Iterative committee elections for collective decision-making in a ride-sharing application*

Sophie L. Dennisen, Jörg P. Müller
Institut für Informatik
Technische Universität Clausthal
38678 Clausthal-Zellerfeld, Germany
sophie.dennisen@tu-clausthal.de, joerg.mueller@tu-clausthal.de

Abstract

We investigate the use of voting methods for multiagent decision-making in cooperative traffic applications. We consider a ride-sharing problem in which passengers use committee elections to collectively decide on sets of points of interest to visit. In this paper, we propose an iterative voting protocol for the committee voting rules Minisum Approval and Minimax Approval. Using this protocol, voters can leave a group if their dissatisfaction with the election result exceeds a threshold value. We evaluate the rules for the ride-sharing problem using an agent-based simulation. Our results indicate that for initial group sizes around 20, both rules tend to require equal numbers of iterations for dissatisfaction threshold values around zero. We showed that Minisum Approval needs distinctly fewer iterations than Minimax Approval for values between zero and half the size of the candidate set. In some cases, for values around half the size of the candidate set, Minimax needs fewer iterations. For higher values, both rules need tendentially the same number of iterations. When aiming at minimising the number of iterations, we recommend to apply Minisum Approval for threshold values between zero and half the size of the candidate set. For higher values, we recommend to use Minimax Approval.

1 Introduction

There are diverse approaches for vehicle routing and ride-sharing problems. In vehicle routing, the goal is to design a low-cost route so that each node is visited by exactly one vehicle. In ride-sharing, it is usually assumed that each passenger has exactly one desired destination.

Here, we consider another approach for the situation that the passengers of the shared vehicles submit their preferences on all possible destinations. In this context, the question arises how to agree on a common subset of destinations to visit.

We propose to use committee voting rules to design an initial solution and to allow dissatisfied passengers to leave the group and to apply iterative winner determination until all remaining passengers are satisfied with the selected subset of destinations.

The trend in transportation systems goes towards automatisation, i.e. non-automated vehicles will be replaced by autonomous vehicles over time, increasing safety and decreasing environment pollution.

Our model is motivated by a future scenario in which traditional urban traffic is replaced by autonomous vehicles (AVs), where the city provides AVs for visitors of the city. Once a visitor boards an AV, s/he transmits his/her preferences regarding the possible destinations etc. to the AV, e.g. via smartphone app.

From the perspective of the urban traffic management, pooling the visitors into as large groups as possible is desirable as it reduces energy consumption and increases safety. Thus, we assume that the urban traffic management encourages the visitors to group together in autonomous vehicles as soon as they enter the intraurban area.

We assume that the visitors are grouped together in autonomous shared vehicles (ASVs) provided by the city at prefined boarding points, as proposed by Dennisen and Müller [2015].

We focus on visitors who submit their preferences regarding all possible destinations in the respective urban area.

This approach raises several questions.

1. How can the passengers of a shared vehicle agree on a common route?
2. How should one deal with passengers who are not satisfied with the selected route?

1.1 Outline

The remainder of the paper is structured as follows. In Section 2, the state-of-the-art is depicted, and in Section 3 the research gap is described. Section 4 gives an overview on the definitions, our solution approach and the voting architecture used for the simulations. Section 5 describes an example scenario. Section 6 includes the experimental settings, the results and the discussion, and Section 7 concludes the paper.

*This research has been supported by the German Research Foundation (DFG) through the Research Training Group SocialCars: Cooperative (De-)centralized Traffic Management (GRK 1931). The focus of the SocialCars Research Training Group is on significantly improving the citys future road traffic, through cooperative approaches. This support is gratefully acknowledged.

2 State-of-the-art

There is a range of works on the areas Ride-Sharing, Vehicle Routing and Transportation Systems. Here, we discuss a selection of those works.

2.1 The Vehicle Routing Problem

According to the review paper by Laporte [1992], the Vehicle Routing Problem (VRP) is defined as follows.

Input: $G = (V, A)$ graph where $V = \{1, .., n\}$ is a set of vertices representing cities with the depot located at vertex 1, and A is the set of arcs. With every arc $(i, j), i \neq j$, is associated a non-negative distance matrix $C = (c_{ij})$. In some contexts, c_{ij} can be interpreted as travel cost or travel time.

The goal is to design a set of least-cost vehicle routes so that

- each city except for the depot is visited by exactly one vehicle
- all vehicle routes start and end at the depot
- some side constraints are satisfied

2.2 Dynamic ride-sharing

In the review article by Agatz et al. [2012], the authors refer by dynamic ride-sharing to a system where an automated system made available by a ride-sharer provider matches up drivers and riders on short notice.

