = 3.66$

$d(CP, GC) = 3.92$, $d(GC, CP) = 3.99$, $\overline{d}(CP, CG) = 3.96$

$d(OW, GC) = 6.30$, $d(GC, OW) = 6.93$, $\overline{d}(OW, CG) = 6.62$

8.2.3.2 Normalised distances

Using the maximum distance between the start/end point and any point, we normalise the cost values

$d_N(BP) = \frac{0.88}{6.62} = 0.13$

$d_N(CM) = \frac{3.66}{6.62} = 0.55$

$d_N(CP) = \frac{3.96}{6.62} = 0.60$

$d_N(OW) = \frac{6.62}{6.62} = 1.00$

which yields the inverse cost values

$d(BP) = 0.87$

$d(CM) = 0.45$

$d(CP) = 0.40$

$d(OW) = 0.00$

8.2.3.3 Example election with identical budgetary weights

Let us assume that three agents starting from the POI GC need to agree on two POIs out of the ones specified in Table 8.1. Furthermore, we assume the following attractiveness values of the agents for the POIs. As committee voting rule, we use Minisum-Approval with $k = 2$.

	BP	CM	CP	OW
a_1	0.96	0.93	0.97	0.91
a_2	0.96	0.40	0.97	0.91
a_3	0.10	0.93	0.97	0.99

Table 8.5: Attractiveness values

Then, the preference values of the agents for the POIs are computed as follows (with $\alpha = 0.5$ for all agents):

- agent 1
  - BP: $0.5 * 0.96 + 0.5 * 0.87 = 0.915$
  - CM: $0.5 * 0.93 + 0.5 * 0.45 = 0.69$
  - CP: $0.5 * 0.97 + 0.5 * 0.40 = 0.685$
  - OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.455$
- agent 2
  - BP: $0.5 * 0.96 + 0.5 * 0.87 = 0.915$
  - CM: $0.5 * 0, 40 + 0.5 * 0.45 = 0.425$

  - CP: $0.5 * 0.97 + 0.5 * 0.40 = 0.685$
  - OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.46$

- agent 3
  - BP: $0.5 * 0.10 + 0.5 * 0.87 = 0.485$
  - CM: $0.5 * 0.93 + 0.5 * 0.45 = 0.69$
  - CP: $0.5 * 0.97 + 0.5 * 0.40 = 0.685$
  - OW: $0.5 * 0.99 + 0.5 * 0.00 = 0.495$

This leads to the to the Approval votes as defined in Table 8.6

	BP	CM	CP	OW
v_1	1	1	1	0
v_2	1	0	1	0
v_3	0	1	1	0

Table 8.6: Resulting Approval votes

With lexicographic tie-breaking, the winning committee is $\{BP, CP\}$, and we visit the POIs GC, BP and CP.

We compute the POI sequence via pgrouting with the following command

```
SELECT * FROM pgr_TSP(
$$
SELECT * FROM pgr_dijkstraCostMatrix(
'SELECT osm_id as id, source, target, to_length_cost
as cost, reverse_length_cost as reverse_cost,
x1, y1, x2, y2
FROM manhattan_ways',
ARRAY [5559, 4986, 7090],
directed := true)
$$,
start_id := 7090,
randomize := false
);
```

8.2.3.4 Example election with different budgetary weights

For the second example election, we use the same assumptions as in the first example, except for the budgetary weights: We assume the following weights.

$\alpha_1 = 0.2$ for agent 1

$\alpha_2 = 0.5$ for agent 2

$\alpha_3 = 0.2$ for agent 3

The preference values of the agents for the POIs are computed as follows:

- agent 1
  - BP: $0.2 * 0.96 + 0.8 * 0.87 = 0.888$
  - CM: $0.2 * 0.93 + 0.8 * 0.45 = 0.546$
  - CP: $0.2 * 0.97 + 0.8 * 0.40 = 0.514$
  - OW: $0.2 * 0.91 + 0.8 * 0.00 = 0.182$
- agent 2
  - BP: $0.5 * 0.96 + 0.5 * 0.87 = 0.915$
  - CM: $0.5 * 0,40 + 0.5 * 0.45 = 0.425$
  - CP: $0.5 * 0.97 + 0.5 * 0.40 = 0.685$
  - OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.46$
- agent 3
  - BP: $0.2 * 0.10 + 0.8 * 0.87 = 0.716$
  - CM: $0.2 * 0.93 + 0.8 * 0.45 = 0.546$
  - CP: $0.2 * 0.97 + 0.8 * 0.40 = 0.514$
  - OW: $0.2 * 0.99 + 0.8 * 0.00 = 0.198$

which leads to the Approval votes as defined in Table 8.7

	BP	CM	CP	OW
v_1	1	1	1	0
v_2	1	0	1	0
v_3	1	1	1	0

Table 8.7: Resulting Approval votes

Here, the committee $\{BP, CP\}$ is the unique winning committee, and we visit the POIs GC, BP and CP.

8.3 Incremental approach

In the previous section, the considered voting example was conducted using committee elections like in the original model described in Chapter 4, which means that cost values were computed only once. Another possible approach is, instead of conducting a committee election, conducting a incremental procedure, i.e. using single-winner voting rules and re-computing the cost values in each step. We consider an incremental procedure which we define as follows: In each phase, we use single-winner voting on the set of POIs which have not be chosen so far. For each not yet selected POI, a cost value is computed based on the average distance of the POI to the already chosen POIs. In the first phase of the procedure, the costs are determined via the distances between the start/end point and the POIs as in Section 8.2.3.

8.3.1 Example with identical budgetary weights

As in Subsubsection 8.2.3.3, let us assume that three agents starting from the POI *GC* need to agree on two POIs, with the attractiveness values as given in Table 8.5

If we assume $\alpha = 0.5$ for all agents, the preference values of the agents for the POIs are computed exactly as in the example election in Subsubsection 8.2.3.3.

- agent 1
  - BP: $0.5 * 0.96 + 0.5 * 0.87 = 0.915$
  - CM: $0.5 * 0.93 + 0.5 * 0.45 = 0.69$
  - CP: $0.5 * 0.97 + 0.5 * 0.40 = 0.685$
  - OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.455$
- agent 2
  - BP: $0.5 * 0.96 + 0.5 * 0.87 = 0.915$
  - CM: $0.5 * 0,40 + 0.5 * 0.45 = 0.425$
  - CP: $0.5 * 0.97 + 0.5 * 0.40 = 0.685$
  - OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.46$
- agent 3
  - BP: $0.5 * 0.10 + 0.5 * 0.87 = 0.485$
  - CM: $0.5 * 0.93 + 0.5 * 0.45 = 0.69$

- CP: $0.5 * 0.97 + 0.5 * 0.40 = 0.685$
- OW: $0.5 * 0.99 + 0.5 * 0.00 = 0.495$

This leads to the Approval votes as defined in Table 8.8 (identical to Table 8.6).

	BP	CM	CP	OW
v_1	1	1	1	0
v_2	1	0	1	0
v_3	0	1	1	0

Table 8.8: Resulting Approval votes for the first phase

With an Approval score of 3, CP (Central Park) is the unique winner of this election.

For phase 2, we compute for the remaining POIs BP, CM and OW the average distance to the already chosen POIs GC and CP. To this end, we compute a partial distance matrix as defined in Table 8.9.

	GC	BP	CM	CP	OW
GC	–	1.03	3.58	–	6.93
BP	0.72	–	–	4.37	–
CM	3.74	–	–	6.88	–
CP	–	4.31	6.68	–	10.36
OW	6.30	–	–	9.96	–

Table 8.9: Distance matrix for POIs in Manhattan for incremental procedure

In the next step, we render the distance matrix symmetrical, see Table 8.10

	GC	BP	CM	CP	OW
GC	–	0.88	3.66	–	6.62
BP	0.88	–	–	4.34	–
CM	3.66	–	–	6.78	–
CP	–	4.34	6.78	–	10.16
OW	6.62	–	–	10.16	–

Table 8.10: Symmetrical distance matrix for POIs in Manhattan for incremental procedure

This leads to the average distances

$\overline{d}(BP) = (0.88 + 4.34)/2 = 2.61$

$\overline{d}(CM) = (3.66 + 6.78)/2 = 5.22$

$\overline{d}(OW) = (6.62 + 10.16)/2 = 8.39$

and the normalised distances

$d_N(BP) = 2.61/8.39 = 0.31$

$d_N(CM) = 5.22/8.39 = 0.62$

$d_N(OW) = 8.39/8.39 = 1.00$

resulting in the inverse cost values

$d(BP) = 0.69$

$d(CM) = 0.38$

$d(OW) = 0.00$

The preference values of the agents are computed as follows:

- agent 1
  - BP: $0.5 * 0.96 + 0.5 * 0.69 = 0.825$
  - CM: $0.5 * 0.93 + 0.5 * 0.38 = 0.655$
  - OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.455$
- agent 2
  - BP: $0.5 * 0.96 + 0.5 * 0.69 = 0.825$
  - CM: $0.5 * 0, 40 + 0.5 * 0.38 = 0.39$
  - OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.455$
- agent 3
  - BP: $0.5 * 0.10 + 0.5 * 0.69 = 0.395$
  - CM: $0.5 * 0.93 + 0.5 * 0.38 = 0.655$
  - OW: $0.5 * 0.99 + 0.5 * 0.00 = 0.455$

This leads to the Approval votes as defined in Table 8.11

	BP	CM	OW
v_1	1	1	0
v_2	1	0	0
v_3	0	1	0

Table 8.11: Resulting Approval votes for second phase

Both POI BP and POI CM have Approval score 2. With lexicographic tie-breaking, BP wins the election. Overall, we choose the POIs CP and BP and visit the POIs GC, CP and BP.

8.3.2 Example with different budgetary weights

For the second example, we use the same assumptions as in the previous example, except for the budgetary weights: We assume the following weights.

$\alpha_1 = 0.2$ for agent 1

$\alpha_2 = 0.5$ for agent 2

$\alpha_3 = 0.8$ for agent 3

The preference values of the agents for the POIs are computed as follows:

- agent 1
  - BP: $0.2 * 0.96 + 0.8 * 0.87 = 0.888$
  - CM: $0.2 * 0.93 + 0.8 * 0.45 = 0.546$
  - CP: $0.2 * 0.97 + 0.8 * 0.40 = 0.514$
  - OW: $0.2 * 0.91 + 0.8 * 0.00 = 0.182$
- agent 2
  - BP: $0.5 * 0.96 + 0.5 * 0.87 = 0.915$
  - CM: $0.5 * 0, 40 + 0.5 * 0.45 = 0.425$
  - CP: $0.5 * 0.97 + 0.5 * 0.40 = 0.685$
  - OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.46$
- agent 3

– BP: $0.8 * 0.10 + 0.2 * 0.87 = 0.254$

– CM: $0.8 * 0.93 + 0.2 * 0.45 = 0.834$

– CP: $0.8 * 0.97 + 0.2 * 0.40 = 0.856$

– OW: $0.8 * 0.99 + 0.2 * 0.00 = 0.792$

which leads to the Approval votes as defined in Table 8.12

	BP	CM	CP	OW
v_1	1	1	1	0
v_2	1	0	1	0
v_3	0	1	1	1

Table 8.12: Resulting Approval votes for first phase

Note that CP is again the unique winner here, but that agent 3 approves of POI OW in this election, as opposed to the first example.

Because the winner of phase 1 is the same as in the first example, the cost values for the second phase are the same as in the first example.

