



UvA-DARE (Digital Academic Repository)

Aggregating Dependency Graphs into Voting Agendas in Multi-Issue Elections

Airiau, S.; Endriss, U.; Grandi, U.; Porello, D.; Uckelman, J.

DOI

[10.5591/978-1-57735-516-8/IJCAI11-015](https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-015)

Publication date

2011

Document Version

Final published version

Published in

IJCAI-11

[Link to publication](#)

Citation for published version (APA):

Airiau, S., Endriss, U., Grandi, U., Porello, D., & Uckelman, J. (2011). Aggregating Dependency Graphs into Voting Agendas in Multi-Issue Elections. In T. Walsh (Ed.), *IJCAI-11: proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence : Barcelona, Catalonia, Spain, 16-22 July 2011* (Vol. 1, pp. 18-23). AAAI Press/International Joint Conferences on Artificial Intelligence. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-015>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Aggregating Dependency Graphs into Voting Agendas in Multi-Issue Elections

Stéphane Airiau, Ulle Endriss, Umberto Grandi, Daniele Porello, and Joel Uckelman

Institute for Logic, Language and Computation

University of Amsterdam

{s.airiau,u.endriss,u.grandi,d.porello,j.d.uckelman}@uva.nl

Abstract

Many collective decision making problems have a combinatorial structure: the agents involved must decide on multiple issues and their preferences over one issue may depend on the choices adopted for some of the others. Voting is an attractive method for making collective decisions, but conducting a multi-issue election is challenging. On the one hand, requiring agents to vote by expressing their preferences over all combinations of issues is computationally infeasible; on the other, decomposing the problem into several elections on smaller sets of issues can lead to paradoxical outcomes. Any pragmatic method for running a multi-issue election will have to balance these two concerns. We identify and analyse the problem of generating an agenda for a given election, specifying which issues to vote on together in local elections and in which order to schedule those local elections.

1 Introduction

Many problems where a group of agents need to make a decision have a combinatorial structure: for each of a set of variables the agents need to choose a value, and each agent has preferences over combinations of such choices. Examples include voting in multiple referenda, where we have to decide which of a set of propositions to accept, or electing a committee, where we have to decide how to fill each seat. Voting in combinatorial domains has been studied in AI for some time, due both to its relevance to multiagent systems and to the fact that methods developed in knowledge representation for modelling preferences provide important tools for tackling some of the challenges it raises [Chevalyere *et al.*, 2008; Lang and Xia, 2009; Li *et al.*, 2010; Rossi *et al.*, 2004].

When the issues to be decided upon are independent from each other, then running a separate election for each issue is a good solution. However, if some of the voters have *preferential dependencies* among issues, then this can lead to paradoxical outcomes, because voters will have to commit to a vote on one issue before all the other issues their preferences depend on have been settled [Brams *et al.*, 1998]. Voting on all the issues in a single election instead is not feasible for computational reasons: there will be too many combinations for the voters

to rank and asking them to rank only some of them will yield too little information to select a winner [Chevalyere *et al.*, 2008]. A useful middle way is to vote *sequentially*: hold local elections on each issue in sequence and reveal the outcome of one election before the next one takes place [Lacy and Niou, 2000]. When there exists an *agenda* (i.e., a way of scheduling these local elections) that is compatible with the preferential dependencies of all the voters, then we avoid the paradoxes typically encountered [Lang and Xia, 2009].

A generalisation of this idea is to hold a *local election* for every combination of issues that cannot be decomposed due to preferential dependencies of some of the voters. If no local election is for a combination of a large number of issues (e.g., up to three issues), then the agents should be able to express a ranking over the relevant combinations, and a standard voting procedure can be used. In this paper, we isolate this problem of *agenda choice*. We do this by introducing choice functions that take the preferential dependencies of the voters (rather than their full preference information) as input and that produce an agenda determining the order in which local elections for (small) combinations of issues are to be held. We make several concrete proposals for such choice functions that allow us to both keep the complexity of local elections at a manageable level and to limit social choice-theoretic paradoxes arising from the voters' uncertainty over their preferences.

The agenda choice problem is central to the challenge of voting in combinatorial domains, but it has not previously been identified as a research question in its own right. Rather than proposing a "solution" to this difficult problem, we introduce a technical framework in which to study it. Our specific suggestions for agenda choice functions should be considered examples against which to measure future proposals.