Most studies on ride-sharing consider one of the following specific objectives when determining ride-sharing matches.

- Minimise system-wide vehicle-miles
- Minimise the system-wide travel time
- Maximise the number of participants

In ride-sharing, it is assumed that each rider wants to travel from his/her origin to his/her destination.

2.3 Sharing Rides with Friends

One aspect which has been considered regarding ride-sharing is the constraints of the social network connecting the commuters. Bistaffa et al. [2014] consider the Social Ridesharing Problem, where a set of commuters, connected through a social network, arrange one-time rides at short notice. They focus on the associated optimisation problem of forming the cars to minimise the travel cost of the overall system, modelling the problem as a graph constrained coalition formation (GCCF) problem, where the set of feasible coalitions is restricted by a graph, i.e. the social network.

They assume real-time ride-sharing, arranging one-time rides with private cars and focus on providing an approach that, given the desired starting points and destinations of a community of commuters, can share cars to lower associated transportation costs, i.e. travel time and fuel, while considering the constraints imposed by the social network that connects such commuters.

2.4 Rural Flexible Transport Systems

Velaga et al. [2012] developed a passenger-centric agent-based flexible transport systems (FTS) platform using argumentation theory. Each passenger provides the following information:

- origin
- destination
- travel time window
- order of preference among: travel cost, number of changes and journey length.

The first three items are requirements, i.e. conditions that must be fulfilled for a journey to be a candidate.

The brokering subsystem gathers all the plausible journeys and composes a certain number of allocations, i.e. an assignment of passengers to sequences of vehicles.

The final step is to choose the globally preferred allocation from this set. For this, Velaga et al. [2012] use a variation of the Borda voting rule; each of the passenger agents votes by assigning a rating to each candidate allocation, and the allocation with the best rating wins. Velaga et al. [2012] do not consider committee elections.

3 Research Gap

According to Laporte [1992], in the VRP, the objective function is usually dependent on travel time or travel cost, depending on the edges. In our approach, we focus on agreeing on a common subset of POIs, disregarding the routing problem in the first phase.

In the RFTS model by Velaga et al. [2012], the candidates are a number of plausible assignments of passengers to journeys, not the points of interest (POIs), i.e. the construction of the journeys is independent from the voting process. In our approach, the voting process is necessary for the construction of the routes.

Bistaffa et al. [2014] do not consider the preferences of passengers over several destinations, but assume that each passenger has exactly one desired destination and generate coalitions with minimal cost routes.

None of these approaches focuses on how to agree on a common subset of destinations to visit based on the passengers' preferences over all possible destinations.

In this paper we propose that the passengers of a shared vehicle agree on a common subset of destinations to visit and to apply iterative winner determination until all remaining passengers are satisfied with the selected subset of destinations, i.e. in each iteration, the most dissatisfied passenger leaves the group and the remaining votes are re-evaluated.

From the operative perspective, a small number of iterations is desirable: On the one hand, the fewer iterations are conducted, the more passengers are left and the better the capacity of the shared vehicle is utilised. On the other hand, in each iteration communication between the visitors and the chair is required, i.e. minimising the number of iterations reduces the communication expense.

Thus, in this paper, we focus on the following research questions.

Assuming that visitors group together dependent on their time of arrival (i.e. in a random fashion) and only change to other shared vehicles at the starting point(s), decide on a common route via commitee election:

1. How do different committee voting rules under an iterative protocol compare regarding the number of iterations?
2. Given a committee voting rule, how many iterations will tendentially be conducted until the committee election is terminated?

4 Definitions and Methods

4.1 Definitions

Election

Here, we follow the definition in Rothe et al. [2012]. An election is defined as a tuple (C, V) where $C = \{c_1, ..., c_m\}$ is the set of candidates and $V = \{v_1, ..., v_n\}$ is the list of votes over C. Each voter is represented via his/her vote which specifies his/her preferences over the candidates in C. Which form the votes take depends on the voting rule.

Voting Rule

Following Rothe et al. [2012], given a candidate set C, a voting rule is a social-choice correspondence $f : \{(C, V)|(C, V) \text{ is a valid election}\} \longrightarrow \mathcal{P}(C)$ which assigns to each valid election (C, V) a set of winning candidates. To determine a unique winner, it can be necessary to apply a tie-breaking rule.

Committee election

Analogously to the above definition, a committee election can be defined as a tuple (C, V, k) with non-negative integer $k \leq m = |C|$.