For the second phase, the preference values of the agents are computed as follows:

- agent 1
  – BP: $0.2 * 0.96 + 0.8 * 0.69 = 0.744$
  – CM: $0.2 * 0.93 + 0.8 * 0.38 = 0.49$
  – OW: $0.2 * 0.91 + 0.8 * 0.00 = 0.182$
- agent 2
  – BP: $0.5 * 0.96 + 0.5 * 0.69 = 0.825$
  – CM: $0.5 * 0,40 + 0.5 * 0.38 = 0.39$
  – OW: $0.5 * 0.91 + 0.5 * 0.00 = 0.455$
- agent 3
  – BP: $0.8 * 0.10 + 0.2 * 0.69 = 0.218$
  – CM: $0.8 * 0.93 + 0.2 * 0.38 = 0.82$
  – OW: $0.8 * 0.99 + 0.2 * 0.00 = 0.792$

This leads to the Approval votes as defined in Table 8.13

	BP	CM	OW
v_1	1	0	0
v_2	1	0	0
v_3	0	1	1

Table 8.13: Resulting Approval votes for second phase

Note that BP is unique winner in this election as opposed to the second-phase election in the previous example. Overall, we choose again the POIs CP and BP and visit the POIs GC, CP and BP.

8.4 Summary

In this chapter, we described and demonstrated several approaches for incorporating distance costs in the preference generation. An approach which is useful for homogeneous networks is using the distance to the centroid, which was demonstrated in Section 8.2.1. For less homogenous networks, it can be useful to use the average distance of each POI to all other POIs instead, as demonstrated in Section 8.2.2. Both of these approaches can be used if you want to avoid biases which favour POIs close to the start/end point. If you intentionally want to favour POIs close to the start/end point, you can use the distance of each POI to the start/end point instead, as demonstrated in Section 8.2.3. If you want to compute the cost values only once, you can use committee voting rules. If you instead aim at a more fine-grained approach which considers in each step the already determined POIs, you can use an incremental approach using single-winner elections as demonstrated in Section 8.3.

Chapter 9

Application: Agreeing on meeting points

In this chapter, we demonstrate in an excursus the generalisability of the voting approach, by sketching and discussing an alternative scenario and application of voting in cooperative traffic management. This application was considered as part of a joint work co-authored with Paul Czioska, Aleksandar Trifunović and Monika Sester. It has been published in a joint publication, "Location- and time-dependent meeting point recommendations for shared interurban rides" [Czioska et al., 2017]. The contribution of the author of this dissertation has been suggesting voting approaches and discussing how to apply those to the application.

The description of the scenario (Section 9.1), the model (Section 9.2) and the simulation (Section 9.3) closely follows the formulations by [Czioska et al., 2017]. In Section 9.4, we briefly discuss the simulation results regarding voting.

9.1 Scenario

The application considered deals with long-distance ridesharing (using privately owned vehicles), which is a possible way of traversing long distances while sharing travel costs. It is often used for traversing long distances and offers an alternative to trains and intercity bus services, where travel costs can be shared among all passengers. When offering rides between cities, the driver and the riders have to find suitable meeting points. There are straightforward approaches such as choosing common, well-known locations. However, since these locations are mostly part of the inner districts, they can lead to big detours. Thus, another common approach is to develop recommendation systems which take better reachable locations into account. Older approaches mostly focus on intra-urban rides covering shorter distances. The joint paper presents a new recommendation procedure, aiming to extend the meeting-point search by including public transportation to allow the passengers to reach more remote meeting points. In this context, voting was used as part of the meeting-point finding process; it was used as a means for riders and drivers to agree on a meeting point from a set of possible suitable meeting points. [Czioska et al., 2017, p.1-3, p.12]

9.2 Model

Following [Czioska et al., 2017, p.3-8., p.10], we explain the recommendation procedure and in particular the operational phase. The recommendation procedure has three parts, namely 1) preparation phase, 2) precomputing phase and 3) operational phase. The preparation phase serves to process the raw data for preparing the precomputation. It includes several steps, the first one is preparing the driving network. Because, from the passenger perspective, precomputing the travel times from all possible passenger origins is computationally inapplicable, in the second step, representative public transport entry (PTE) nodes are created. It is assumed that the passengers use the public transportation system, so these fictive origins will always be close to public transport stops. In the third step, meeting point candidates are prepared. Subsequently, in the precomputing phase, travel times are precomputed and stored in matrices. The operational phase consists of the following steps, voting being the last one.

1. Estimation of driver arrival times at meeting point candidates
2. Determination of reachable PTE nodes for the passengers
3. Estimation of passenger arrival times at meeting point candidates
4. Computation of total travel times
5. Voting

It is important to recognise that when you consider several persons who need to agree on a meeting point, they will most likely have different preferences regarding the possible meeting points, not only because they have different distances to the possible meeting points, but also because of other properties, for example (subjective) safety at the meeting points, prominence, sheltering possibilities, accessibility, etc. [Czioska et al., 2017, p.12]

In order to aggregate the differing preferences into a socially acceptable agreement, we applied voting. We discussed several possibilities for transforming individual preferences into votes for the election. To keep the model simple, we disregarded aspects such as safety or prominence and based the votes solely on the travel times to the possible meeting points. We considered two voting rules, the first one being a range voting rule, where each voter scores all candidates based on the inverse travel time. The scores are summed up, and the candidate with the highest value wins, i.e. the election result is the meeting point with the lowest total travel time. This is equivalent to optimising the travel times in a utilitarian manner. We also considered a second voting rule which follows the minimax principle. This is equivalent to optimising the travel time according to an egalitarian manner. Here, the winner of the election is the meeting point that minimises the maximum travel time. The idea behind this is to have a more balanced, fair distribution of travel times. [Czioska et al., 2017, p.12f.]

It is also possible to assign an individual meaning of travel time to the votes using individual weights. This enables weighting the votes according to the urgency and can be useful if one or multiple customers are under time pressure and have less time they can spend on the journey. Customers under high time pressure would then be favoured in the voting, with increased probability for their higher ranked meeting points to be recommended. [Czioska et al., 2017, p.13]

9.3 Simulation

In the context of the publication, the recommendation procedure was evaluated with a simulation. In this section, we give an overview of the simulation details and the simulation results regarding voting.

9.3.1 Technical components

For the simulation, the following technical components were used, as [Czioska et al., 2017, p. 19] explained.

- Python 3.5
- OpenTripPlanner
- FreeBSD
- Intel Xeon E5410 with 32 GB RAM

9.3.2 Simulation setting and data

The model for the simulation is based on the medium-sized city of Braunschweig with roughly 250.000 inhabitants. It includes a city centre with a historical core and a pedestrian precinct. The city centre is surrounded by a ring road and areas of high population. The city also includes outer areas with lower population density and some industrial regions. There is an outer street ring consisting of five motorways, with no motorway on the eastern side of the city. Note that the city has a well-developed public transport system including a tram network with 5 tram lines and a bus network with 37 bus lines, with the main public transport lines operating at all times, except for a night break from 2 to 4 am. [Czioska et al., 2017, p. 13]

9.3.3 Travel times

For the computation of travel times for driving, walking and public transport, an instance of OpenTripPlanner[1] was used. OpenTripPlanner is a JAVA-based open-source multi-modal routing engine. OpenStreetMap was used to obtain the necessary data for the street

[1] `http://www.opentripplanner.org/`

network. As for the timetable information and the stop locations, those were obtained in GTFS (General Transit Feed Specification)[2] format from an open data pool. As a result of the preparation phase, 380 PTE nodes were determined. [Czioska et al., 2017, p. 13]

9.3.4 Meeting points and inlet nodes

In order to find meeting point candidates, all petrol stations and the centroids of publicly accessible and free parking places within the considered area were extracted from OpenStreetMap, because these locations offer safe and convenient boarding possibilities. In total, 705 meeting point candidates were extracted. Assuming a uniform distribution of meeting points in the considered area (approximately 200 km^2), there is an average coverage of around 0.28 km^2 per meeting point. Considering that the density of meeting points is very low in non-built-up areas, the meeting point coverage in the city centre is expected to be sufficient for the given purpose. As result after all refinement steps, 94 meeting point candidates were obtained. As inlet nodes, six locations on the motorways surrounding the city were selected manually.[Czioska et al., 2017, p. 14]

9.3.5 Random demand

For the simulation, random demand was generated as follows: The groups for which meeting points were determined consisted of one driver and one to three randomly selected passengers. The driver route was randomly chosen among the six available inlet nodes. To avoid the night break, the time of driver arrival at the inbound inlet node was chosen at random between 6 am and 11 pm. As for the passenger origin locations, those were randomly sampled based on residential building geometries within the service area. The building information necessary for this step was obtained from the municipality of Braunschweig. The probability of a building being chosen depends on its volume, i.e. bigger buildings were chosen more often than smaller buildings. The rationale behind this is that more people are living there, creating a higher demand. In the simulation, a request consists of a driver route and a set of one to three passengers, which are selected at random. [Czioska et al., 2017, p. 14]

9.3.6 Example meeting point recommendation

Figure 9.1, which is directly taken from [Czioska et al., 2017, p.16], shows an example meeting point recommendation involving three passengers. The passengers start in the central districts and use the public transport to reach the recommended meeting point close to the motorway in the north of the city. The driver only has to take a minor detour. [Czioska et al., 2017, p. 15]

[2] `http://gtfs.org/`

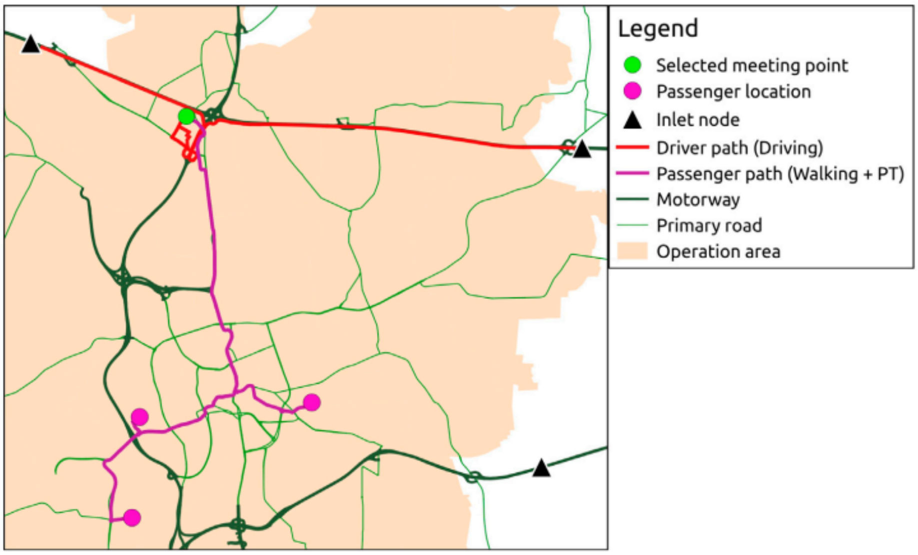

Figure 9.1: Example meeting point selection with three passengers involved

9.3.7 Simulation results regarding voting

In this section, the simulation results regarding voting are described, following the explanations in [Czioska et al., 2017, p.16f., p.19].

The differences between the range voting rule and the minimax voting rule are depicted, based on different group sizes, which range from two (driver and one passenger) to five (driver and four passengers). Table 9.1 is taken from [Czioska et al., 2017, p.19].

Voting of driver and ...	1 passenger	2 passengers	3 passengers	4 passengers
Equal results of both voting rules	59%	31%	30%	29%
Different results: Average lateness using minimax voting	12:29 min	5:32 min	4:30 min	3:57 min
Different results: Average maximum travel time using range voting	19:33 min	34:04 min	38:25 min	41:14 min
Different results: Average maximum travel time using minimax voting	9:20 min	26:16 min	31:11 min	34:37 min

Table 9.1: Range voting vs. minimax voting (10,000 simulation runs)

The first row shows that for a group size of two, the two voting rules recommend the same meeting point in 59% of the cases. This percentage decreases for larger groups: for a group size of five, to a percentage of 29%.

The other rows refer to the case where the voting rules yield different meeting points. The second row specifies the average additional lateness using minimax voting. The difference is fairly large for small groups, but decreases for larger groups; for group size five, it is neglectable.

Row three and four show the average maximum travel time for both voting rules. As can be seen, for group size two there is a large difference. With minimax voting, the average maximum travel time is 9:20 minutes, as opposed to 19:33 minutes for range voting. For larger groups, the difference between range and minimax voting decreases, i.e. the advantage of minimax voting decreases for larger groups.

