Itero: An Online Iterative Voting Application*

Joseph Boudou , Rachael Colley , Umberto Grandi
IRIT, University of Toulouse, France
joseph.boudou@matabio.net
{rachael.colley, umberto.grandi}@irit.fr

Abstract

Iterative voting allows a group of agents to take a collective decision in a dynamic fashion: a series of plurality elections are staged, making the relative scores of the candidates public after each round. Voters can thus adjust their ballots at each step until the process converges (or a maximal number of steps is reached). Research in computational social choice has shown that this method has the potential of reaching good-quality decisions while at the same time being easy to explain to voters. This paper presents our implementation of iterative voting on a voting platform accessible on the web.

1 Introduction

Improving collective decision-making with AI techniques is one of the challenges faced by researchers in computational social choice for the last couple of decades. The first stream of papers in the field explored the algorithmic and game-theoretic aspects of voting (for an overview see, e.g., Brandt *et al.* [2016]). A second wave pushed the research agenda further by, among other things, testing algorithms and theoretical results on voting data obtained through behavioural experiments and voting platforms, leading to the creation of a repository for voting and preference data named Preflib [Mattei and Walsh, 2013; Mattei and Walsh, 2017].

Interest grew in iterated elections among researchers in multi-agent systems as a mean to study voting equilibria reachable with best-response dynamics (see, e.g., Meir *et al.* [2010]; Lev and Rosenschein [2012]). Since then, iterative voting has received considerable attention both as a predictive model for voters' response to poll information (see, e.g., Lev and Rosenschein [2016]; Wilczynski [2019]; Meir *et al.* [2020]), and as a novel approach to the design of voting rules for human or artificial agents (see, e.g., Grandi *et al.* [2013]; Obraztsova *et al.* [2015]; Airiau *et al.* [2017]). Two main reasons motivate the latter research agenda. First, iterative voting can improve the result of simple voting rules with bad properties. Plurality elections are the most well-known example of this phenomenon. In a plurality election voters are simply asked to designate their favourite candidate and the candidate with the most votes is named the plurality winner. Plurality winners can have a lower social welfare than other candidates, and the plurality rule often does not elect a Condorcet winner when one exists.[1] Instead, theoretical work and computer simulations showed that iterated plurality elections have a higher social welfare and Condorcet efficiency than one-shot plurality elections [Grandi *et al.*, 2013; Bowman *et al.*, 2014; Kavner and Xia, 2021].[2] Second, repeating a simple rule still leads to a voting rule that is easy to explain to voters, which is not the case for many of the voting rules which enjoy many beneficial axiomatic properties. Moreover, the iterative process has the potential of reaching a consensus among the voters, improving their participatory experience.

In previous work we tested iterated voting methods with behavioural experiments in a lab facing a simple combinatorial election with two binary questions [Grandi *et al.*, 2020], as well as with autonomous learning agents based on reinforcement learning [Airiau *et al.*, 2017], showing that such autonomously reached collective decisions are comparable to those obtained by well-known and well-studied voting rules. In this paper we present our implementation of iterative voting as a web application called Itero. Its main functionalities are the following:

- The election organiser can specify a set of alternatives, a maximal round duration, and a final deadline.
- The election organiser can share a link with a set of voters or make the election public.
- A series of iterated plurality elections are taken among all users who voted in the first round and continue until the deadline.
- At each step the plurality score (and the percentage of the votes) of each candidate as well as the scores from all past rounds are accessible to the voters.

To the best of our knowledge, Itero is the first voting platform that proposes iterative elections. A survey of social choice experiences on the web was compiled by Bou-

*Webpage: https://itero.irit.fr

[1] Plurality received no support from experts in social choice when asked to vote on voting rules [Laslier, 2012].

[2] Such findings are confirmed by one behavioural experiment [Grandi *et al.*, 2020], while a second one did not find any evidence of improvement in social welfare [Meir *et al.*, 2020].