Much previous work on voting in combinatorial domains concentrating on computational issues has assumed that the preferences of voters are represented using CP-nets [Boutilier *et al.*, 2004]. We do not make this assumption, but our framework is compatible with preferences being modelled as CP-nets. Like us, Lang and Xia [2009] address the problem of running several local elections to manage a large multi-issue election. They study a domain restriction on preference profiles that guarantees that voters can vote issue-by-issue without provoking a paradoxical outcome. Other related work includes the contributions of Li *et al.* [2010], who propose an algorithm for aggregating CP-nets which have a *common* graph struc-

ture (the opposite of our setting), and of Rossi *et al.* [2004], who introduce so-called *mCP*-nets, a language for describing profiles of CP-nets (rather than aggregating them).

The remainder of this paper is organised as follows. Section 2 defines our formal framework, including the problem of choosing an agenda. Section 3 then makes several proposals for procedures to make that choice. Section 4 proposes axioms for agenda choice functions and analyses to what extent our proposed procedures satisfy these. Before concluding, Section 5 reports on an experiment highlighting the need to balance computational and social choice-theoretic concerns when deciding on a function to choose an agenda.

2 Formal Framework

Next, we introduce our framework for multi-issue elections and the problem of selecting an agenda for such an election.

2.1 Multi-Issue Elections

Let $\mathcal{I} = \{1, \dots, p\}$ be a set of $p \in \mathbb{N}$ issues. Each issue $i \in \mathcal{I}$ is associated with a finite domain D_i (a set of possible values) with $|D_i| \geq 2$. The Cartesian product of our domains is a combinatorial domain $\mathcal{D} = D_1 \times \dots \times D_p$. For any subset $I \subseteq \mathcal{I}$, we write $\mathcal{D}[I]$ as a shorthand for the Cartesian product of the corresponding domains $\times_{i \in I} D_i$.

Let $\mathcal{N} = \{1, \dots, n\}$ be a set of $n \in \mathbb{N}$ voters (or agents). Each voter $i \in \mathcal{N}$ is equipped with a preference relation \succeq_i on \mathcal{D} . We take each \succeq_i to be a preorder, i.e., a binary relation that is reflexive and transitive. For two alternatives $x, y \in \mathcal{D}$, $x \succeq_i y$ means that voter i weakly prefers x over y . We write $x \succ_i y$ (strict preference) if $x \succeq_i y$ but not $y \succeq_i x$. If both $x \succeq_i y$ and $y \succeq_i x$, then i is indifferent between x and y . If neither $x \succeq_i y$ nor $y \succeq_i x$, then i is unable to rank them. Let $\text{Pre}(\mathcal{D})$ be the set of all preorders on \mathcal{D} .

Remark 1 While most work in social choice theory takes preferences to be linear or weak orders [Taylor, 2005], for applications in AI it has been argued that it is appropriate to work with more general structures, such as preorders, as agents may lack the cognitive and computational resources to fully rank all alternatives and as eliciting all relative rankings may be costly and unnecessary [Pini *et al.*, 2008].

Definition 2 A voting procedure for voters \mathcal{N} and domain \mathcal{D} is a function $V : \text{Pre}(\mathcal{D})^{\mathcal{N}} \rightarrow 2^{\mathcal{D}} \setminus \{\emptyset\}$.

That is, V maps profiles of preorders over \mathcal{D} supplied by the voters in \mathcal{N} to nonempty sets of winning alternatives in \mathcal{D} . (This is the standard definition of a voting correspondence, except that we allow for preorders in the input [Taylor, 2005].)

2.2 Preferential Dependence

Next we define what it means for one issue to be *preferentially dependent* on another issue, given a particular preference relation \succeq . But let us first consider an example.

Example 3 This example is due to Lang and Xia [2009]. Suppose the residents of a town need to collectively decide whether to build a swimming pool (S) or a tennis court (T). If a voter's preference is $ST \succ \bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T}$, then her preferences for one issue are not affected by the choice made for the other (e.g., she always prefers having the swimming pool). But if her

preferences are $S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$ (she wants a swimming pool only in case building the tennis court is rejected), then we say that she exhibits a preferential dependence.

We now define issue i to be preferentially dependent on issue j , given some preference relation \succeq , if there exists a situation where knowing the values chosen for all domains other than D_j is not sufficient to decide which of two given values in D_i should be (weakly) preferred according to \succeq .

Definition 4 We say that issue $i \in \mathcal{I}$ is *preferentially dependent* on issue $j \in \mathcal{I}$ given preference relation \succeq , if there exist values $x, x' \in D_i$, $y, y' \in D_j$, and a vector of values $\vec{z} \in \mathcal{D}[\mathcal{I} \setminus \{i, j\}]$ for the remaining domains such that $x.y.\vec{z} \succeq x'.y.\vec{z}$ but $x.y'.\vec{z} \not\succeq x'.y'.\vec{z}$.