9.4 Discussion

In this chapter, we demonstrated another useful application of voting for creating consensus in travel groups. Here, we focused on the effects of different ways to measure societal welfare by comparing two simple voting rules which implement the utilitarian perspective on the one hand and the egalitarian perspective on the other hand. For smaller groups, minimax leads to a high improvement regarding the maximum travel time. For larger groups, the difference between range and minimax voting regarding maximum travel time decreases. As for the average lateness using minimax voting, it is high for smaller groups and decreases for larger groups.

This means that, if one aims at fairer distribution of travel times, it makes sense to use minimax voting, but it has to be noted that the advantage of minimax voting decreases for larger groups. If the aim is instead to avoid delay for the group, it would make sense to use the range voting rule, especially for smaller groups.

Chapter 10

Conclusion and Outlook

In this chapter, we summarise the contributions of this dissertation, describe limitations, discuss strengths and weaknesses of the voting approach and give an outlook on possible further work.

10.1 Contribution and limitations

In this thesis, we considered the topic voting for traveller groups in urban areas, motivated by the situation that travellers are requested by traffic management to form groups and need to agree on common destinations to visit. In our approach, we consider committee voting rules as a means for meso-level decisions. We ask how several voting rules, voting protocols and grouping algorithms compare regarding the target quantities group size, preference dissatisfaction and organisational effort.

For our main simulation series as described in Chapter 6, we developed the agent-based simulation tool LightVoting, see Chapter 5. We evaluated several algorithms, which are depicted in Chapter 4 for the situation that travellers form groups in the vicinity of an urban area and need to agree on common destinations to visit. We considered three committee voting rules, Minisum-Approval, Minimax-Approval and Minisum-Ranksum, as well as two grouping algorithms, sequential and coordinated, and two voting protocols, basic and iterative.

Regarding the main simulation series, we arrive at the conclusion that out of the considered algorithms, in order to achieve a compromise between system and user goals, we would recommend the voting rule Minisum-Ranksum in combination with the basic voting protocol and the coordinated grouping algorithm, see Subsection 6.2.6.

We also briefly demonstrated how travel costs differ for different combinations of voting protocols and grouping algorithms when assuming that the routing is conducted via TSP approximations, see Chapter 7. It can be seen that the combination of coordinated grouping and basic protocol yields a good compromise between preference satisfaction and travel costs.

Furthermore, we offered some thoughts in Chapter 8 on how to develop a model which not only takes simple traveller preferences into account but also distance costs. We present three different possibilities for computing the distance costs. Furthermore, we consider two approaches for determining the winning POIs, committee voting rules versus an incremental, more fine-grained approach where the costs are recomputed in each step and the next POI is determined via a single-winner election.

In an excursus, we described a related application where voting is used in ridesharing to agree on meeting points, see Chapter 9. We conclude that, especially for smaller groups, it makes sense to use the minimax voting rule if one aims at fairer distribution of travel times. If one aims at avoiding delay for the group, it would instead make sense to use the range voting rule for smaller groups.

As described in Chapter 3, what is new in comparison to related works on collective decision-making in traffic is that we focus on comparing the effects of several grouping algorithms, voting rules and voting protocols on system and user goals. Other works on collective decision-making in traffic set other points of emphasis, meaning that they for example do not consider group formation, do not compare the effects of several voting algorithms, use other voting algorithms or use other collective decision-making algorithms than voting. The main contributions of this work are:

- We proposed to consider different voting protocols together with committee voting rules as a means of collective decision-making for travellers who visit several POIs together, following [Dennisen and Müller, 2015] and [Dennisen and Müller, 2016].

- We compared several voting rules, voting protocols and grouping algorithms regarding their inherent effects on the user and system quantities group size, preference dissatisfaction and organisational effort.

- We developed an agent-based simulation tool for evaluating different voting rules, voting protocols and grouping algorithms.

- We concluded that out of the considered algorithms, we would recommend the committee voting rule Minisum-Ranksum in combination with the basic voting protocol and the coordinated grouping algorithm in order to achieve a compromise between system and user goals.

- We also briefly demonstrated how travel costs differ for different combinations of voting protocols and grouping algorithms. The combination of the basic voting protocol and the coordinated grouping algorithm yields a good compromise between preference satisfaction and travel costs.

- We presented an extended model for taking distance costs into account when generating the preferences, describing different possibilities for computing the distance costs and determining the winning POIs.

- In an excursus, we compared the effect of a utilitarian and an egalitarian voting rule when agreeing on meeting points for a ride-sharing approach. We conclude that it makes sense to use the egalitarian (minimax) voting rule if one aims at fairer distribution of travel times, especially for smaller groups. If the aim is instead to avoid delay for the group, it makes sense to use the utilitarian (range) voting rule for smaller groups.

Our work has some limitations, one being that we do not consider real-world constraints. When conducting our simulations, we assumed “perfect” communication. Furthermore, we considered a relatively small number of algorithms. We did not conduct simulations for the extended model described in Chapter 8. Our model does not include the role of financial incentives.

10.2 Strengths and weaknesses of the voting approach

The approach to use voting in cooperative traffic management has several strengths. One aspect is that using voting as a decision mechanism can be combined, if so desired, with integrated (anonymised) surveys of user preferences. These surveys can be conducted ad hoc or via user profiles which are created before the concrete journey. Another strength of voting is flexibility of architecture: It can be applied and implemented in both decentralised and centralised manners. A decentralised implementation would enable restricting the sharing of user data. Organising the travellers into groups which decide on common decisions via voting offers advantages for both traffic management and the individual travellers: Fewer, larger groups of travellers are better manageable and the larger the groups, the smaller the travel cost is for the individual travellers.

Insights from Computational Social Choice can be used to design protocols and mechanisms for different requirements. If an application is extremely prone to specific forms of control, it would be sensible to use voting rules which are either immune to these forms of control or resistant to them because the control problems are $\mathcal{NP}$-hard under this voting rule. Similarly, if an application is prone to specific forms of manipulation, it would be advantageous to use voting rules for which these manipulation problems are $\mathcal{NP}$-hard. For example, the Coalitional Weighted Manipulation problem is $\mathcal{NP}$-hard for more than two candidates under the voting rules Borda and Veto (see [Conitzer et al., 2007]), i.e. for these rules it is hard for coalitions of strategic voters to manipulate their votes such that a specific candidate wins.

It should be noted that voting approaches have some weaknesses as well. For example, the transmission of votes depends on reliable communication. If the communication is not reliable, there can be stability problems. Every voting approach will be prone to some forms of attacks such as strategic voting, control attacks, false identities, denial-of-service attacks and others. There are also security issues such as data theft and eavesdropping. All of those weaknesses need to be considered, weighed and addressed in concrete implementations for real-world applications.

10.3 Outlook

There are several possible approaches for future work. In the following, we describe three future research opportunities and how LightVoting could be used for further research.

10.3.1 Considering real-world constraints and requirements

In this dissertation, real-world constraints like communication-technological challenges were not considered. For future investigations focused on behaviour under real-world conditions, it could be sensible to adapt the simulation concept to consider constraints as presented in [Teixeira et al., 2018, Teixeira et al., 2019]. One main aim in their works was to investigate the suitability of voting rules as collective decision-making mechanism in platoon applications considering unreliable communication. One challenge for further works could be to follow the underlying concept of [Teixeira et al., 2018, Teixeira et al., 2019] for taking real-world constraints into account while setting other points of emphasis, such as considering other target quantities and considering several grouping algorithms. Another aspect for real-world applications is that user data need to be elicited and processed in a privacy-conform way. Here, decentralised design and encryption techniques could be helpful. The voting rules should also be chosen carefully so that in applications where specific forms of control or manipulation are likely, voting rules which are inherently protected against those forms are preferred.

10.3.2 Considering additional algorithms and models

As mentioned above, only a limited number of algorithms was considered. For future works, one could consider additional algorithms and models, for example other models for preference generation, more voting rules, more refined iterative voting protocols or more refined grouping algorithms. In the real world, travellers might not have preferences over all possible POIs. For real-world investigations, it could be helpful to generate the preferences based on surveys. This dissertation focused on committee elections. For future work, it might be useful to also consider the more general case of combinatorial voting, see for example [Lang and Xia, 2016]. It might be interesting to apply alternative solution approaches for voting in combinatorial domains to the urban visitors scenario. It could also be useful to consider single-winner voting rules, for example in the context of the incremental approach as described in Chapter 8. An interesting extension might be to run simulations for the extended model including distance costs as proposed in Chapter 8. In particular, it would be interesting to compare the performance of the approach based on pre-computed cost values for all POIs with the performance of the incremental approach where in each phase, the cost values for the not yet selected POIs are re-computed.

10.3.3 Considering financial incentives

Another possible extension would be to consider the role of financial incentives to encourage ridesharing or platooning. This was for example considered by [Storch et al., 2020]. In this paper, the authors arrive at the conclusion that financial incentives have the potential of significantly increasing the dissemination of ridesharing. In this vein, it would make sense to incorporate different models for financial incentives into the voting-based model presented in this dissertation and to investigate their effect. For example, [Sebe et al., 2020] propose the notion of compensational platooning using automated negotiation between agents representing vehicles. Vehicles can offer monetary compensation to nearby vehicles to incentivise them to join together for a part of the route in the future. One challenge for future research could be to combine the negotiation-based way of decentralised platoon building as described in [Sebe et al., 2020] with the voting-based approach of collective decision-making.

10.3.4 Further applications for LightVoting in research

In the following, we discuss how the simulation tool LightVoting as decribed in Chapter 5 could be used for application in research. We consider three use cases as depicted in Figure 10.1.

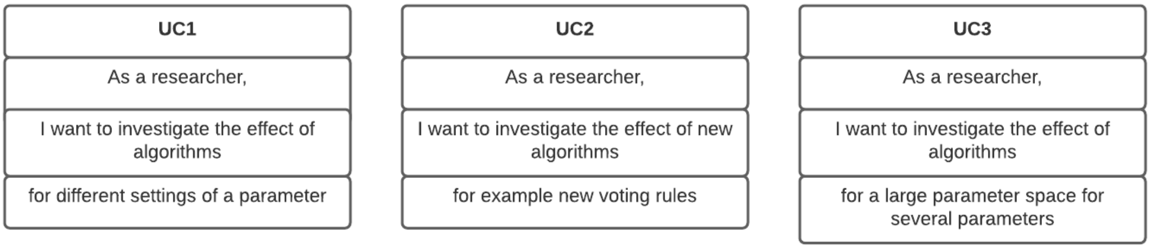

Figure 10.1: Use Cases 1, 2 and 3

UC1 aims at considering different settings for a parameter, for example dissatisfaction threshold.

UC2 aims at considering new algorithms, for example a new voting rule.

UC3 aims at considering different algorithms with different settings for several parameters, similar to the main series in Chapter 6, but with a larger parameter space.

In **UC1**, one would consider different settings for a parameter for the already defined algorithms to explore the space of this parameter more thoroughly.

One possibility for **UC2** is exploring the effects of the extended model described in Chapter 8. Another possibility is investigating the effects of additional voting rules or refined versions of the iterative protocol or the coordinated grouping.

In the discussion of the results of the main simulation series (Subsection 6.2.6), we described an example for **UC3**. According to the hypotheses generated based on the main simulation series, it would make sense for further research to explore a larger parameter space for Minisum-Ranksum under the combination coordinated grouping and basic protocol with different numbers of travellers, different numbers of alternatives, different group capacities and different join thresholds.

Technically, the application of LightVoting for further research could be realised as follows, see also Figure 10.2: We independently simulate different settings using LightVoting and merging the results via a PostgreSQL database server. In the last step, the results can be visualised.