Committee voting rule

Analogously to voting rules, one can define committee voting rules. For a given candidate set C and non-negative integer $k \leq m = |C|$, a committee voting rule is a function which assigns to each valid committee election (C, V, k) a set of winning committees. To determine a unique winning committee, it can be necessary to apply a tie-breaking rule.

Following the definition in Baumeister et al. [2015], a committee voting rule is a mapping $g : \{(C, V, k)|(C, V, k) \text{ is a valid committee election}\} \longrightarrow F_k(C)$ with $F_k(C)$ the set of all committees from C of size k.

Voting protocol

Here, we use the notion of voting protocols in the sense that a voting protocol defines the communication processes between the agents involved in the election.

Voting mechanism

In the context of this paper, a voting mechanism consists of a voting protocol and a voting rule or committee voting rule.

Committee voting rules for scenario

Both committee voting rules considered for the ride-sharing scenario assume Approval vectors, i.e. votes from $\{0, 1\}^n$, where a "0" at i-th position stands for disapproval and a "1" at i-th position for approval of the i-th candidate.

Following Brams et al. [2007a,b], the dissatisfaction of a voter v with a selected committee com is measured via the Hamming distance $HD(v, com)$.

Minisum Approval Minisum Approval selects a committee for which the sum of the Hamming distances between all votes and the committee is minimal. This corresponds to a utilitarian approach.

Minimax Approval Minimax Approval as proposed by Brams et al. [2007a,b]; Kilgour et al. [2006] selects a committee for which the maximum Hamming distance between a vote and the committee is minimal. This corresponds to an egalitarian approach.

4.2 Approach

Under the assumption that visitors of a city are encouraged to conduct round-trips in shared vehicles provided by the city, there is the question how the passengers of a shared vehicle agree on a common route. We assume that the vehicles can rank in size from taxi size to bus size.

We propose to use committee elections to agree on an initial solution. When considering the initial solution, it is possible that some passengers are dissatisfied. Such dissatisfied passengers can be allowed to leave the shared vehicle at the start point(s) and change to other vehicles. This leads to iterative winner determination. We propose to use an iterative voting protocol as depicted in Figure 1. The figure depicts the steps for the non-iterative protocol as described in Dennisen and Müller [2015] in solid lines and the additional steps for the iterative protocol in dashed lines.

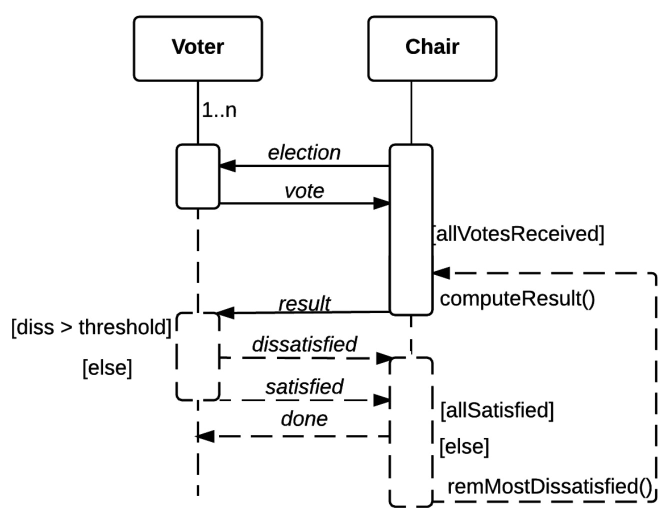


Figure 1: Centralised iterative protocol

In the non-iterative protocol, the chair starts the election by sending an `election` message to all voters and the voters respond by submitting their votes to the chair. As soon as the chair has received all votes, s/he computes the result of the election according to the given voting rule and sends the result to all voters.

In the iterative protocol, after receiving the result, the voters check via a dissatisfaction threshold if they are dissatisfied with the result. Based on their (unaltered) votes, they submit a `satisfied` or a `dissatisfied` message to the chair. If there is at least one dissatisfied voter, the chair removes the most dissatisfied voter and computes the result for the remaining voters. Otherwise, the election is terminated.

4.3 Voting Architecture

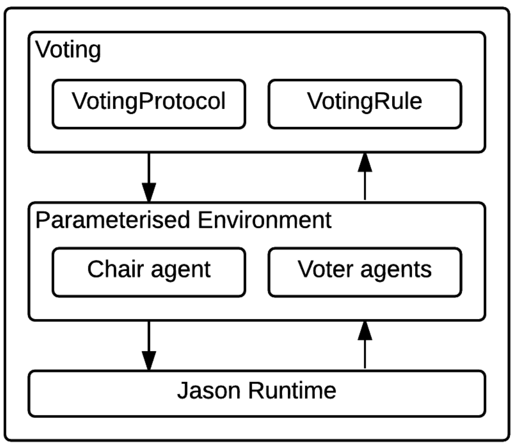


Figure 2: Voting Architecture

We decided to evaluate the behaviour of different voting mechanisms via multiagent-based simulation. This allows for testing diverse input combinations and later extension to dynamic traffic simulations.