Preferential dependence induces an irreflexive directed graph on \mathcal{I} , with an edge from i to j whenever j depends on i .

Definition 5 A *dependency graph* is an irreflexive directed graph on \mathcal{I} . The set of all such graphs is $\text{DG}(\mathcal{I})$.

Remark 6 Dependency graphs are reminiscent of CP-nets [Boutilier *et al.*, 2004], a language for compact preference representation. Note that here we do not introduce any specific language. Instead, we consider the class of all possible preorders (CP-nets can only express a fraction of these).

2.3 Choosing an Agenda

Suppose our voters want to choose an alternative from \mathcal{D} . In principle, standard voting procedures, adapted for use with preorders [Endriss *et al.*, 2009], could be used, but in practice this will be computationally infeasible. Instead, we want to partition \mathcal{D} into several smaller domains and run a sequence of elections for these smaller domains. Which partition and which sequence, i.e., which *agenda*, should we choose?

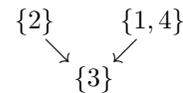
Definition 7 An *agenda* for the issues in \mathcal{I} is a linear order on a partition of \mathcal{I} . The set of all agendas on \mathcal{I} is $\text{AG}(\mathcal{I})$.

If $I \subseteq \mathcal{I}$ is one of the subsets in the partition, then there will be a *local election* to choose an alternative from the domain $\mathcal{D}[I]$; that is, all the issues in I will be voted on at the same time. The linear order defined over the partition determines the order in which these local elections are held.

Definition 8 A *meta-agenda* for \mathcal{I} is an acyclic graph on a partition of \mathcal{I} . The set of all meta-agendas on \mathcal{I} is $\text{MAG}(\mathcal{I})$.

Sometimes it will be unnecessary to specify a fixed ordering for two local elections (for instance, if none of the issues in the first depend on any in the second, and *vice versa*, for any of the voters). In such a case it suffices to specify a meta-agenda representing an entire class of agendas, namely all linear orders extending that meta-agenda.

Example 9 The following is an example for a meta-agenda on four issues $\mathcal{I} = \{1, 2, 3, 4\}$:



This meta-agenda represents two agendas: (1) first run an election on issue 2, then run an election on combinations of issues 1 and 4, and finally run an election on issue 3; (2) first vote on 1 and 4, then on 2, and finally on 3.

Our main object of study in this paper are functions that take as input the dependency graphs over issues of the individual agents and return a meta-agenda (or, more precisely, a nonempty set of meta-agendas, to account for possible ties):

Definition 10 A *meta-agenda choice function* (MACF) is a function $F : \text{DG}(\mathcal{I})^{\mathcal{N}} \rightarrow 2^{\text{MAG}(\mathcal{I})} \setminus \{\emptyset\}$.

2.4 Designing a Voting Procedure

Before we turn to the study of MACFs, the main topic of this paper, let us briefly sketch how we can design a voting procedure for electing an element of \mathcal{D} using a particular MACF. Such a voting procedure will have three components:

- (1) a MACF that takes the (reported) dependency graphs of the voters as input and returns a set of meta-agendas S ;
- (2) an *agenda selection function* $f_{\text{AG}} : 2^{\text{MAG}(\mathcal{I})} \setminus \{\emptyset\} \rightarrow \text{AG}(\mathcal{I})$ that selects an agenda A from those represented by the meta-agendas in S ; and
- (3) a *voting procedure selection function* mapping any possible subset of \mathcal{I} to a voting procedure for that subset. This function specifies which voting procedure should be used for each of the local elections in A .

The agenda selection function plays a similar role as a tie-breaking rule in standard voting theory. The voting procedure selection function can, in principle, specify a different local procedure for each possible combination of issues. A more natural choice, however, is to use a single voting procedure (e.g., the Borda or the plurality rule [Taylor, 2005]) for all local elections; or to specify a function that, say, selects the Borda rule for small elections (e.g., four issues or fewer) and the plurality rule otherwise. For simplicity, let us assume that all local voting procedures are *resolute* (they always return a unique winner, i.e., they incorporate a tie-breaking rule).

We now have a practical voting procedure: the MACF together with f_{AG} determines an agenda, and then, following that agenda, local elections are run using the selected local voting procedure(s). After each local election, the voters are informed about the issues that have been settled.