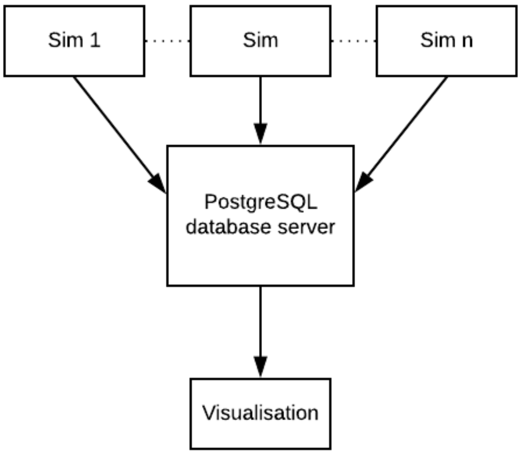

Figure 10.2: Technical realisation

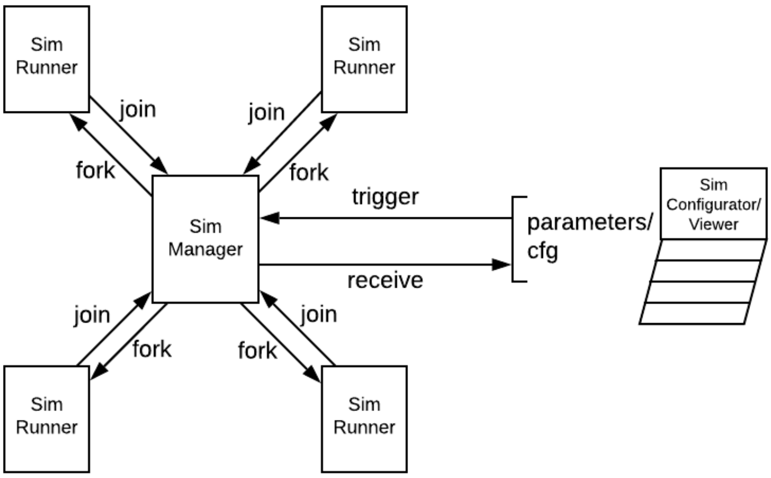

Figure 10.3: Concept view

In the following, we describe the concept view, see also Figure 10.3. Consider a traffic researcher who wants to compare the simulation results for several different configurations. This could be simulations with different settings for one parameter (**UC1**), simulations for exploring the effects of new algorithms (**UC2**) or for exploring a large parameter space for several parameters (**UC3**). The researcher, who acts as simulation configurator, stores the configurations in the database. The simulation manager agent translates the configurations into tasks. There are several simulation runner agents who regularly check the database for still open tasks. The tasks can be independently processed by the simulation runners, which is advantageous, especially for **UC3**, when exploring a large parameter space for several parameters. When a simulation runner is done with a task, it marks the task as completed. The simulation manager regularly checks the database to revise if all tasks are completed. If the tasks are completed, the simulation manager sends the results back to the traffic researcher who then can visualise the results. Note that in the case of **UC2**, the researcher needs an extended version of the code for implementing further algorithms, which can be realised by creating an own branch of the simulation code and specifying the branch name in the configuration. The simulation runner conducting the task then can pull the specified branch to execute the code.

Bibliography

[Agatz et al., 2012] Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. European Journal of Operational Research, 223(2):295–303.

[Airiau et al., 2017] Airiau, S., Grandi, U., and Perotto, F. S. (2017). Learning Agents for Iterative Voting. In International Conference on Algorithmic DecisionTheory, pages 139–152. Springer.

[Al Alam et al., 2010] Al Alam, A., Gattami, A., and Johansson, K. H. (2010). An experimental study on the fuel reduction potential of heavy duty vehicle platooning. In Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, pages 306–311. IEEE.

[Alam et al., 2015] Alam, A., Besselink, B., Turri, V., Martensson, J., and Johansson, K. H. (2015). Heavy-Duty Vehicle Platooning for Sustainable Freight Transportation: A Cooperative Method to Enhance Safety and Efficiency. IEEE Control Systems, 35(6):34–56.

[Alcalde and Revilla, 2004] Alcalde, J. and Revilla, P. (2004). Researching with whom? Stability and manipulation. Journal of Mathematical Economics, 40(8):869–887.

[Alkhateeb et al., 2010] Alkhateeb, F., Al Maghayreh, E., and Aljawarneh, S. (2010). A Multi Agent-Based System for Securing University Campus: Design and Architecture. In 2010 International Conference on Intelligent Systems, Modelling and Simulation, pages 75–79. IEEE.

[Arrow, 1963] Arrow, K. J. (1951, revised 1963). Social Choice and Individual Values. John Wiley and Sons.

[Aschermann et al., 2017] Aschermann, M., Friedrich, B., and Müller, J. P. (2017). Towards Fair and Efficient Traffic Flow Coordination Mechanisms for 2+ 1 Roadways. Transportation Research Procedia, 27:436–443.

[Aschermann et al., 2016] Aschermann, M., Kraus, P., and Müller, J. P. (2016). LightJason: A BDI Framework Inspired by Jason. Technical Report IfI Technical Report IfI-16-04, Department of Computer Science, TU Clausthal, Clausthal-Zellerfeld, Germany.

[Asplund et al., 2017] Asplund, M., Lövhall, J., and Villani, E. (2017). Specification, Implementation and Verification of Dynamic Group Membership for Vehicle Coordination.

In 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC), pages 321–328. IEEE.

[Aziz et al., 2019] Aziz, H., Brandl, F., Brandt, F., Harrenstein, P., Olsen, M., and Peters, D. (2019). Fractional Hedonic Games. ACM Transactions on Economics and Computation (TEAC), 7(2):6.

[Aziz et al., 2016] Aziz, H., Savani, R., and Moulin, H. (2016). Hedonic Games. In Handbook of Computational Social Choice, pages 356–376. Cambridge University Press.

[Banerjee et al., 2001] Banerjee, S., Konishi, H., and Sönmez, T. (2001). Core in a simple coalition formation game. Social Choice and Welfare, 18(1):135–153.

[Bartholdi et al., 1989a] Bartholdi, J., Tovey, C. A., and Trick, M. A. (1989a). Voting schemes for which it can be difficult to tell who won the election. Social Choice and welfare, 6(2):157–165.

[Bartholdi et al., 1989b] Bartholdi, J. J., Tovey, C. A., and Trick, M. A. (1989b). The computational difficulty of manipulating an election. Social Choice and Welfare, 6(3):227–241.

[Baumeister and Dennisen, 2015] Baumeister, D. and Dennisen, S. (2015). Voter Dissatisfaction in Committee Elections. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pages 1707–1708.

[Bazzan and Klügl, 2009] Bazzan, A. L. and Klügl, F., editors (2009). Multi-agent Systems for Traffic and Transportation Engineering. IGI Global.

[Beer et al., 1999] Beer, M., d'Inverno, M., Luck, M., Jennings, N., Preist, C., and Schroeder, M. (1999). Negotiation in multi-agent systems. The Knowledge Engineering Review, 14(3):285–289.

[Ben-Akiva et al., 1985] Ben-Akiva, M. E., Lerman, S. R., and Lerman, S. R. (1985). Discrete Choice Analysis: Theory and Application to Travel Demand, volume 9. MIT press.

[Bergenhem et al., 2012] Bergenhem, C., Shladover, S., Coelingh, E., Englund, C., and Tsugawa, S. (2012). Overview of platooning systems. In Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012).

[Beroldo, 1991] Beroldo, S. (1991). Ridematching System Effectiveness: A Coast-to-Coast Perspective. Transportation Research Record, 1321:7–12.

[Bilbao et al., 2002] Bilbao, J. M., Fernandez, J. R., Jiménez, N., and Lopez, J. J. (2002). Voting power in the European Union enlargement. European Journal of Operational Research, 143(1):181–196.

[Bogomolnaia and Jackson, 2002] Bogomolnaia, A. and Jackson, M. O. (2002). The Stability of Hedonic Coalition Structures. Games and Economic Behavior, 38(2):201–230.

[Bogomolnaia et al., 2005] Bogomolnaia, A., Moulin, H., and Stong, R. (2005). Collective choice under dichotomous preferences. Journal of Economic Theory, 122(2):165–184.

[Bomey, 2021] Bomey, N. (2021). Volkswagen's long-awaited revival of microbus goes autonomous: ID.Buzz electric van on its way. USA Today. `https://eu.usatoday.com/story/money/cars/2021/03/01/volkswagen-microbus-vw-id-buzz-electric-van-self-driving/6866703002/`. Published 01 March 2021.

[Bonnet and Fritz, 2000] Bonnet, C. and Fritz, H. (2000). Fuel Consumption Reduction in a Platoon: Experimental Results with two Electronically Coupled Trucks at Close Spacing. Technical report, SAE Technical Paper.

[Borda, 1781] Borda, J.-C. d. (1781). Mémoire sur les élections au scrutin. Histoire de l'Académie Royale des Sciences, Paris.

[Bordini and Hübner, 2005] Bordini, R. H. and Hübner, J. F. (2005). BDI agent programming in AgentSpeak using Jason. In International Workshop on Computational Logic in Multi-Agent Systems, pages 143–164. Springer.

[Bordini and Hübner, 2009] Bordini, R. H. and Hübner, J. F. (2009). Agent-based simulation using BDI programming in Jason. Multi-agent systems: simulation and applications, pages 451–471.

[Bordini et al., 2005] Bordini, R. H., Hübner, J. F., and Vieira, R. (2005). Jason and the Golden Fleece of agent-oriented programming. In Multi-agent programming, pages 3–37. Springer.

[Brams et al., 2004] Brams, S., Kilgour, D. M., and Sanver, M. R. (2004). A Minimax Procedure for Negotiating Multilateral Treaties. In Reasoned Choices: Essays in Honor of Hannu Nurmi. Finnish Political Science Association.

[Brams and Fishburn, 1978] Brams, S. J. and Fishburn, P. C. (1978). Approval Voting. The American Political Science Review, pages 831–847.

[Brandt et al., 2016a] Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. (2016a). Introduction to Computational Social Choice. In Handbook of Computational Social Choice, pages 1–20. Cambridge University Press.

[Brandt et al., 2016b] Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D., editors (2016b). Handbook of Computational Social Choice. Cambridge University Press.

[Bratman, 1987] Bratman, M. (1987). Intention, Plans, and Practical Reason, volume 10. Harvard University Press Cambridge, MA.

[Bratman et al., 1988] Bratman, M. E., Israel, D. J., and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. Computational Intelligence, 4(3):349–355.

[Cambridge Dictionary, 2021] Cambridge Dictionary (2021). PLATOONING | definition in the Cambridge English Dictionary. https://dictionary.cambridge.org/us/dictionary/english/platooning. Retrieved 19 March 2021.

[Carbaugh et al., 1998] Carbaugh, J., Godbole, D. N., and Sengupta, R. (1998). Safety and capacity analysis of automated and manual highway systems. Transportation Research Part C: Emerging Technologies, 6(1-2):69–99.

[Carley, 1999] Carley, K. M. (1999). On generating hypotheses using computer simulations. Systems Engineering: The Journal of The International Council on Systems Engineering, 2(2):69–77.

[Chalkiadakis and Wooldridge, 2016] Chalkiadakis, G. and Wooldridge, M. (2016). Weighted Voting Games. In Handbook of Computational Social Choice, pages 377–395. Cambridge University Press.

[Condorcet, 1785] Condorcet, J. A. (1785). Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. Imprimerie Royale.

[Conitzer et al., 2007] Conitzer, V., Sandholm, T., and Lang, J. (2007). When are elections with few candidates hard to manipulate? Journal of the ACM (JACM), 54(3):14–es.

[Copeland, 1951] Copeland, A. (1951). A "reasonable" social welfare function. Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences. University of Michigan.

[Czioska et al., 2017] Czioska, P., Trifunović, A., Dennisen, S., and Sester, M. (2017). Location- and time-dependent meeting point recommendations for shared interurban rides. Journal of Location Based Services, 11(3-4):181–203. https://doi.org/10.1080/17489725.2017.1421779.

[Darmann, 2015] Darmann, A. (2015). Group activity selection from ordinal preferences. In International Conference on Algorithmic Decision Theory, pages 35–51. Springer.