As a voting architecture, we developed an adaptation of J-MADeM, an agent-based architecture implemented in Jason by Grimaldo et al. [2010]. Jason is an interpreter developed by Bordini et al. [2005], written in Java for an extended version of AgentSpeak, a logic-based agent-oriented programming language that is suitable for the implementation of reactive planning systems according to the Belief-Desire-Intention (BDI) architecture.

Currently, in our voting architecture, two voting protocols and two committee voting rules are implemented. The architecture allows for extension to further voting protocols and voting rules such as decentralised non-iterative protocols, decentralised iterative protocols, the voting rules Condorcet (based on pairwise comparisons), Borda and the committee voting rule Minisum-Ranksum (based on positional scores) as described in Baumeister and Dennisen [2015].

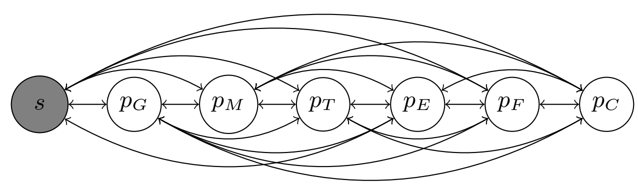
The architecture is structured as depicted in Figure 2. The Jason Runtime handles the agent cycles and the communication between the agents. The chair and voter agents are located in a parameterised environment. They receive the simulation parameters in form of initial beliefs and call the voting protocol/rule; this is realised via customisation of the Agent Architecture class.

Which modules in the Agent Architecture class are used by the respective agent depends on the role of the agent. The chair agent uses the election launcher module responsible for starting the election, the votes manager module which collects the votes and the winner determination module which computes the result of the votes according to the voting rule. The voter agents uses the voting module responsible for submitting votes.

5 Example scenario

Consider as example the following scenario. Four visitors t_1, t_2, t_3, t_4 who want to visit Manhattan, NY form together to a group at a predefined point s in Northern Manhattan. Each of them submits his/her preferences regarding all possible POIs. Due to time constraints on the operative side, their common route can only cover exactly three POIs. For simplicity, we assume that there are six possible POIs: Guggenheim Museum (p_G), MoMA (p_M), Times Square (p_T), Empire State Building (p_E), Flatiron Building (p_F) and Chinatown (p_C). The corresponding graph is depicted in Figure 3.

Figure 3: Graph for illustrating example



In the simplest model, the passengers of a shared vehicle submit their preferences in form of approval votes, i.e. they indicate approval of a candidate by assigning a "1" to it and disapproval of a candidate by assigning a "0" to it.

In the illustrating scenario, we assume that the visitors submit their preferences as approval vectors. In this case, one can apply the committee voting rule Minisum Approval. The votes and the scores are depicted in Table 1.

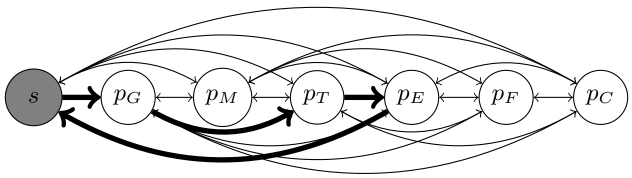
Table 1: Approval scores for illustrating example

POI	p_G	p_M	p_T	p_E	p_F	p_C
t_1	1	0	1	1	0	1
t_2	1	0	1	1	0	0
t_3	1	0	1	0	1	0
t_4	0	0	0	1	1	1
Score	**3**	0	**3**	**3**	2	2

For committee size $k = 3$, the winning committee is $K = \{p_G, p_T, p_E\}$. Assuming that the shared vehicle drives from North to South until it heads back to its starting point, the shared vehicle would take the route $s \to p_G \to p_T \to p_E \to s$, as depicted in Figure 4.

For approval vectors, the straightforward approach to measure the dissatisfaction with an elected committee is to con-

Figure 4: Resulting route



sider the Hamming distance between the respective vote and the elected committee.

The Hamming distances between the votes and the elected committee p_G, p_T, p_E are depicted in Table 2

Table 2: Hamming distances for illustrating example

	1	0	1	1	0	0	Hamming distance
t_1	1	0	1	1	0	1	1
t_2	1	0	1	1	0	0	0
t_3	1	0	1	0	1	0	2
t_4	0	0	0	1	1	1	4

Assuming dissatisfaction threshold $t = 2$, t_4 leaves the group and looks for another shared vehicle.