3 Meta-Agenda Choice Functions

In this section we define several concrete meta-agenda choice functions. We do this by defining aggregation procedures on dependency graphs; the resulting graphs can then be transformed into meta-agendas by means of a process known as *graph condensation*. We also discuss some of the properties of the MACFs proposed and of the agendas they generate.

3.1 Preliminaries

As a first proposal, consider the following simple idea:

Example 11 To aggregate a profile of dependency graphs (G_1, \dots, G_n) into a meta-agenda, we construct a collective dependency graph G^* . Consider each (ordered) pair of issues (i, j) in turn, and include an edge from i to j in G^* iff a majority of agents claims that j depends on i . G^* determines a meta-agenda: partition G^* into its maximal strongly connected components and link components x and y iff there is an edge in G^* from one of the nodes in x to one of those in y . We call this method *edgewise majority choice*.

Transforming a graph G^* on \mathcal{I} into an acyclic graph on a partition of \mathcal{I} (i.e., into a meta-agenda) is known as *condensation* in graph theory. All procedures we will define are *graph aggregation procedures* $F : \text{DG}(\mathcal{I})^{\mathcal{N}} \rightarrow \text{DG}(\mathcal{I})$ that have to be combined with condensation to obtain a MACF.

We will assess the quality of agendas in terms of the *complexity* they introduce (measured in terms of the number of issues involved in a local election) and the extent to which they generate *uncertainty* amongst the voters as to how they should vote in a local election over an issue that depends on other issues that have not yet been decided.

Example 12 Suppose three agents each have linearly ordered dependencies over three issues: $a \rightarrow_1 b \rightarrow_1 c$, $b \rightarrow_2 c \rightarrow_2 a$, and $c \rightarrow_3 a \rightarrow_3 b$. If we use edgewise majority choice to find an agenda, then the result is an election with maximal complexity: all three issues need to be decided together.

Example 13 Suppose three agents have only one dependency each: $a \rightarrow_1 b$, $b \rightarrow_2 c$, and $a \rightarrow_3 c$. Edgewise majority choice suggests to vote separately on each issue. If the selected order is $c \rightarrow b \rightarrow a$, then each agent will face a local election where she will be uncertain about her preferences.

The notion of uncertainty can be made precise in several ways. One option is to use a *dependency violation measure*.

Definition 14 Let G^* and G_i be dependency graphs. We say that G_i has a **dependency violation** in G^* iff $(x, y) \in G_i$ and $(x, y) \notin G^*$ for some $(x, y) \in \mathcal{I} \times \mathcal{I}$.

Definition 15 A **dependency violation measure** is a function $H : \text{DG}(\mathcal{I})^{\mathcal{N}} \times \text{DG}(\mathcal{I}) \rightarrow \mathbb{N}$ with $H(G_1, \dots, G_n, G^*) = 0$ iff no G_i , for $i \leq n$, has a dependency violation in G^* .

Examples for choices of H include the sum of violations between the G_i s and G^* , the number of agent/election pairs where the agent experiences at least one uncertainty, the maximal number of violations experienced by any one agent, the number of agents experiencing some violation, etc.

3.2 Quota-Based Choice Functions

One way to balance complexity and uncertainty is to generalise edgewise majority voting by allowing for arbitrary quotas:

Definition 16 *Edgewise choice with quota q* , for $q \in [0, 1]$, is the graph aggregation procedure mapping (G_1, \dots, G_n) to

$$G^* = \{(x, y) \in \mathcal{I}^2 : |\{i : (x, y) \in G_i\}| \geq q \cdot n\}.$$

That is, under this rule we will respect a dependency iff at least $q \cdot n$ agents express this dependency. By increasing the quota we can reduce the size of the biggest cycle, while by lowering it we can minimise the number of dependency violations.

3.3 Canonical Agendas

We now define a MACF that does not allow for dependency violations. We call it the *canonical* MACF: the output is a unique meta-agenda that merges all individual preferential dependencies without adding any new ones.

Definition 17 The **canonical** graph aggregation procedure is the function mapping each profile of dependency graphs (G_1, \dots, G_n) to the union graph $\bigcup_{i \in \mathcal{N}} G_i$.

In combination with graph condensation, this defines the *canonical MACF*. The following properties are easy to verify:

Proposition 1 *The canonical MACF is equivalent to the MACF based on edgewise choice with quota $\frac{1}{n}$.*

Proposition 2 *The canonical MACF minimises the size of the largest local election amongst all MACFs without dependency violations.*

Canonical agendas are not difficult to compute and we can easily design an algorithm that will run in quadratic time. If the size of the largest cycle is reasonably small, then this represents a good method of choosing voting agendas.