[Darmann et al., 2012] Darmann, A., Elkind, E., Kurz, S., Lang, J., Schauer, J., and Woeginger, G. (2012). Group Activity Selection Problem. In International Workshop on Internet and Network Economics, pages 156–169. Springer.

[Darmann and Lang, 2016] Darmann, A. and Lang, J. (2016). Group Activity Selection Problems. In Trends in Computational Social Choice, pages 87–101. AI Access.

[de Abreu Batista and Coutinho, 2013] de Abreu Batista, A. and Coutinho, L. R. (2013). A multiagent system for combining green wave and adaptive control in a dynamic way. In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pages 2439–2444. IEEE.

[Dennisen, 2021a] Dennisen, S. (2021a). LightVoting database containing the simulation results of my Ph.D. thesis. https://doi.org/10.5281/zenodo.4663329.

[Dennisen, 2021b] Dennisen, S. (2021b). LightVoting simulation tool. https://doi.org/10.5281/zenodo.5516232.

[Dennisen and Müller, 2015] Dennisen, S. L. and Müller, J. P. (2015). Agent-Based Voting Architecture for Traffic Applications. In German Conference on Multiagent System Technologies, pages 200–217. Springer.

[Dennisen and Müller, 2016] Dennisen, S. L. and Müller, J. P. (2016). Iterative Committee Elections for Collective Decision-Making in a Ride-Sharing Application. In ATT@ IJCAI.

[Dodgson, 1876] Dodgson, C. (1876). A method of taking votes on more than two issues. Clarendon Press.

[Dresner and Stone, 2008] Dresner, K. and Stone, P. (2008). A Multiagent Approach to Autonomous Intersection Management. Journal of Artificial Intelligence Research, 31:591–656.

[Dreze and Greenberg, 1980] Dreze, J. H. and Greenberg, J. (1980). Hedonic Coalitions: Optimality and Stability. Econometrica: Journal of the Econometric Society, pages 987–1003.

[DroidDrive GmbH, 2021a] DroidDrive GmbH (2021a). About - Ducktrain. https://ducktrain.io/about/. Retrieved 19 March 2021.

[DroidDrive GmbH, 2021b] DroidDrive GmbH (2021b). Ducktrain_bicyclelane2.png. https://ducktrain.io/wp-content/uploads/2020/07/Ducktrain_BicycleLane2.png. Retrieved 19 March 2021.

[Dubernet and Axhausen, 2012] Dubernet, T. and Axhausen, K. W. (2012). Including joint trips in a multi-agent transport simulation. In Proceedings of the 13th International Conference on Travel Behaviour Research, Toronto, July 2012.

[Dubernet and Axhausen, 2013] Dubernet, T. and Axhausen, K. W. (2013). A Framework to Represent Joint Decisions in a Multi-Agent Transport Simulation. In 13th Swiss Transport Research Conference, Ascona.

[Endriss, 2017] Endriss, U., editor (2017). Trends in Computational Social Choice. AI Access.

[EU, 2010] EU (2010). DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 7 July 2010. Official Journal of the European Union. https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:207:0001:0013:EN:PDF.

[Faliszewski et al., 2008] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., and Rothe, J. (2008). Copeland Voting Fully Resists Constructive Control. In International Conference on Algorithmic Applications in Management, pages 165–176. Springer.

[Faliszewski et al., 2009] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., and Rothe, J. (2009). Llull and Copeland Voting Computationally Resist Bribery and Constructive Control. Journal of Artificial Intelligence Research, 35:275–341.

[Ferguson, 1997] Ferguson, E. (1997). The rise and fall of the American carpool: 1970–1990. Transportation, 24(4):349–376.

[Ferreira and d'Orey, 2011] Ferreira, M. and d'Orey, P. M. (2011). On the Impact of Virtual Traffic Lights on Carbon Emissions Mitigation. IEEE Transactions on Intelligent Transportation Systems, 13(1):284–295.

[Furuhata et al., 2013] Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. Transportation Research Part B: Methodological, 57:28–46.

[Gale and Shapley, 1962] Gale, D. and Shapley, L. S. (1962). College admissions and the stability of marriage. The American Mathematical Monthly, 69(1):9–15.

[Georgeff et al., 1998] Georgeff, M., Pell, B., Pollack, M., Tambe, M., and Wooldridge, M. (1998). The Belief-Desire-Intention Model of Agency. In International Workshop on Agent Theories, Architectures, and Languages, pages 1–10. Springer.

[Georgeff and Lansky, 1987] Georgeff, M. P. and Lansky, A. L. (1987). Reactive reasoning and planning. In AAAI, volume 87, pages 677–682.

[Görmer and Müller, 2012] Görmer, J. and Müller, J. P. (2012). Multiagent system architecture and method for group-oriented traffic coordination. In 2012 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST), pages 1–6. IEEE.

[Grimaldo et al., 2008] Grimaldo, F., Lozano, M., and Barber, F. (2008). MADeM: a multi-modal decision making for social MAS. In Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 1, pages 183–190. International Foundation for Autonomous Agents and Multiagent Systems.

[Grimaldo et al., 2010] Grimaldo, F., Lozano, M., Barber, F., and Guerra-Hernández, A. (2010). J-MADeM v. 1.1: A full-fledge AgentSpeak (L) multimodal social decision library in Jason. In The 8th European Workshop on Multi-Agent Systems (EUMAS 2010).

[Grimaldo et al., 2012] Grimaldo, F., Lozano, M., Barber, F., and Guerra-Hernández, A. (2012). Towards a model for urban mobility social simulation. Progress in Artificial Intelligence, 1(2):149–156.

[Haas and Friedrich, 2017] Haas, I. and Friedrich, B. (2017). Developing a micro-simulation tool for autonomous connected vehicle platoons used in city logistics. Transportation Research Procedia, 27:1203–1210.

[Hamming, 1950] Hamming, R. W. (1950). Error detecting and error correcting codes. Bell System Technical Journal, 29(2):147–160.

[Igarashi et al., 2017a] Igarashi, A., Bredereck, R., and Elkind, E. (2017a). On Parameterized Complexity of Group Activity Selection Problems on Social Networks. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, pages 1575–1577. International Foundation for Autonomous Agents and Multiagent Systems.

[Igarashi et al., 2017b] Igarashi, A., Peters, D., and Elkind, E. (2017b). Group Activity Selection on Social Networks. In AAAI, pages 565–571.

[Janssen et al., 2015] Janssen, R., Zwijnenberg, H., Blankers, I., and de Kruijff, J. (2015). Truck platooning: Driving the future of transportation. `http://resolver.tudelft.nl/uuid:778397eb-59d3-4d23-9185-511385b91509`.

[Jennings, 2000] Jennings, N. R. (2000). On agent-based software engineering. Artificial Intelligence, 117(2):277–296.

[Kelley, 2007] Kelley, K. L. (2007). Casual Carpooling-Enhanced. Journal of Public Transportation, 10(4):6.

[Klaus et al., 2016] Klaus, B., Manlove, D. F., and Rossi, F. (2016). Matching under Preferences. In Handbook of Computational Social Choice, pages 333–355. Cambridge University Press.

[Kokkinogenis et al., 2019] Kokkinogenis, Z., Teixeira, M., d'Orey, P. M., and Rossetti, R. J. (2019). Tactical Level Decision making for Platoons of Autonomous Vehicle using Auction Mechanisms. In 2019 IEEE Intelligent Vehicles Symposium (IV), pages 1632–1638. IEEE.

[Lamport, 1998] Lamport, L. (1998). The Part-Time Parliament. ACM Transactions on Computer Systems (TOCS), 16(2):133–169.

[Lang and Xia, 2016] Lang, J. and Xia, L. (2016). Voting in Combinatorial Domains.

[Le Breton et al., 2008] Le Breton, M., Ortuño-Ortin, I., and Weber, S. (2008). Gamson's law and hedonic games. Social Choice and Welfare, 30(1):57–67.

[Lee and Shoham, 2014] Lee, H. and Shoham, Y. (2014). Stable Group Scheduling. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, pages 1347–1348.

[Lee and Shoham, 2015] Lee, H. and Shoham, Y. (2015). Stable invitations. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 29, no.1.

[Liang et al., 2013] Liang, K.-Y., Mårtensson, J., and Johansson, K. H. (2013). When is it Fuel Efficient for a Heavy Duty Vehicle to Catch Up With a Platoon? IFAC Proceedings Volumes, 46(21):738–743.

[Liang et al., 2016] Liang, K.-Y., van de Hoef, S., Terelius, H., Turri, V., Besselink, B., Mårtensson, J., and Johansson, K. H. (2016). Networked control challenges in collaborative road freight transport. European Journal of Control, 30:2–14.

[Liao, 2017] Liao, F. (2017). Joint travel problem in space–time multi-state supernetworks. Transportation, pages 1–25.

[LightJason, 2019a] LightJason (2019a). LightJason. `https://lightjason.org/`. Retrieved 05 March 2021.

[LightJason, 2019b] LightJason (2019b). Theoretical Introduction: Difference to Jason. `https://lightjason.org/knowledgebase/differencetojason/`. Retrieved 05 March 2021.

[LightJason, 2019c] LightJason (2019c). Theoretical Introduction: From Finite-State Machine and Logical Programming to an Agent. `https://lightjason.org/knowledgebase/agent/`. Retrieved 05 March 2021.

[Mårtensson et al., 2012] Mårtensson, J., Alam, A., Behere, S., Khan, M. A. A., Kjellberg, J., Liang, K.-Y., Pettersson, H., and Sundman, D. (2012). The Development of a Cooperative Heavy-Duty Vehicle for the GCDC 2011: Team Scoop. IEEE Transactions on Intelligent Transportation Systems, 13(3):1033–1049.

[Micro Mobility Systems AG, 2018] Micro Mobility Systems AG (2018). microlino-preseries-red-blue.jpg. `https://www.micro-mobility.com/resources/public/lava3/media/microlino-preseries-red-blue.jpg`. Via `https://www.micro-mobility.com/de/news/microlino-vorserie`. Published 05 February 2018.

[Najafian et al., 2020] Najafian, S., Herzog, D., Qiu, S., Inel, O., and Tintarev, N. (2020). You Do Not Decide for Me! Evaluating Explainable Group Aggregation Strategies for Tourism. In Proceedings of the 31st ACM Conference on Hypertext and Social Media, pages 187–196.

[Nelson, 2015] Nelson, G. (2015). Tesla beams down 'autopilot' mode to Model S. Automotive News. `https://www.autonews.com/article/20151014/OEM06/151019938/tesla-beams-down-autopilot-mode-to-model-s`. Published 14 October 2015.

[Next, 2020a] Next (2020a). Home | NextFutureTransport. `https://www.next-future-mobility.com/`. Retrieved 19 March 2021.

[Next, 2020b] Next (2020b). Home | NextFutureTransport (Demovideo). `https://www.next-future-mobility.com/`. Screenshot from first demo video. Retrieved 19 March 2021.

[Ploeg et al., 2011] Ploeg, J., Scheepers, B. T., Van Nunen, E., Van de Wouw, N., and Nijmeijer, H. (2011). Design and experimental evaluation of cooperative adaptive cruise control. In Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, pages 260–265. IEEE.

[Poczter and Jankovic, 2014] Poczter, S. L. and Jankovic, L. M. (2014). The Google Car: Driving Toward A Better Future? Journal of Business Case Studies (Online), 10(1):7.

[Ramakers et al., 2011] Ramakers, R., Henning, K., Gies, S., Abel, D., and Haberstroh, M. (2011). Electronically coupled truck platoons on German highways. In Automation, Communication and Cybernetics in Science and Engineering 2009/2010, pages 441–451. Springer.

[Rossetti et al., 2000] Rossetti, R. J., Bampi, S., Liu, R., Van Vliet, D., and Cybis, H. (2000). An agent-based framework for the assessment of drivers' decision-making. In ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 00TH8493), pages 387–392. IEEE.