Table 3: Approval scores for illustrating example, second iteration

POI	p_G	p_M	p_T	p_E	p_F	p_C
t_1	1	0	1	1	0	1
t_2	1	0	1	1	0	0
t_3	1	0	1	0	1	0
Score	**4**	0	**4**	**3**	1	1

In this case, the removal of the dissatisfied voter doesn't alter the outcome of the committee election, so the route stays the same.

6 Evaluation

We investigated the impact of the dissatisfaction threshold for the visitors t, the number of POIs to be visited k and the number of offered POIs m on the number of iterations needed by the voting rules Minisum and Minimax Approval.

6.1 Experimental Settings

Our simulations were conducted with the following technical settings.

- Jason-1.4.2
- Java 1.8.0_65
- Windows 8.1
- HDF5 for storing input and output data
- R x64 3.2.3 for evaluation

As configuration parameters for the simulation, we have

- the number $n = 20$ of voters (visitors): $n = 20$ is oriented towards bus sizes
- the number m of candidates (POIs)
- the size k of the committee to be elected
- the dissatisfaction threshold t
- the committee voting rules
- the voting protocol(s)

For each run, the votes are generated as follows: For each position in each vote, a "1" or a "0" is selected with equal probability, resulting in homogenous electorates.

In each run, the number of necessary iterations is saved.

We measured and compared the number of iterations under the iterative protocol for Minimax Approval and Minimax Approval for several input combinations (n, m, k, t).

In each simulation, we conduct 100 runs and measure the median of the numbers of iterations for Minisum and Minimax as well as the median of differences $iteration\_minisum - iterations\_minimax$. In our simulations, we consider three different settings.

1. Vary the values for dissatisfaction threshold t
2. Vary the values for committee size k
3. Vary the values for number of candidates m

6.2 Results

Exploring the impact of dissatisfaction threshold t on iteration numbers

In our first setting, we considered all possible values of dissatisfaction threshold t for number of voters $n = 20$, number of candidates $m = 10$, committee size $k = 5$, i.e. $0 \le t \le m = 10$. The results are depicted in Figure 5. The figure shows that the number of iterations decreases for both voting rules with increasing dissatisfaction threshold t. For values of t around 0, it is impossible to satisfy the voters, so that both voting rules need 20 iterations, creating empty groups. For smaller values of t above zero, i.e. for hard-to-please voters, Minisum needs tendentially fewer iterations than Minimax. For values of t above $m/2$ and near $m/2$, i.e. for more tolerant voters, Minimax needs fewer iterations than Minisum. For higher values, both rules need zero iterations.

Consider the input combination $n = 20, m = 10, k = 5, t = 4$. The boxplot for the differences between Minisum and Minimax is depicted in Figure 6. The median of the differences is -3, i.e. Minisum needs tendentially 3 iterations fewer than Minimax for dissatisfaction value $t = 3$.

For $t = 4$, the Wilcoxon rank sum test with continuity correction yields $W = 988.5$ and p-value $< 2.2e - 16 < 0.05$, i.e. we can reject the null hypothesis that there is no statistical difference between the distributions.

Consider the input combination $n = 20, m = 10, k = 5, t = 6$. The boxplot for the differences between Minisum and Minimax is depicted in Figure 7. The median of the differences is 1, i.e. Minimax needs tendentially one iterations fewer than Minisum.

For $t = 6$, the Wilcoxon rank sum test with continuity correction yields $W = 8734.5$ and p-value $< 2.2e - 16 < 0.05$, i.e. we can reject the null hypothesis that there is no statistical difference between the distributions.

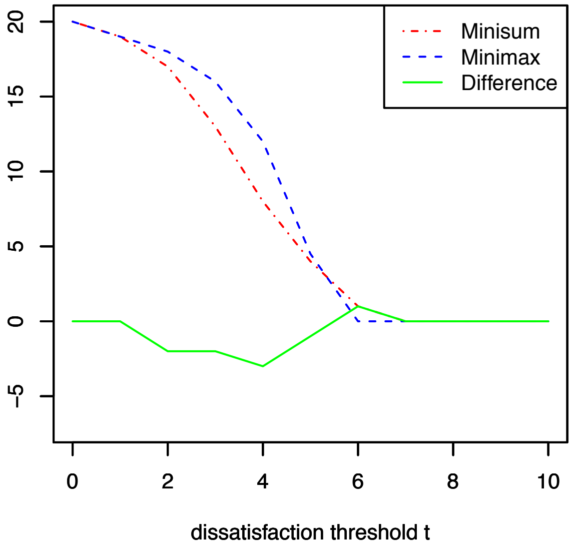


Figure 5: Median of iteration numbers for Minisum and Minimax and median of differences (Minisum-Minimax) for $n = 20, m = 10, k = 5$ against dissatisfaction threshold t.