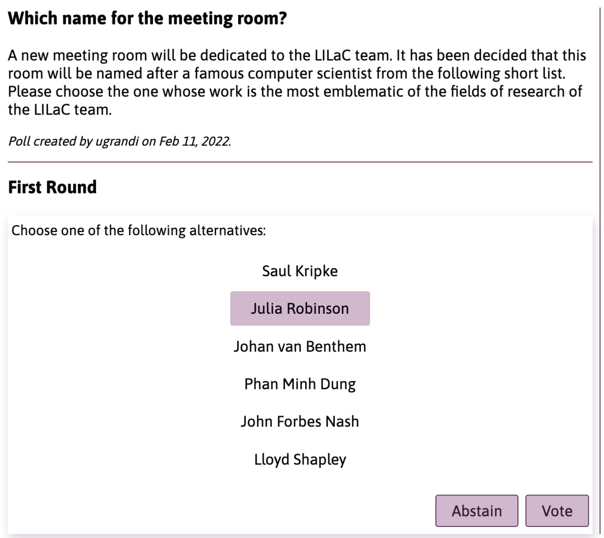

Figure 1: The voting interface allows users to see the question, the description, and the alternatives of the poll. Voters are able to vote for one of the options or abstain from this round of the poll.

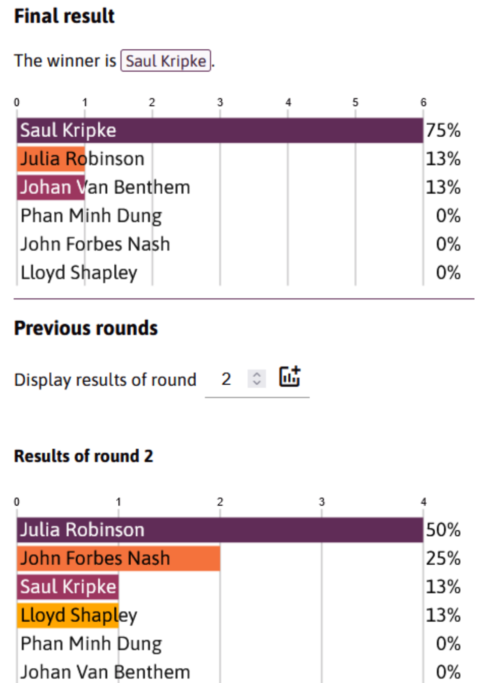

Figure 2: Final results of an iterated election, showing the percentage of the votes received by each alternative in the final round and in any past round.

veret [2017], but more platforms have appeared since then. One such example is OPRA[3] [Chen *et al.*, 2021], a complete platform where different voting rules and elicitation protocols can be experimented on, and that also allows some forms of iterative voting, albeit without a turn-taking protocol. Also, recent work by Meir *et al.* [2020] proposed a simple interface for experimenting with voters' response to a poll.[4]

2 The Application

The Itero application is an iterative voting platform that helps groups reach a collective decision. Users can create iterative voting campaigns (we will refer to this as a poll in this paper) and share them with the voters. We report screenshots from a pilot that we ran for our research group, where the poll asked which logician should the meeting room be named after. The screenshots are from the laptop version of the application, but the UI is also optimised for mobile.

2.1 Voting UI

Voters access polls by using the link provided by the creator of the poll (in this case users can participate in the poll without an Itero account) or by accessing public polls posted on the website. When they have accessed a poll, the voter interface displays to them the question, the poll description, and the alternatives, as shown in Figure 1. The users vote by either choosing the alternative that they currently prefer, which could be influenced by the previous rounds of voting, or abstaining from this round of the poll. Once an alternative has been confirmed or a voter has abstained, their vote for the current round is recorded. If the round is still in progress then voters still have the option to change their vote for the current round. Note that voters are unable to see the outcome of the current round until it has finished. The results of each of the previous rounds are accessible in a simple visual interface which can be seen in Figure 2.

2.2 Poll Creation

The user interface guides users in creating a poll in four steps. The first step requires the user to enter the question that should be answered by the poll. The poll creator also has the option to give a description of the poll (this could be details about the alternatives or how the outcome of the poll will be used). The description is displayed to the voters at each voting round, as seen in Figure 1.

Following this, the creator enters the alternatives of the poll, i.e., the options that voters will be voting on. Note that a minimum of two alternatives is required.