But how often will we get a canonical agenda with “small” local elections? Take an agenda A scheduling p single-issue elections in turn, and consider how many dependency graphs there are that are compatible with A . The k th issue may or may not depend on any number of the $k-1$ issues scheduled before it (but it must not depend on any of the issues scheduled after it). Thus, there are $\sum_{k=1}^p (k-1) = \frac{p(p-1)}{2}$ potential dependencies that may or may not be realised, i.e., there are $2^{\frac{p(p-1)}{2}}$ dependency graphs that are compatible with A . When the agenda may contain elections on combinations of at most two issues each, then eliciting the preferences still requires relatively little effort, but this increases the number of dependency graphs compatible with it:

Proposition 3 *An agenda with local elections of size at most 2 is compatible with exponentially more dependency graphs than an agenda with local elections of size 1.*

Proof. An agenda with one-issue elections is compatible with $2^{\frac{p(p-1)}{2}}$ graphs. If we allow for one election on a combination of 2 issues, then this doubles the number of compatible graphs (we can add one more edge). For p issues, we can schedule up to $\lfloor \frac{p}{2} \rfloor$ elections of size 2. Thus, by iterating the previous argument, an agenda where all (or all but one) elections have size 2 is compatible with $2^{\frac{p(p-1)}{2}} \cdot 2^{\lfloor \frac{p}{2} \rfloor}$ graphs. \square

Lang and Xia [2009] make a similar point and show that there are exponentially more *legal* (compatible with a linear agenda) preference profiles than there are profiles with *separable* (dependency-free) preferences.

3.4 Constraint-Based Choice Functions

Next we introduce a general class of MACFs balancing the size of local elections and the severeness of dependency violations.

Definition 18 *Given a dependency violation measure H , the **constraint-based** graph aggregation procedure $F_{k,\ell}^H$ is the function mapping each profile of dependency graphs (G_1, \dots, G_n) to the set of dependency graphs*

$$\{G^* : H(G_1, \dots, G_n, G^*) \leq k \text{ and } \text{mc}(G^*) \leq \ell\},$$

where $\text{mc}(G^*)$ is the size of the maximal cycle in G^* .

Note that while constraint-based procedures yield sets of graphs and quota-based procedures yield single graphs, both can be extended to MACFs via graph condensation. The MACF based on $F_{k,\ell}^H$ is the function returning all meta-agendas with maximal cluster size ℓ that result in an aggregated number of dependency violations of at most k , when

these violations are measured using H . That is, k relates to the parameter of uncertainty identified earlier, while ℓ directly corresponds to the parameter of complexity.

Constraint-based procedures will often be computationally intractable. For instance, if H is the maximal Hamming distance, then there is a simple reduction from the CLOSEST STRING problem, which is known to be NP-hard [Li *et al.*, 1999]. A full complexity analysis of the family of constraint-based procedures is an important project for future work.

In Definition 18 one of the two parameters (k and ℓ) can be kept fixed whilst performing an optimisation on the other. This is the case for the canonical MACF, where ℓ is optimised for $k := 0$. Another special case of this approach is the following, which we call *distance-based* MACFs. The bound on “computational badness” is irrelevant here (namely, the size of the maximal cluster is $\ell := n$).

Definition 19 *The **distance-based** graph aggregation procedure $F_{d,g}$ for distance metric $d : \text{DG}(\mathcal{I}) \times \text{DG}(\mathcal{I}) \rightarrow \mathbb{R}$ on dependency graphs and aggregation function $g : \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}$ is the function mapping each profile of dependency graphs (G_1, \dots, G_n) to the set of dependency graphs*

$$\text{argmin}_{G^* \in \text{DG}(\mathcal{I})} g(d(G_1, G^*), \dots, d(G_n, G^*)).$$

4 Axiomatic Analysis

Inspired by applications of the *axiomatic method* in social choice theory, particularly voting theory [Taylor, 2005] and judgment aggregation [List and Puppe, 2009], we now formulate a number of axioms reflecting basic desirable properties of MACFs, and we analyse to what extent some of the MACFs defined in Section 3 satisfy these axioms.

The first axiom postulates that the set of meta-agendas computed should be independent of the identity of the voters.