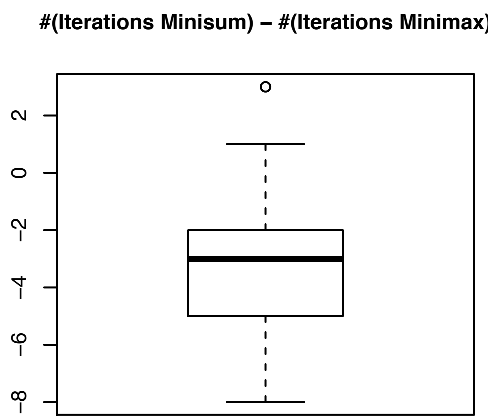


Figure 6: Boxplot of differences for $n = 20, m = 10, k = 5$, $t = 4$

Exploring the impact of committee size k on iteration numbers

In the second setting, we fixed the number of candidates $m = 10$ the number of voters $n = 20$, varied committee size k, i.e. the number of effectively visited POIs and measured the median of differences for dissatisfaction threshold values $t = 5$ and $t = 6$.

The results for $t = 5$ are depicted in Table 4 and Figure 8, the results for $t = 6$ in Table 5 and Figure 9.

Table 4 shows that both voting rules need fewer iterations for medium values of k, i.e. the closer k is to $m/2$, the fewer iterations are needed. For $t = 6$, the median of the differences between Minisum and Minimax is -1, i.e. Minisum needs tendentially one iteration fewer than Minimax.

Table 5 and Figure 9 show a similar trend. Both voting rules need fewer iterations for medium values of k. Here, Minimax needs one iteration fewer than Minisum for medium values of k. For other values of k, both rules need tendentially the same number of iterations.

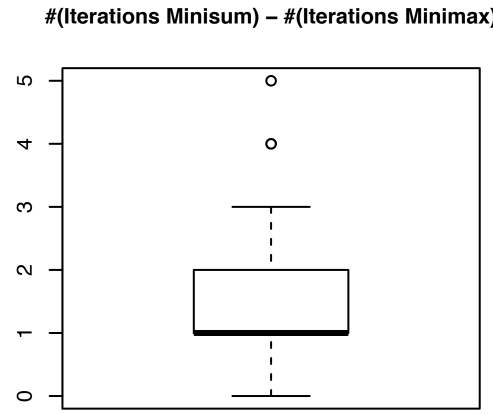


Figure 7: Boxplot of differences for $n = 20, m = 10, k = 5$, $t = 6$

Table 4: Medians for $m = 10, n = 20, t = 5$ and varying k

k	Iterations Minisum	Iterations Minimax	Difference
1	6	7	-1
2	5	6	-1
3	5	5	-1
4	5	5	-1
5	4	5	-1
6	4	5	-1
7	5	6	-1
8	5	6	-1
9	7	9	-1

Table 5: Medians for $m = 10, n = 20, t = 6$ and varying k

k	Iterations Minisum	Iterations Minimax	Difference
1	2	2	0
2	2	1	0
3	2	0	1
4	1	0	1
5	1	0	1
6	1	0	1
7	2	0	1
8	2	1	1
9	2	3	0

Consider the input combination $n = 20, m = 10, k = 5, t = 6$. The boxplot for the differences between Minisum and Minimax is depicted in Figure 10. The median of the differences is 1, i.e. Minimax needs tendentially one iterations fewer than Minisum.

The Wilcoxon rank sum test with continuity correction yields $W = 8667.5$ and p-value $< 2.2e - 16 < 0.05$, i.e. we can reject the null hypothesis that there is no statistical difference between the distributions.