In the third step the poll creator is required to set up the parameters of the rounds for the iteration, using an interface that can be seen in Figure 3. First, specifying when the poll begins and the latest end time of the poll. Second, giving the minimum and the maximum number of rounds they would like their poll to have (the default values are between 2 and 10 rounds). Third, entering the maximum length of any given round. The poll will end when either the maximum number of rounds has been met or the end date has been reached. After the first round ended the mechanism does not allow for new voters to enter. Due to this, all following rounds end either when every voter has updated their vote or when the round's duration has elapsed. In each round, voters are able to change

[3]https://opra.cs.rpi.edu

[4]Their data, together with code to evaluate election data, is accessible at https://pypi.org/project/votelib/.

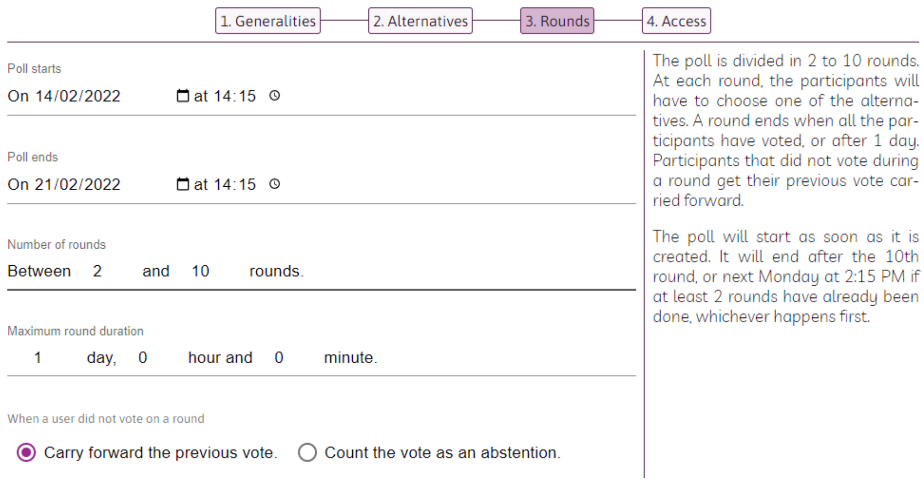

Figure 3: Screenshot of the third step of the UI for the poll creator, where the duration of the poll, the minimal and maximal number of rounds, and the maximal duration of each round can be specified. The explanatory text on the right of the figure is updated in real-time while the poll creator enters data in the form.

their vote, abstain, or do nothing, the creator decides beforehand if a voter not updating their vote in a round equates to either their vote remaining the same or changing to an abstention. In the fourth step the poll creator decides if it is required for voters to have an Itero account. Also, whether the poll should be public (visible to all users through the platform) or users be invited manually through a shareable link.

Note that while the poll is being created, a description of the poll (as shown on the right-hand side of Figure 3) at each step is immediately updated while changes are being made.

2.3 Technical Specification

Itero has been designed such that it can be easily modified and extended. Thus, only standard technologies have been used. Its architecture is as follows. A front end application, implemented in Angular,[5] runs on the users' browser, displaying the user interface. This front end application communicates with the Itero server through a simple API consisting of JSON messages sent using HTTPS. The Itero server, implemented using the language Go,[6] provides information to the front end and updates the database according to users' actions. The server also sends email notifications to users. The Itero database is stored on a MariaDB DBMS server.[7] The source code of Itero is freely accessible on GitHub.[8] It is provided under an open-source license.

2.4 Experiences

We have not yet performed experiments using the Itero platform going further than pilot sessions. Here, we want to highlight some of our experiences with the platform. In the pilot test shown in the screenshots, i.e., the decision on the name of the meeting room, we observed an interesting exploratory behaviour by the voters, with the winner changing in several of the initial steps after which the voters coordinated or converged on one name. Note that most research in iterative voting assumes that voters change their ballots one-by-one, while in Itero all changes occur simultaneously, making coordination among the voters harder. We also used the Itero platform to collectively decide on which presenter should win the best talk prize at a workshop we organised. In this scenario we observed instead a convergence from the first round on the winner, with voters trying to ensure in subsequent rounds that all other candidates had a least some votes and adjusting the ranking from the second position onward.

3 Conclusions

The primary purpose of the Itero application is to showcase the benefits of iterative elections at outreach events and online. Its interface is simple enough to be used by any election organiser, for example, to decide on the date for an important meeting, which projects to give priority to, or any collective decision where the opinion of the decision-makers tending towards a consensus is of importance. Observe that the plurality score at convergence can be used to obtain a ranking of the alternatives if the situation requires more than a single winner. The Itero application can also be used by researchers for experimenting on voting behaviour, such as monitoring the response of voters to poll information.