[Rothe et al., 2012] Rothe, J., Baumeister, D., Lindner, C., and Rothe, I. (2012). Einführung in Computational Social Choice: Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Springer-Verlag.

[Russell and Norvig, 1995] Russell, S. and Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, 25(27):79–80.

[Saad et al., 2010] Saad, W., Han, Z., Basar, T., Debbah, M., and Hjorungnes, A. (2010). Hedonic Coalition Formation for Distributed Task Allocation among Wireless Agents. IEEE Transactions on Mobile Computing, 10(9):1327–1344.

[Sanderson et al., 2012] Sanderson, D., Busquets, D., and Pitt, J. (2012). A Micro-Meso-Macro Approach to Intelligent Transportation Systems. In Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2012 IEEE Sixth International Conference on, pages 77–82. IEEE.

[Sanderson and Pitt, 2012] Sanderson, D. and Pitt, J. (2012). Institutionalised Consensus in Vehicular Networks: Executable Specification and Empirical Validation. In Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2012 IEEE Sixth International Conference on, pages 71–76. IEEE.

[Sebe et al., 2020] Sebe, S.-M., Baarslag, T., and Müller, J. P. (2020). Automated Negotiation Mechanism and Strategy for Compensational Vehicular Platooning. PRIMA 2020.

[Sierzputowski, 2018] Sierzputowski, K. (2018). An Eight Foot Micro Vehicle Will Soon Make its Way onto European Streets. Colossal. `https://www.thisiscolossal.com/2018/07/microlino-european-streets/`. Published July 2018.

[Singh and Gupta, 2015] Singh, B. and Gupta, A. (2015). Recent trends in intelligent transportation systems: a review. Journal of Transport Literature, 9(2):30–34.

[Singh, 2013] Singh, M. P. (2013). Norms as a basis for governing sociotechnical systems. ACM Transactions on Intelligent Systems and Technology (TIST), 5(1):21.

[Singh et al., 2018] Singh, P. K., Sharma, S., Nandi, S. K., Singh, R., and Nandi, S. (2018). Leader Election in Cooperative Adaptive Cruise Control Based Platooning. In Proceedings of the 1st International Workshop on Communication and Computing in Connected Vehicles and Platooning, pages 8–14.

[Soares et al., 2013] Soares, G., Kokkinogenis, Z., Macedo, J. L., and Rossetti, R. J. (2013). Agent-Based Traffic Simulation Using SUMO and JADE: An Integrated Platform for Artificial Transportation Systems. In Simulation of Urban MObility User Conference, pages 44–61. Springer.

[Stenquist, 2013] Stenquist, P. (2013). On the Road to Autonomous, a Pause at Extrasensory. https://www.nytimes.com/2013/10/27/automobiles/on-the-road-to-autonomous-a-pause-at-extrasensory.html?pagewanted=all. Published 27 October 2013.

[Storch et al., 2020] Storch, D.-M., Timme, M., and Schröder, M. (2020). Incentive-driven discontinuous transition to high ride-sharing adoption. arXiv preprint arXiv:2008.11079.

[Taylor, 1995] Taylor, A. D. (1995). Mathematics and Politics. Springer.

[Taylor, 2005] Taylor, A. D. (2005). Social Choice and the Mathematics of Manipulation. Cambridge University Press.

[Taylor and Zwicker, 1999] Taylor, A. D. and Zwicker, W. S. (1999). Simple Games: Desirability Relations, Trading, Pseudoweightings. Princeton University Press.

[Teixeira et al., 2018] Teixeira, M., d'Orey, P. M., and Kokkinogenis, Z. (2018). Autonomous Vehicles Coordination Through Voting-Based Decision-Making. In International Conference on Agreement Technologies, pages 199–207. Springer.

[Teixeira et al., 2019] Teixeira, M., d'Orey, P. M., and Kokkinogenis, Z. (2019). Simulating collective decision-making for autonomous vehicles coordination enabled by vehicular networks: A computational social choice perspective. Simulation Modelling Practice and Theory, 98:101983.

[Tsugawa et al., 2011] Tsugawa, S., Kato, S., and Aoki, K. (2011). An automated truck platoon for energy saving. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 4109–4114. IEEE.

[Van De Hoef et al., 2015] Van De Hoef, S., Johansson, K. H., and Dimarogonas, D. V. (2015). Fuel-optimal centralized coordination of truck platooning based on shortest paths. In American Control Conference (ACC), 2015, pages 3740–3745. IEEE.

[Varaiya, 1993] Varaiya, P. (1993). Smart cars on smart roads: problems of control. IEEE Transactions on automatic control, 38(2):195–207.

[Vasirani and Ossowski, 2009] Vasirani, M. and Ossowski, S. (2009). A market-inspired approach to reservation-based urban road traffic management. In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1, pages 617–624. International Foundation for Autonomous Agents and Multiagent Systems.

[Wang, 2018] Wang, Y. (2018). Collaborative activity-based ridesharing: Linking human movement, social network, and platial semantics for future transportation. PhD thesis, The University of Melbourne.

[Wang et al., 2016] Wang, Y., Kutadinata, R., and Winter, S. (2016). Activity-based ridesharing: increasing flexibility by time geography. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, page 1. ACM.

[Watts, 2016] Watts, J. M. (2016). World's First Self-Driving Taxis Hit the Road in Singapore. The Wall Street Journal. `https://www.wsj.com/articles/worlds-first-self-driving-taxis-hit-the-road-in-singapore-1472102747`. Published 25 August 2016.

[Weiss, 1999] Weiss, G. (1999). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT press.

[Wooldridge, 2009] Wooldridge, M. (2009). An Introduction to MultiAgent Systems. John Wiley & Sons.

[Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10(2):115–152.

[Zimmermann, 2015] Zimmermann, D. (2015). Getting started with PostgreSQL in R. R-bloggers. `https://www.r-bloggers.com/2015/05/getting-started-with-postgresql-in-r/`. Retrieved 18 March 2018.

Appendix

Simulations for uniform preferences

Series 1: Minisum-Approval, uniform preferences, 100 runs each

Combination	Threshold	Quantity	Min	1st Qu.	Median	3rd Qu.	Max
sequential_basic		Dissatisfaction	0.00	0.24	0.33	0.43	0.90
sequential_iterative	DissThr 0.5	Dissatisfaction	0.00	0.16	0.25	0.34	0.50
sequential_iterative	DissThr 0.6	Dissatisfaction	0.00	0.20	0.29	0.40	0.60
sequential_iterative	DissThr 0.65	Dissatisfaction	0.00	0.23	0.32	0.41	0.65
sequential_iterative	DissThr 0.7	Dissatisfaction	0.00	0.24	0.33	0.42	0.70
sequential_iterative	DissThr 0.8	Dissatisfaction	0.00	0.24	0.33	0.42	0.76
sequential_iterative	DissThr 0.9	Dissatisfaction	0.00	0.24	0.33	0.43	0.90
coordinated_basic	JoinThr 5	Dissatisfaction	0.00	0.08	0.12	0.17	0.39
coordinated_basic	JoinThr 6	Dissatisfaction	0.00	0.08	0.12	0.17	0.53
coordinated_basic	JoinThr 7	Dissatisfaction	0.00	0.09	0.13	0.18	0.56
coordinated_basic	JoinThr 8	Dissatisfaction	0.00	0.10	0.15	0.20	0.62
coordinated_basic	JoinThr 9	Dissatisfaction	0.00	0.11	0.16	0.23	0.70
coordinated_basic	JoinThr 10	Dissatisfaction	0.00	0.12	0.19	0.27	0.64
coordinated_basic	JoinThr 11	Dissatisfaction	0.00	0.15	0.22	0.31	0.76
coordinated_basic	JoinThr 12	Dissatisfaction	0.00	0.17	0.25	0.34	0.84
coordinated_basic	JoinThr 13	Dissatisfaction	0.00	0.19	0.28	0.37	0.83
sequential_basic		Organisational Effort	3.00	9.00	14.00	19.00	34.00
sequential_iterative	DissThr 0.5	Organisational Effort	4.00	25.00	34.00	102.00	380.00
sequential_iterative	DissThr 0.6	Organisational Effort	4.00	14.00	21.00	31.00	290.00
sequential_iterative	DissThr 0.65	Organisational Effort	4.00	11.00	16.00	22.00	267.00
sequential_iterative	DissThr 0.7	Organisational Effort	4.00	10.00	15.00	21.00	260.00
sequential_iterative	DissThr 0.8	Organisational Effort	4.00	10.00	15.00	20.00	262.00
sequential_iterative	DissThr 0.9	Organisational Effort	5.00	10.00	15.00	20.00	34.00
coordinated_basic	JoinThr 5	Organisational Effort	3.00	22.00	38.00	51.00	117.00
coordinated_basic	JoinThr 6	Organisational Effort	3.00	25.00	37.00	47.00	105.00
coordinated_basic	JoinThr 7	Organisational Effort	3.00	26.00	34.00	43.00	91.00
coordinated_basic	JoinThr 8	Organisational Effort	4.00	24.00	30.00	40.00	76.00
coordinated_basic	JoinThr 9	Organisational Effort	4.00	22.00	29.00	40.00	69.00
coordinated_basic	JoinThr 10	Organisational Effort	4.00	26.00	34.00	41.00	73.00
coordinated_basic	JoinThr 11	Organisational Effort	6.00	17.00	25.00	33.00	54.00
coordinated_basic	JoinThr 12	Organisational Effort	7.00	19.00	26.00	33.00	50.00
coordinated_basic	JoinThr 13	Organisational Effort	6.00	19.00	25.00	32.00	48.00
sequential_basic		Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.5	Group Size	1.00	2.00	2.00	2.00	20.00
sequential_iterative	DissThr 0.6	Group Size	1.00	2.00	2.00	16.00	20.00
sequential_iterative	DissThr 0.65	Group Size	1.00	2.00	7.00	20.00	20.00
sequential_iterative	DissThr 0.7	Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.8	Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.9	Group Size	1.00	1.00	20.00	20.00	20.00
coordinated_basic	JoinThr 5	Group Size	1.00	1.00	1.00	1.00	3.00
coordinated_basic	JoinThr 6	Group Size	1.00	1.00	1.00	1.00	4.00
coordinated_basic	JoinThr 7	Group Size	1.00	1.00	1.00	2.00	5.00
coordinated_basic	JoinThr 8	Group Size	1.00	1.00	2.00	3.00	8.00
coordinated_basic	JoinThr 9	Group Size	1.00	1.00	2.00	4.00	11.00
coordinated_basic	JoinThr 10	Group Size	1.00	2.00	3.00	5.00	15.00
coordinated_basic	JoinThr 11	Group Size	1.00	3.00	6.00	9.00	20.00
coordinated_basic	JoinThr 12	Group Size	1.00	4.00	7.00	11.00	20.00
coordinated_basic	JoinThr 13	Group Size	1.00	5.00	10.00	14.00	20.00