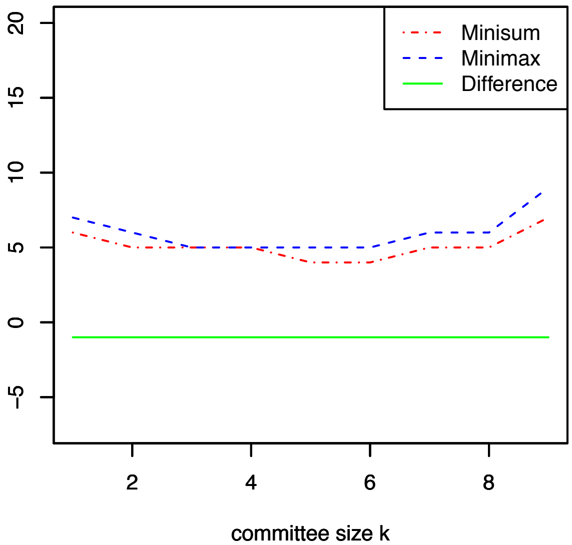


Figure 8: Median of iteration numbers for Minisum and Minimax and median of differences (Minisum-Minimax) for $n = 20, m = 10, t = 5$ against committee size k

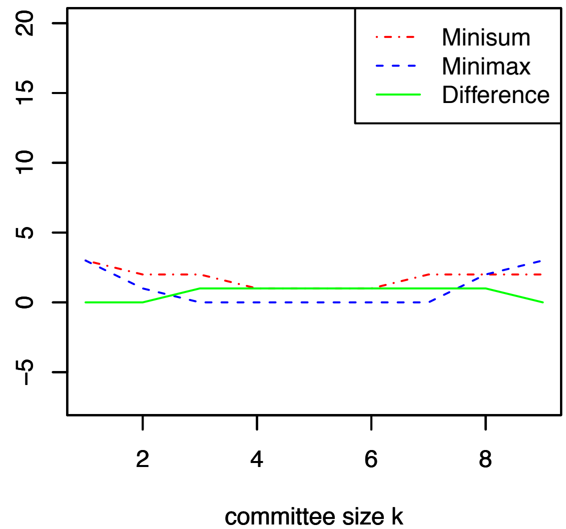


Figure 9: Median of iteration numbers for Minisum and Minimax and median of differences (Minisum-Minimax) for $n = 20, m = 10, t = 6$ against committee size k

Exploring the impact of number of candidates m on iteration numbers

In the third setting, we fixed $n = 20, k = 5, t = \lfloor m/2 \rfloor, \lceil m/2 \rceil$ and measured the median of differences for different numbers of offered POIs $m = 10, 15, 20$. The results are depicted in Table 6.

Table 6: Median of differences for $m = 10, 15, 20, n = 20, k = 5, t = \lfloor m/2 \rfloor, \lceil m/2 \rceil$

	Minisum	Minimax	Difference
$m = 10, k = 5, t = 5$	4	4.5	-1
$m = 15, k = 5, t = 7$	6	6	0
$m = 15, k = 5, t = 8$	3	1	2
$m = 20, k = 5, t = 10$	5	4	1

Table 6 shows no clear relation between the number of the candidates and the iteration numbers for Minisum and Minimax Approval. For $m = 15, k = 5$, one can again see the influence of t on the numbers of iterations.

6.3 Discussion

Assuming shared vehicles with a capacity of 20 and numbers of offered POIs up to 20, our results indicate that a favourable constellation from operative perspective would be to offer two times as many POIs as can be visited by a shared vehicle - the voting rules need fewer iterations if the number of effectively visited POIs lies around half the number of offered POIs.

The dissatisfaction threshold has an considerable impact: For threshold values around 0, Minisum and Minimax tend to need the same number of iterations. For higher values below $m/2$, i.e. if the voters are hard to please, Minisum needs distinctly fewer iterations than Minimax.

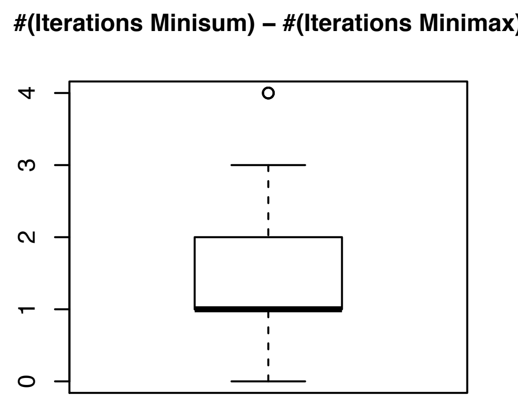


Figure 10: Boxplot of differences for $n = 20, m = 10, k = 5, t = 6$

If k lies around $m/2$, Minimax needs tendentially fewer iterations than Minisum for values of t close to $m/2$, i.e. for more tolerant voters.

For higher values of t, the difference decreases until both committee voting rules tend to need the same number of iterations.

In practice, there are several motives for minimising the number of iterations.

- Capacity utilisation: The fewer iterations are conducted, the more visitors remain in the shared vehicle
- Communication expense: In each iteration, the visitors have to communicate with the chair in order to indicate if they are satisfied or dissatisfied.