Definition 20 *A meta-agenda choice function F satisfies **anonymity** if $F(G_1, \dots, G_n) = F(G_{\sigma(1)}, \dots, G_{\sigma(n)})$ for any profile of dependency graphs (G_1, \dots, G_n) and any permutation $\sigma : \mathcal{N} \rightarrow \mathcal{N}$ of the set of voters.*

The next axiom is a form of neutrality, which we call *dependency-neutrality*. Given a meta-agenda M and two issues $a, b \in \mathcal{I}$, we say that M respects the (potential) dependency of b on a , if either (1) a and b are part of the same subset (i.e., will be decided upon in the same local election), or if (2) a belongs to a subset preceding the subset containing b (i.e., a will be decided upon in a local election that will take place before the local election involving b). Our neutrality axiom requires MACFs to be neutral wrt. these kinds of dependencies. It stipulates that if for two edges (a, b) and (a', b') it is the case that for each voter i either both (a, b) and (a', b') or neither of them occur in i 's dependency graph, then any meta-agenda returned by F should either respect both of the corresponding potential dependencies or neither of them.

Definition 21 *A meta-agenda choice function F satisfies **dependency-neutrality** if, for any profile of dependency graphs (G_1, \dots, G_n) and any issues $a, b, a', b' \in \mathcal{I}$, it is the case that whenever $(a, b) \in G_i$ iff $(a', b') \in G_i$ for all voters $i \in \mathcal{N}$ then any meta-agenda in $F(G_1, \dots, G_n)$ respects dependency (a, b) iff it respects dependency (a', b') .*

The third axiom we introduce is *reinforcement*. It is closely modelled on the standard reinforcement axiom familiar from voting theory and stipulates that if we aggregate two separate profiles of dependency graphs and the resulting sets of meta-agendas have a nonempty intersection, then aggregating the union of these two profiles should result in precisely the meta-agendas in that intersection. To express this axiom formally we need to index MACFs by the sets of voters they are used with: $F^{N \subseteq \mathcal{N}} : \text{DG}(\mathcal{I})^{N \subseteq \mathcal{N}} \rightarrow 2^{\text{MAG}(\mathcal{I})} \setminus \{\emptyset\}$. That is, F is now a family of MACFs, one for every possible electorate.

Definition 22 A meta-agenda choice function F satisfies *reinforcement* if $F^N(\vec{G}) \cap F^{N'}(\vec{G}') = S \neq \emptyset$ implies $F^{N \cup N'}(\vec{G} \cup \vec{G}') = S$ for any disjoint electorates $N, N' \subseteq \mathcal{N}$ and dependency graph profiles $\vec{G} \in \text{DG}(\mathcal{I})^N$ and $\vec{G}' \in \text{DG}(\mathcal{I})^{N'}$ for these electorates.

Some of our MACFs satisfy all three axioms:

Proposition 4 Any quota-based MACF (including the canonical MACF) satisfies anonymity, neutrality and reinforcement.

Proof. For lack of space, we only sketch the main ideas. Anonymity is immediately seen to be satisfied, as all of the functions covered are symmetric wrt. dependency graphs in the input. Dependency-neutrality is satisfied by virtue of the fact that the procedures concerned decide on which dependencies to respect one by one, always using the same criterion for inclusion. To see that reinforcement is satisfied, first observe that a quota-based MACF always returns a single meta-agenda; the claim then follows easily from the definitions. \square

In our context, reinforcement is in fact a very weak property. However it is *not* satisfied by all procedures. For instance, constraint-based procedures will typically not satisfy it, because a bound on the number of violations can be much more easily respected in a small electorate than in a large one.

Finally, we state two results for distance-based MACFs $F_{d,g}$ that relate the properties of the distance d and the aggregator g to the corresponding MACF. Elkind *et al.* [2010] prove similar results in the context of voting procedures defined in terms of consensus profiles and distances between profiles. For the next result, we require the notion of a *symmetric* aggregator $g : \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}$. An aggregator g is symmetric if for every permutation $\sigma : \mathcal{N} \rightarrow \mathcal{N}$ it is the case that $g(x_1, \dots, x_n) = g(x_{\sigma(1)}, \dots, x_{\sigma(n)})$ for all $(x_1, \dots, x_n) \in \mathbb{R}^{\mathcal{N}}$.

Proposition 5 If g is a symmetric aggregator, then the MACF based on $F_{d,g}$ is anonymous for any distance d .

We omit the simple proof. For the next result, we need to define what constitutes a *neutral* distance between graphs. Let $\sigma : \mathcal{I} \rightarrow \mathcal{I}$ be a permutation of the vertices of the graph G . We can extend σ to G in the obvious manner by defining the edges of $\sigma(G)$ as follows: $(a, b) \in E_{\sigma(G)}$ if and only if $(\sigma^{-1}(a), \sigma^{-1}(b)) \in E_G$. A distance d defined on graphs is neutral if $d(G, G') = d(\sigma(G), \sigma(G'))$ for $G, G' \in \text{DG}(\mathcal{I})$.