Table 10.1: MS-AV, Simulations for uniform preferences, $c = 20$

Series 2: Minimax-Approval, uniform preferences, 100 runs each

Combination	Threshold	Quantity	Min	1st Qu.	Median	3rd Qu.	Max
sequential_basic		Dissatisfaction	0.00	0.30	0.39	0.48	0.74
sequential_iterative	DissThr 0.5	Dissatisfaction	0.00	0.19	0.28	0.38	0.50
sequential_iterative	DissThr 0.6	Dissatisfaction	0.00	0.23	0.33	0.43	0.60
sequential_iterative	DissThr 0.7	Dissatisfaction	0.00	0.27	0.36	0.46	0.70
sequential_iterative	DissThr 0.8	Dissatisfaction	0.00	0.28	0.37	0.46	0.79
coordinated_basic	JoinThr 5	Dissatisfaction	0.00	0.08	0.12	0.17	0.39
coordinated_basic	JoinThr 6	Dissatisfaction	0.00	0.09	0.13	0.18	0.50
coordinated_basic	JoinThr 7	Dissatisfaction	0.00	0.10	0.15	0.20	0.50
coordinated_basic	JoinThr 8	Dissatisfaction	0.00	0.11	0.17	0.23	0.55
coordinated_basic	JoinThr 9	Dissatisfaction	0.00	0.13	0.20	0.27	0.66
coordinated_basic	JoinThr 10	Dissatisfaction	0.00	0.16	0.23	0.32	0.67
coordinated_basic	JoinThr 11	Dissatisfaction	0.00	0.19	0.27	0.36	0.73
coordinated_basic	JoinThr 12	Dissatisfaction	0.00	0.21	0.30	0.39	0.80
sequential_basic		Organisational Effort	3.00	9.00	14.00	19.00	34.00
sequential_iterative	DissThr 0.5	Organisational Effort	5.00	30.00	41.00	137.00	378.00
sequential_iterative	DissThr 0.6	Organisational Effort	4.00	15.00	23.00	34.00	316.00
sequential_iterative	DissThr 0.7	Organisational Effort	5.00	11.00	16.00	22.00	299.00
sequential_iterative	DissThr 0.8	Organisational Effort	4.00	10.00	15.00	20.00	249.00
coordinated_basic	JoinThr 5	Organisational Effort	3.00	23.00	38.00	52.00	119.00
coordinated_basic	JoinThr 6	Organisational Effort	3.00	24.00	37.00	47.00	102.00
coordinated_basic	JoinThr 7	Organisational Effort	3.00	27.00	37.00	48.00	101.00
coordinated_basic	JoinThr 8	Organisational Effort	3.00	24.00	30.00	39.00	77.00
coordinated_basic	JoinThr 9	Organisational Effort	3.00	22.00	28.00	38.00	70.00
coordinated_basic	JoinThr 10	Organisational Effort	4.00	19.00	27.00	36.00	63.00
coordinated_basic	JoinThr 11	Organisational Effort	7.00	20.00	27.00	34.00	59.00
coordinated_basic	JoinThr 12	Organisational Effort	9.00	23.00	30.00	36.00	67.00
sequential_basic		Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.5	Group Size	1.00	2.00	2.00	2.00	20.00
sequential_iterative	DissThr 0.6	Group Size	1.00	2.00	2.00	7.00	20.00
sequential_iterative	DissThr 0.7	Group Size	1.00	2.00	7.00	20.00	20.00
sequential_iterative	DissThr 0.8	Group Size	1.00	1.00	20.00	20.00	20.00
coordinated_basic	JoinThr 5	Group Size	1.00	1.00	1.00	1.00	3.00
coordinated_basic	JoinThr 6	Group Size	1.00	1.00	1.00	1.00	4.00
coordinated_basic	JoinThr 7	Group Size	1.00	1.00	1.00	2.00	5.00
coordinated_basic	JoinThr 8	Group Size	1.00	1.00	2.00	3.00	10.00
coordinated_basic	JoinThr 9	Group Size	1.00	1.00	2.00	4.00	12.00
coordinated_basic	JoinThr 10	Group Size	1.00	2.00	3.00	6.00	17.00
coordinated_basic	JoinThr 11	Group Size	1.00	3.00	5.00	8.00	20.00
coordinated_basic	JoinThr 12	Group Size	1.00	4.00	7.00	10.75	20.00

Table 10.2: MM-AV, Simulations for uniform preferences, $c = 20$

Series 3: Minisum-Ranksum, uniform preferences, 100 runs each

Combination	Threshold	Quantity	Min	1st Qu.	Median	3rd Qu.	Max
sequential_basic		Dissatisfaction	0.00	0.23	0.32	0.42	0.80
sequential_iterative	DissThr 0.5	Dissatisfaction	0.00	0.13	0.23	0.34	0.50
sequential_iterative	DissThr 0.6	Dissatisfaction	0.00	0.18	0.29	0.39	0.60
sequential_iterative	DissThr 0.685	Dissatisfaction	0.00	0.21	0.30	0.41	0.68
sequential_iterative	DissThr 0.7	Dissatisfaction	0.00	0.21	0.31	0.41	0.70
sequential_iterative	DissThr 0.8	Dissatisfaction	0.00	0.22	0.32	0.42	0.78
sequential_iterative	DissThr 0.9	Dissatisfaction	0.00	0.23	0.32	0.42	0.80
coordinated_basic	JoinThr 30	Dissatisfaction	0.00	0.06	0.13	0.20	0.56
coordinated_basic	JoinThr 40	Dissatisfaction	0.00	0.13	0.21	0.29	0.68
coordinated_basic	JoinThr 50	Dissatisfaction	0.00	0.17	0.26	0.35	0.72
sequential_basic		Organisational Effort	3.00	9.00	14.00	19.00	34.00
sequential_iterative	DissThr 0.5	Organisational Effort	5.00	22.00	32.00	98.00	353.00
sequential_iterative	DissThr 0.6	Organisational Effort	4.00	13.00	20.00	28.00	260.00
sequential_iterative	DissThr 0.685	Organisational Effort	4.00	10.00	16.00	22.00	260.00
sequential_iterative	DissThr 0.7	Organisational Effort	5.00	10.00	16.00	21.00	265.00
sequential_iterative	DissThr 0.8	Organisational Effort	4.00	10.00	15.00	20.00	260.00
sequential_iterative	DissThr 0.9	Organisational Effort	4.00	10.00	15.00	20.00	34.00
coordinated_basic	JoinThr 30	Organisational Effort	4.00	26.00	34.00	42.00	75.00
coordinated_basic	JoinThr 40	Organisational Effort	8.00	20.00	28.00	35.00	54.00
coordinated_basic	JoinThr 50	Organisational Effort	7.00	18.00	25.00	32.00	49.00
sequential_basic		Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.5	Group Size	1.00	2.00	2.00	2.00	20.00
sequential_iterative	DissThr 0.6	Group Size	1.00	2.00	2.00	19.00	20.00
sequential_iterative	DissThr 0.685	Group Size	1.00	2.00	7.00	20.00	20.00
sequential_iterative	DissThr 0.7	Group Size	1.00	1.00	19.00	20.00	20.00
sequential_iterative	DissThr 0.8	Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.9	Group Size	1.00	1.00	20.00	20.00	20.00
coordinated_basic	JoinThr 30	Group Size	1.00	2.00	4.00	5.00	10.00
coordinated_basic	JoinThr 40	Group Size	1.00	4.00	7.00	9.00	18.00
coordinated_basic	JoinThr 50	Group Size	1.00	6.00	11.00	14.00	20.00

Table 10.3: MS-RS, Simulations for uniform preferences, $c = 20$

Comparing dissatisfaction values for sequential_basic, uniform

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV and MM-AV different: W = 6488600, p-value < 2.2e-16 (significant)
- alternative hypothesis: distribution for MS-AV and MS-RS different: W = 8814800, p-value = 0.0001317 (significant)

Comparing dissatisfaction values for sequential_iterative, uniform

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV and MM-AV different: W = 6944600, p-value < 2.2e-16 (significant)

- alternative hypothesis: distribution for Simulation MS-AV and MS-RS different: W = 8787900, p-value = 0.0003536 (significant)

  - alternative hypothesis: distribution for MS-AV greater than for MS-RS: W = 8787900, p-value = 0.0001768 (significant)

Comparing dissatisfaction values for coordinated_basic, uniform

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV smaller than for MM-AV: W = 6749800, p-value < 2.2e-16 (significant)

- alternative hypothesis: distribution for MS-AV greater than for MS-RS: W = 10312000, p-value < 2.2e-16 (significant)

Comparing group size for sequential_iterative, uniform

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV is greater than for MM-AV: W = 379560, p-value = 1.231e-07 (significant)

- alternative hypothesis: distribution for MS-AV is different from MS-RS: W = 173020, p-value = 0.4898 (not significant)

Comparing group size for coordinated_basic, uniform

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV is greater than for MM-AV: W = 208360, p-value < 2.2e-16 (significant)

- alternative hypothesis: distribution for MS-AV different from MS-RS: W = 80296, p-value = 0.5657 (not significant)

Comparing organisational effort for sequential_iterative, uniform

Wilcoxon test result for

- alternative hypothesis: distribution for Simulation MS-AV greater than for MS-RS: W = 8940567, p-value = 2.884e-07 (significant)
- alternative hypothesis: distribution for MS-AV smaller than for MM-AV: W = 7680924, p-value = 7.008e-12 (significant)

Comparing organisational effort for coordinated_basic, uniform

Wilcoxon test result for

- alternative hypothesis: distribution for MM-AV greater than for MS-RS: W = 9535076, p-value < 2.2e-16 (significant)
- alternative hypothesis: distribution for MS-AV different from MS-RS: W = 8442418, p-value = 0.7269 (not significant)

Minisum-Approval: Comparing dissatisfaction values for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si and cb different: W = 10625000, p-value < 2.2e-16 (significant)

Minimax-Approval: Comparing dissatisfaction values for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si and cb different: W = 10598000, p-value < 2.2e-16 (significant)

Minisum-Ranksum: Comparing dissatisfaction values for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si and cb different: W = 11837000, p-value < 2.2e-16 (significant)

Minisum-Approval: Comparing group size for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si is smaller than for cb: W = 80800, p-value < 2.2e-16 (significant)

Minimax-Approval: Comparing group size for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si is smaller than for cb: W = 208510, p-value < 2.2e-16 (significant)

Minisum-Ranksum: Comparing group size for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si is smaller than for cb: W = 78061, p-value < 2.2e-16 (significant)

Minisum-Approval: Comparing organisational effort for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si is smaller than for cb: W = 7059000, p-value < 2.2e-16 (significant)

Minimax-Approval: Comparing organisational effort for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si is greater than for cb: W = 13025000, p-value < 2.2e-16 (significant)

Minisum-Ranksum: Comparing organisational effort for sequential_iterative and coordinated_basic, uniform

Wilcoxon test result for alternative hypothesis: distribution for si is greater than for cb: W = 10489000, p-value < 2.2e-16 (significant)

Simulations for Foursquare-based preferences

Series 4: Minisum-Approval, Foursquare-based preferences, 100 runs each

Combination	Threshold	Quantity	Min	1st Qu.	Median	3rd Qu.	Max
sequential_basic		Dissatisfaction	0.00	0.15	0.58	0.77	1.00
sequential_iterative	DissThr 0.5	Dissatisfaction	0.00	0.00	0.03	0.20	0.83
sequential_iterative	DissThr 0.6	Dissatisfaction	0.00	0.00	0.04	0.36	0.84
sequential_iterative	DissThr 0.7	Dissatisfaction	0.00	0.01	0.10	0.48	0.86
sequential_iterative	DissThr 0.8	Dissatisfaction	0.00	0.04	0.33	0.62	0.84
sequential_iterative	DissThr 0.9	Dissatisfaction	0.00	0.07	0.45	0.73	0.90
sequential_iterative	DissThr 1.0	Dissatisfaction	0.00	0.15	0.58	0.77	1.00
coordinated_basic	JoinThr 5	Dissatisfaction	0.00	0.00	0.01	0.04	0.88
coordinated_basic	JoinThr 6	Dissatisfaction	0.00	0.00	0.02	0.04	0.96
coordinated_basic	JoinThr 7	Dissatisfaction	0.00	0.00	0.02	0.04	0.96
coordinated_basic	JoinThr 8	Dissatisfaction	0.00	0.01	0.04	0.29	1.00
sequential_basic		Organisational Effort	3.00	9.00	14.00	19.00	34.00
sequential_iterative	DissThr 0.5	Organisational Effort	14.00	56.00	106.50	297.00	498.00
sequential_iterative	DissThr 0.6	Organisational Effort	13.00	51.00	88.00	275.00	496.00
sequential_iterative	DissThr 0.7	Organisational Effort	5.00	42.00	65.00	216.00	497.00
sequential_iterative	DissThr 0.8	Organisational Effort	5.00	28.00	40.00	97.00	453.00
sequential_iterative	DissThr 0.9	Organisational Effort	5.00	15.00	23.00	38.00	373.00
sequential_iterative	DissThr 1.0	Organisational Effort	4.00	10.00	15.00	20.00	34.00
coordinated_basic	JoinThr 5	Organisational Effort	4.00	21.00	28.00	37.00	66.00
coordinated_basic	JoinThr 6	Organisational Effort	7.00	18.00	25.00	34.00	56.00
coordinated_basic	JoinThr 7	Organisational Effort	6.00	18.00	26.00	34.00	58.00
coordinated_basic	JoinThr 8	Organisational Effort	7.00	16.00	23.00	32.00	53.00
sequential_basic		Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.5	Group Size	1.00	1.00	2.00	2.00	14.00
sequential_iterative	DissThr 0.6	Group Size	1.00	1.00	2.00	2.00	15.00
sequential_iterative	DissThr 0.7	Group Size	1.00	2.00	2.00	2.00	20.00
sequential_iterative	DissThr 0.8	Group Size	1.00	2.00	2.00	3.00	20.00
sequential_iterative	DissThr 0.9	Group Size	1.00	2.00	2.00	8.00	20.00
sequential_iterative	DissThr 1.0	Group Size	1.00	1.00	20.00	20.00	20.00
coordinated_basic	JoinThr 5	Group Size	1.00	1.00	3.00	5.00	12.00
coordinated_basic	JoinThr 6	Group Size	1.00	2.00	5.00	7.00	15.00
coordinated_basic	JoinThr 7	Group Size	1.00	2.00	5.00	7.00	15.00
coordinated_basic	JoinThr 8	Group Size	1.00	2.00	6.00	8.00	19.00