Several extensions of the platform will be explored in future work. The most prominent one is allowing for continuous elections in which users can join and change their vote at any time, without predefined rounds. A second direction is to allow for iterative versions of more complex collective decisions such as participatory budgeting, making them more interactive and visual in an attempt to increase engagement.

[5] https://angular.io/
[6] https://go.dev/
[7] https://mariadb.org/
[8] https://github.com/JBoudou/Itero

Acknowledgments

The Itero platform is funded by the ANR JCJC project SCONE (ANR 18-CE23-0009-01). We also acknowledge the work of five students from Master MIAGE "Ingénierie Métier" who worked on the mobile version of the application.

References

[Airiau *et al.*, 2017] Stéphane Airiau, Umberto Grandi, and Filipo Studzinski Perotto. Learning agents for iterative voting. In *Proceedings of the 5th International Conference on Algorithmic Decision Theory (ADT)*, 2017.

[Bouveret, 2017] Sylvain Bouveret. Social choice on the web. In Ulle Endriss, editor, *Trends in Computational Social Choice*, chapter 20, pages 387–402. AI Access, 2017.

[Bowman *et al.*, 2014] Clark Bowman, Jonathan K. Hodge, and Ada Yu. The potential of iterative voting to solve the separability problem in referendum elections. *Theory and Decision*, 77(1):111–124, 2014.

[Brandt *et al.*, 2016] Felix Brandt, Vincent Conitzer, Ulle Endriss, Ariel D Procaccia, and Jérôme Lang. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[Chen *et al.*, 2021] Yiwei Chen, Jingwen Qian, Junming Wang, Lirong Xia, and Gavriel Zahavi. OPRA: an open-source online preference reporting and aggregation system. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[Grandi *et al.*, 2013] Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Restricted manipulation in iterative voting: Condorcet efficiency and Borda score. In *Proceeding of the 3rd International Conference on Algorithmic Decision Theory (ADT)*, 2013.

[Grandi *et al.*, 2020] Umberto Grandi, Jérôme Lang, Ali Ozkes, and Stéphane Airiau. Voting behavior in one-shot and iterative multiple referenda. *Université Paris-Dauphine Research Paper Number 3747713*, ssrn.com/abstract=3747713, 2020.

[Kavner and Xia, 2021] Joshua Kavner and Lirong Xia. Strategic behavior is bliss: Iterative voting improves social welfare. *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[Laslier, 2012] Jean-François Laslier. And the loser is...plurality voting. In *Electoral systems*, pages 327–351. Springer, 2012.

[Lev and Rosenschein, 2012] Omer Lev and Jeffrey S. Rosenschein. Convergence of iterative voting. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

[Lev and Rosenschein, 2016] Omer Lev and Jeffrey S. Rosenschein. Convergence of iterative scoring rules. *Journal of Artificial Intelligence Research*, 57:573–591, 2016.

[Mattei and Walsh, 2013] Nicholas Mattei and Toby Walsh. Preflib: A library of preference data HTTP://PREFLIB.ORG. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*, 2013.

[Mattei and Walsh, 2017] Nicholas Mattei and Toby Walsh. Apreflib. org retrospective: Lessons learned and new directions. In *Trends in Computational Social Choice*, pages 289–309. AI Access, 2017.

[Meir *et al.*, 2010] Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, and Nicholas R. Jennings. Convergence to equilibria in plurality voting. In *Proceedings of the 24th conference on Artificial Intelligence (AAAI)*, 2010.

[Meir *et al.*, 2020] Reshef Meir, Kobi Gal, and Maor Tal. Strategic voting in the lab: compromise and leader bias behavior. *Autonomous Agents and Multi Agent Systems*, 34(1):31, 2020.

[Obraztsova *et al.*, 2015] Svetlana Obraztsova, Evangelos Markakis, Maria Polukarov, Zinovi Rabinovich, and Nicholas R. Jennings. On the convergence of iterative voting: How restrictive should restricted dynamics be? In *Proceedings of the 19th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[Wilczynski, 2019] Anaëlle Wilczynski. Poll-confident voters in iterative voting. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 2019.