Proposition 6 If d is a neutral distance, then the MACF based on $F_{d,g}$ is dependency-neutral for any aggregator g .

The proof is tedious but conceptually not difficult.

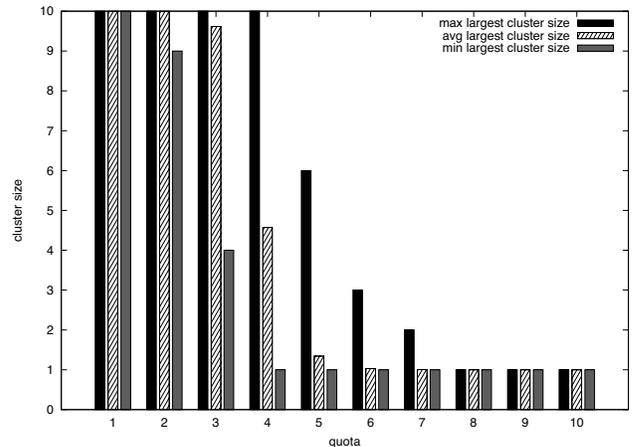


Figure 1: Largest cluster size for edgewise voting with 10 agents, 10 issues, and a full range of quotas

5 Experiments

While the canonical MACF rules out dependency violations, it will often result in local elections that involve too many issues to be feasible in practice. Constraint-based MACFs do allow us to balance computational and social choice-theoretic needs, but they are (typically) computationally intractable. This leaves the quota-based edgewise choice functions, which are appealing due to their conceptual and computational simplicity, but which do not give us direct control over either the size of local elections or the number of dependency violations. In this section we report on an experimental study of edgewise choice functions in view of these two criteria.

To generate an input profile, we fix a graph G_0 from which we generate n perturbed copies (one for each agent). For each copy, we delete each edge in G_0 with probability r_1 , and we add each edge not present in G_0 with probability r_2 . For our experiments, G_0 is a graph where each of the p issues is involved in exactly one dependency relation and $r_1 = r_2 = 0.2$. Thus, our agents have sparse dependency graphs that differ, but not too dramatically.

For each quota¹ from 1 to 10, we generated 1000 10-agent, 10-issue instances, and computed the resulting meta-agendas. Figure 1 shows the minimum, average, and maximum largest cluster size found in these meta-agendas. (For example, with a quota of 4 agents, the largest cluster in the meta-agendas contained an average of 4.57 issues.) Figure 2 shows, for the same instances, the number of dependency violations according to two measures: (1) the *agent-issue* violations, i.e., the sum of the dependency violations over all voters, and (2) the *agent-election* violations, i.e., the number of pairs of an agent i and a local election E such that, according to i , an issue in E depends on an issue to be decided in a later election.

Edgewise choice with quotas 1 and 10, respectively, generate the union and intersection graphs of the profiles; hence, the meta-agendas for quota 1 are monolithic while for quota 10 they are atomic. High quotas will produce more violations

¹In this section only, “quota” is the number of agents required to accept a dependency (i.e., this is $q \cdot n$ in Definition 16).

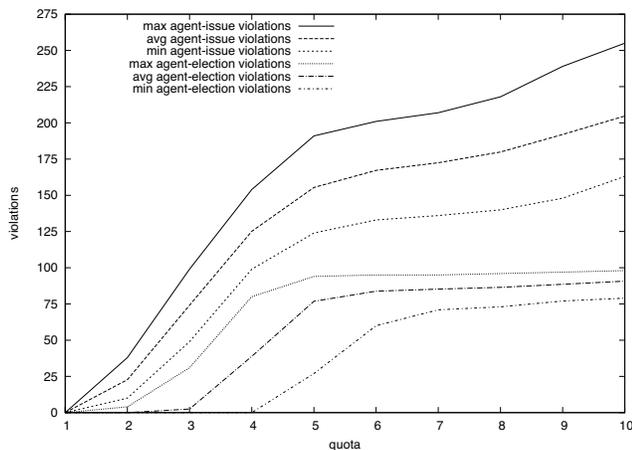


Figure 2: Number of violations for edgewise voting with 10 agents, 10 issues, and a full range of quotas

than lower quotas; a quota of 1 will produce no violations at all. Note the dramatic change which occurs over quotas 3–5. When the quota increases from 3 to 4, the average cluster size halves, and (more than) halves again from 4 to 5. The minimum cluster size drops precipitously over the same range. At the same time, the rate of increase in number of violations begins to decline around quota 4. With 10 agents, a quota of 4 gives a good balance between low maximum cluster size—which limits the number of alternatives in any local election—and low numbers of dependency violations.