Table 10.4: MS-AV, Simulations for Foursquare-based preferences

Series 5: Minimax-Approval, Foursquare-based preferences, 100 runs each

Combination	Threshold	Quantity	Min	1st Qu.	Median	3rd Qu.	Max
sequential_basic		Dissatisfaction	0.00	0.59	0.60	0.61	0.65
sequential_iterative	DissThr 0.5	Dissatisfaction	0.00	0.14	0.31	0.40	0.83
sequential_iterative	DissThr 0.6	Dissatisfaction	0.00	0.28	0.40	0.50	0.84
sequential_iterative	DissThr 0.625	Dissatisfaction	0.00	0.40	0.58	0.60	0.85
sequential_iterative	DissThr 0.65	Dissatisfaction	0.00	0.58	0.60	0.61	0.84
sequential_iterative	DissThr 0.7	Dissatisfaction	0.00	0.59	0.60	0.61	0.84
coordinated_basic	JoinThr 5	Dissatisfaction	0.00	0.00	0.03	0.19	0.86
coordinated_basic	JoinThr 6	Dissatisfaction	0.00	0.01	0.14	0.33	0.89
coordinated_basic	JoinThr 7	Dissatisfaction	0.00	0.01	0.18	0.38	0.88
coordinated_basic	JoinThr 8	Dissatisfaction	0.00	0.04	0.28	0.41	0.89
coordinated_basic	JoinThr 9	Dissatisfaction	0.00	0.12	0.30	0.41	0.92
sequential_basic		Organisational Effort	3.00	8.75	14.00	19.00	34.00
sequential_iterative	DissThr 0.5	Organisational Effort	20.00	64.00	132.00	302.00	495.00
sequential_iterative	DissThr 0.6	Organisational Effort	13.00	48.00	79.00	246.00	495.00
sequential_iterative	DissThr 0.625	Organisational Effort	5.00	18.00	28.00	77.25	452.00
sequential_iterative	DissThr 0.65	Organisational Effort	5.00	10.00	16.00	21.00	401.00
sequential_iterative	DissThr 0.7	Organisational Effort	4.00	10.00	15.00	21.00	323.00
coordinated_basic	JoinThr 5	Organisational Effort	4.00	22.00	30.00	38.00	76.00
coordinated_basic	JoinThr 6	Organisational Effort	6.00	18.00	25.00	34.00	56.00
coordinated_basic	JoinThr 7	Organisational Effort	8.00	18.00	25.50	34.00	57.00
coordinated_basic	JoinThr 8	Organisational Effort	8.00	19.00	25.00	33.00	54.00
coordinated_basic	JoinThr 9	Organisational Effort	7.00	19.00	25.00	33.00	55.00
sequential_basic		Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.5	Group Size	1.00	1.00	2.00	2.00	13.00
sequential_iterative	DissThr 0.6	Group Size	1.00	2.00	2.00	2.00	18.00
sequential_iterative	DissThr 0.625	Group Size	1.00	2.00	2.00	2.00	20.00
sequential_iterative	DissThr 0.65	Group Size	1.00	1.00	19.00	20.00	20.00
sequential_iterative	DissThr 0.7	Group Size	1.00	1.00	20.00	20.00	20.00
coordinated_basic	JoinThr 5	Group Size	1.00	1.00	3.00	5.00	12.00
coordinated_basic	JoinThr 6	Group Size	1.00	2.00	5.00	7.00	16.00
coordinated_basic	JoinThr 7	Group Size	1.00	2.00	5.00	7.00	20.00
coordinated_basic	JoinThr 8	Group Size	1.00	2.00	5.00	8.00	20.00
coordinated_basic	JoinThr 9	Group Size	1.00	3.00	5.00	8.00	20.00

Table 10.5: MM-AV, Simulations for Foursquare-based preferences

Series 6: Minisum-Ranksum, Foursquare-based preferences, 100 runs each

Combination	Threshold	Quantity	Min	1st Qu.	Median	3rd Qu.	Max
sequential_basic		Dissatisfaction	0.00	0.29	0.51	0.66	0.84
sequential_iterative	DissThr 0.5	Dissatisfaction	0.00	0.00	0.25	0.40	0.57
sequential_iterative	DissThr 0.6	Dissatisfaction	0.00	0.00	0.29	0.44	0.61
sequential_iterative	DissThr 0.7	Dissatisfaction	0.00	0.16	0.42	0.61	0.70
sequential_iterative	DissThr 0.8	Dissatisfaction	0.00	0.28	0.51	0.66	0.80
sequential_iterative	DissThr 0.9	Dissatisfaction	0.00	0.29	0.51	0.66	0.84
coordinated_basic	JoinThr 20	Dissatisfaction	0.00	0.00	0.00	0.01	0.61
coordinated_basic	JoinThr 30	Dissatisfaction	0.00	0.00	0.13	0.31	0.82
coordinated_basic	JoinThr 40	Dissatisfaction	0.00	0.01	0.20	0.45	0.89
coordinated_basic	JoinThr 50	Dissatisfaction	0.00	0.13	0.33	0.61	0.93
sequential_basic		Organisational Effort	3.00	9.00	14.00	19.00	34.00
sequential_iterative	DissThr 0.5	Organisational Effort	18.00	55.00	104.00	295.00	498.00
sequential_iterative	DissThr 0.6	Organisational Effort	14.00	48.00	76.00	268.00	496.00
sequential_iterative	DissThr 0.7	Organisational Effort	5.00	20.00	33.00	101.00	453.00
sequential_iterative	DissThr 0.8	Organisational Effort	4.00	10.00	15.00	20.00	257.00
sequential_iterative	DissThr 0.9	Organisational Effort	4.00	10.00	15.00	20.00	34.00
coordinated_basic	JoinThr 20	Organisational Effort	8.00	17.00	24.00	34.00	57.00
coordinated_basic	JoinThr 30	Organisational Effort	9.00	21.00	29.00	36.00	60.00
coordinated_basic	JoinThr 40	Organisational Effort	7.00	15.00	21.00	28.00	53.00
coordinated_basic	JoinThr 50	Organisational Effort	7.00	14.00	19.00	26.00	42.00
sequential_basic		Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.5	Group Size	1.00	1.00	2.00	2.00	16.00
sequential_iterative	DissThr 0.6	Group Size	1.00	1.00	2.00	2.00	17.00
sequential_iterative	DissThr 0.7	Group Size	1.00	2.00	2.00	2.00	20.00
sequential_iterative	DissThr 0.8	Group Size	1.00	1.00	20.00	20.00	20.00
sequential_iterative	DissThr 0.9	Group Size	1.00	1.00	20.00	20.00	20.00
coordinated_basic	JoinThr 20	Group Size	1.00	2.00	5.00	7.00	15.00
coordinated_basic	JoinThr 30	Group Size	1.00	3.00	6.00	9.00	20.00
coordinated_basic	JoinThr 40	Group Size	1.00	4.00	7.00	12.00	20.00
coordinated_basic	JoinThr 50	Group Size	1.00	6.00	10.00	16.75	20.00

Table 10.6: MS-RS, Simulations for Foursquare-based preferences

Comparing dissatisfaction values for sequential_basic, Foursquare

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV and MM-AV different: W = 8241000, p-value = 0.126 (not significant)
- alternative hypothesis: distribution for MS-AV and MS-RS different: W = 9628300, p-value < 2.2e-16 (significant)

Comparing dissatisfaction values for sequential_iterative, Foursquare

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV and MM-AV different: W = 8013800, p-value = 0.0002818 (significant)

- alternative hypothesis: distribution for MS-AV and MS-RS different: W = 9593800, p-value < 2.2e-16 (significant)

Comparing dissatisfaction values for coordinated_basic, Foursquare

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV and MM-AV different: W = 3462945, p-value < 2.2e-16 (significant)

- alternative hypothesis: distribution for MS-AV and MS-RS different: W = 11394576, p-value < 2.2e-16 (significant)

Comparing group sizes for sequential_iterative, Foursquare

Wilcoxon test result for alternative hypothesis: distribution for MS-AV different from MS-RS: W = 1043292, p-value < 2.2e-16 (significant)

Comparing group size for coordinated_basic, Foursquare

Wilcoxon test result for alternative hypothesis: distribution for MS-AV greater than MM-AV: W = 700066, p-value < 2.2e-16 (significant)

Comparing organisational effort for sequential_iterative, Foursquare

Wilcoxon test result for

- alternative hypothesis: distribution for MS-AV smaller than for MS-RS: W = 4265460, p-value < 2.2e-16 (significant)

- alternative hypothesis: distribution for MS-RS smaller than for MM-AV: W = 6638671, p-value < 2.2e-16 (significant)

Comparing organisational effort for coordinated_basic, Foursquare

Wilcoxon test result for alternative hypothesis:

- alternative hypothesis: distribution for MS-AV greater than for MS-RS: W = 8900488, p-value = 1.873e-06 (significant)
- alternative hypothesis: distribution for MS-AV smaller than for MM-AV: W = 6603299, p-value $<$ 2.2e-16 (significant)

Minisum-Approval: Comparing group size for sequential_iterative and coordinated_basic, Foursquare

Wilcoxon test result for alternative hypothesis: distribution for si smaller than for cb: W = 238460, p-value $<$ 2.2e-16 (significant)

Minimax-Approval: Comparing group size for sequential_iterative and coordinated_basic, Foursquare

Wilcoxon test result for alternative hypothesis: distribution for si smaller than for cb: W = 234180, p-value $<$ 2.2e-16 (significant)

Minisum-Ranksum: Comparing group size for sequential_iterative and coordinated_basic, Foursquare

Wilcoxon test result for alternative hypothesis: distribution for si smaller than for cb: W = 45862, p-value $<$ 2.2e-16 (significant)

Minisum-Approval: Comparing organisational effort for sequential_iterative and coordinated_basic, Foursquare

Wilcoxon test result for alternative hypothesis: distribution for si is greater than for cb: W = 16437000, p-value $<$ 2.2e-16 (significant)

Minimax-Approval: Comparing organisational effort for sequential_iterative and coordinated_basic, Foursquare

Wilcoxon test result for alternative hypothesis: distribution for si is greater than for cb: W = 16317000, p-value $<$ 2.2e-16 (significant)

Minisum-Ranksum: Comparing organisational effort for sequential_iterative and coordinated_basic, Foursquare

Wilcoxon test result for alternative hypothesis: distribution for si is greater than for cb: W = 16576000, p-value < 2.2e-16 (significant)