In order to minimise the number of iterations, we recommend to apply Minisum Approval for the case that it is expected that the visitors are hard-to-please. If you expect that the visitors are more tolerant, we recommend to use Minimax Approval.

So far, Computational Social Choice methods have been largely subject to theoretical analysis. There are hardly any attempts to use them in the engineering of socio-technical multiagent systems such as traffic modeling and management. The ride-sharing scenario is relatively simple but we believe it is yet suitable as an experimental scenario due to its relative generality and the relevance (and hardness) of the underlying optimisation problems. The concept is applicable for non-autonomous driving as well: In the case of non-autonomous driving, one could equate the chair agent with the owner/ driver.

Our next step will be to reproduce our results for further input combinations (n, m, k, t). Note that we conducted investigations for relative small numbers of available POIs. For larger numbers of POIs, we aim to compare the properties of Minisum Approval and Minimax Approximation algorithms.

In the setting considered in this paper, we used a a fixed dissatisfaction threshold to determine the dissatisfaction of the visitors. In a dynamic scenario, it would make sense to let the visitors decide individually if they are satisfied or dissatisfied. To simulate this, one would need a stochastic model to determine the dissatisfaction thresholds.

Furthermore, we will consider the situation that the voters cannot only leave their initially assigned groups but change to another groups.

Also, a challenge for future research is to study the runtime performance of the voting mechanisms taking the time requirements of collective decision situations in real traffic into account.

7 Conclusion

In this paper, we investigated the usability of methods known from tha area of computational social choice in future cooperative traffic environments consisting of automated or human-operated vehicles, able to communicate with each other, e.g. using Vehicle-to-X communication technologies. In particular, we considered a ride-sharing scenario where visitors of a city share vehicles with seating capacities similar to buses to visit points of interest.

We proposed an iterative voting protocol based on the well-known Minisum Approval and Minimax Approval committee voting rules, allowing dissatisfied travellers to leave a group and join a different one. Using an agent-based simulation, we compared iterative Minisum Approval and Minimax Approval with respect to their convergence properties.

The main result is that iterative Minisum Approval outperforms Minimax approval in this respect for threshold values higher than 0 and lower than $m/2$. If k lies around $m/2$, there is a slight advantage to Minimax Approval for values of t close to $m/2$.

References

Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012.

Dorothea Baumeister and Sophie Dennisen. Voter Dissatisfaction in Committee Elections. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1707–1708. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

Dorothea Baumeister, Sophie Dennisen, and Lisa Rey. Winner Determination and Manipulation in Minisum and Minimax Committee Elections. In *Algorithmic Decision Theory*, pages 469–485. Springer, 2015.

Filippo Bistaffa, Alessandro Farinelli, and Sarvapali D Ramchurn. Sharing rides with friends: a coalition formation algorithm for ridesharing. 2014.

Rafael H Bordini, Jomi F Hübner, and Renata Vieira. Jason and the golden fleece of agent-oriented programming. In *Multi-agent programming*, pages 3–37. Springer, 2005.

Steven J Brams, D Marc Kilgour, and M Remzi Sanver. A Minimax Procedure for Electing Committees. *Public Choice*, 132(3-4):401–420, 2007a.

Steven J Brams, D Marc Kilgour, and M Remzi Sanver. A Minimax Procedure for Negotiating Multilateral Treaties. In *Diplomacy games*, pages 265–282. Springer, 2007b.

Sophie L Dennisen and Jörg P Müller. Agent-Based Voting Architecture for Traffic Applications. In *Multiagent System Technologies*, pages 200–217. Springer, 2015.

Francisco Grimaldo, Miguel Lozano, Fernando Barber, and Alejandro Guerra-Hernández. J-MADeM v1. 1: A full-fledge AgentSpeak (L) multimodal social decision library in Jason. In *The 8th European Workshop on Multi-Agent Systems (EUMAS 2010)*, 2010.

D Marc Kilgour, Steven J Brams, and M Remzi Sanver. How to Elect a Representative Committee using Approval Balloting. In *Mathematics and Democracy*, pages 83–95. Springer, 2006.

Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

Jörg Rothe, Dorothea Baumeister, Claudia Lindner, and Irene Rothe. *Einführung in Computational Social Choice: Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen*. Springer-Verlag, 2012.

Nagendra R Velaga, Nicolás D Rotstein, Nir Oren, John D Nelson, Timothy J Norman, and Steve Wright. Development of an integrated flexible transport systems platform for rural areas using argumentation theory. *Research in Transportation Business & Management*, 3:62–70, 2012.