6 Conclusions

Voting in combinatorial domains is a highly challenging problem, with no general solution in sight. In this paper, we have introduced the notion of a *meta-agenda choice function* as a manner of isolating the choice of a suitable agenda, given only the preferential dependency information provided by the voters, from the rest of the problem.

We have introduced several procedures for making this choice in practice, ranging from the canonical procedure that does not violate any preferential dependencies, over constraint-based procedures in which violations of dependencies and the size of local elections can be balanced explicitly, to the conceptually simple quota-based edgewise choice functions, in which a dependency is respected only if a certain number of voters ask for it to be respected. We have also formulated a number of simple axioms for meta-agenda choice functions, suggesting that this aspect of the more general problem of voting in combinatorial domains is in itself a candidate for deeper axiomatic analysis, and we have established some basic results regarding the satisfaction of these axioms by our procedures. Finally, we have reported on an experimental study of the class of quota-based procedures as a means of exemplifying the dilemma faced by a designer of a voting procedure for multi-issue elections when having to balance computational and social choice-theoretic considerations.

We believe that the main contribution of this paper is of a conceptual nature: to isolate the problem of agenda choice as

an important subtask in mastering the challenges of voting in combinatorial domains and to provide a technical framework for studying this subtask. Of course, this raises more questions than it does answer, and there are numerous directions for future work that should be explored. We only mention three of the most pressing ones here. First, it will be important to understand how the properties of the (meta-)agenda choice function and the properties of the local voting procedures together determine the properties of the overall voting procedure. Second, there are a number of different assumptions that we could make regarding the “attitudes” of voters when faced with a local election for which they are uncertain about their preferences (e.g., they might be optimists, assuming that the remaining issues will be decided in their favour). The question then arises how these attitudes influence election outcomes. Third, we require a better understanding of the reduction in elicitation complexity achieved by first eliciting only preferential dependencies and then votes on small domains rather than directly eliciting a vote on the full combinatorial domain.

References

- [Boutilier *et al.*, 2004] C. Boutilier, R.I. Brafman, C. Domshlak, H.H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
- [Brams *et al.*, 1998] S.J. Brams, D.M. Kilgour, and W.S. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15(2):211–236, 1998.
- [Chevalere *et al.*, 2008] Y. Chevalere, U. Endriss, J. Lang, and N. Maudet. Preference handling in combinatorial domains: From AI to social choice. *AI Magazine*, 29(4):37–46, 2008.
- [Elkind *et al.*, 2010] E. Elkind, P. Faliszewski, and A.M. Slinko. On the role of distances in defining voting rules. In *Proc. 9th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010)*, 2010.
- [Endriss *et al.*, 2009] U. Endriss, M.S. Pini, F. Rossi, and K.B. Venable. Preference aggregation over restricted ballot languages: Sincerity and strategy-proofness. In *Proc. 21st International Joint Conference on AI (IJCAI-2009)*, 2009.
- [Lacy and Niou, 2000] D. Lacy and E.M.S. Niou. A problem with referendums. *Journal of Theoretical Politics*, 12(1):5–31, 2000.
- [Lang and Xia, 2009] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57:304–324, 2009.
- [Li *et al.*, 1999] M. Li, B. Ma, and L. Wang. Finding similar regions in many strings. In *Proc. 31st Annual ACM Symposium on Theory of Computing (STOC-1999)*, 1999.
- [Li *et al.*, 2010] M. Li, Q.B. Vo, and R. Kowalczyk. An efficient procedure for collective decision-making with CP-nets. In *Proc. 19th European Conference on AI (ECAI-2010)*, 2010.
- [List and Puppe, 2009] C. List and C. Puppe. Judgment aggregation: A survey. In *Handbook of Rational and Social Choice*. Oxford University Press, 2009.
- [Pini *et al.*, 2008] M.S. Pini, F. Rossi, K.B. Venable, and T. Walsh. Aggregating partially ordered preferences. *Journal of Logic and Computation*, 19(3):475–502, 2008.
- [Rossi *et al.*, 2004] F. Rossi, K.B. Venable, and T. Walsh. mCP Nets: Representing and reasoning with preferences of multiple agents. In *Proc. 19th AAAI Conference on AI (AAAI-2004)*, 2004.
- [Taylor, 2005] A.D. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005.