AUTHOR QUERY FORM

ELSEVIER	**Journal:** ARTINT **Article Number:** 3098	**Please e-mail your responses and any corrections to:** **E-mail:** **corrections.esch@elsevier.vtex.lt**

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof. **It is crucial that you NOT make direct edits to the PDF using the editing tools as doing so could lead us to overlook your desired changes.** Rather, please request corrections by using the tools in the Comment pane to annotate the PDF and call out the changes you would like to see. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult *http://www.elsevier.com/artworkinstructions*

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof.

Location in article	**Query / Remark: Click on the Q link to find the query's location in text** **Please insert your reply or correction at the corresponding line in the proof**
Q1	Your article is registered as a regular item and is being processed for inclusion in a regular issue of the journal. If this is NOT correct and your article belongs to a Special Issue/Collection please contact <p.salma@elsevier.com> immediately prior to returning your corrections. (p. 1/ line 1)
Q2	The author names have been tagged as given names and surnames (surnames are highlighted in teal color). Please confirm if they have been identified correctly and are presented in the desired order. (p. 1/ line 15)
Q3	Please indicate which author(s) should be marked as 'Corresponding author'. (p. 1/ line 16)
Q4	Please check if email addresses are OK. (p. 1/ line 58)
Q5	Please clarify the meaning of the sentence "...restriction is consists in counting ...". (p. 23/ line 51)
Q6	Ref. [25]: please supply more publication details (vol. no., issue no., publication year, page range, doi number). (p. 26/ line 39)
Q7	Please check if sponsor names have been identified correctly and correct if necessary. (p. 28/ line 1)
	Please check this box or indicate your approval if you have no corrections to make to the PDF file ☐

Artificial Intelligence ••• (••••) •••–•••

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Voting on multi-issue domains with conditionally lexicographic preferences ☆

Jérôme Lang [a], Jérôme Mengin [b], Lirong Xia [c]

[a] LAMSADE, CNRS – Université Paris-Dauphine, France
[b] IRIT, Université de Toulouse, CNRS, Toulouse, France
[c] Rensslaer Polytechnic Institute, USA

ARTICLE INFO

Article history:
Received 10 November 2016
Received in revised form 17 May 2018
Accepted 21 May 2018
Available online xxxx

Keywords:
Computational social choice
Voting
Winner determination
Lexicographic preferences
Maximum satisfiability

ABSTRACT

One approach to voting on several interrelated issues consists in using a language for compact preference representation, from which the voters' preferences are elicited and aggregated. Such a language can usually be seen as a domain restriction. We consider a well-known restriction, namely, *conditionally lexicographic preferences*, where both the relative importance between issues and the preference between the values of an issue may depend on the values taken by more important issues. The naturally associated language consists in describing conditional importance and conditional preference by trees together with conditional preference tables. In this paper, we study the aggregation of conditionally lexicographic preferences for several common voting rules and several classes of lexicographic preferences. We address the computation of the winning alternative for some important rules, both by identifying the computational complexity of the relevant problems and by showing that for several of them, computing the winner reduces in a very natural way to a MAXSAT problem.

1. Introduction

There are many situations where a group of agents has to make a common decision in *multi-issue domains*, i.e. a set of possibly interrelated binary issues. A typical example of such a situation is that of *multiple referenda*: there is a set of binary issues (such as building a sport centre, building a cultural centre etc.), and the group must make a yes/no decision on each of them [1]. Another example is *group product configuration*, where the group must agree on a complex object consisting of several components, each taking one of two possible values.

Voting on several interrelated issues is a challenging problem. If the agents vote separately on each issue, then paradoxes generally arise [1,2]. Such paradoxes rule out this 'decompositional' approach, except in the very restricted case when voters have separable preferences. A second way consists in using a sequential voting protocol: issues are considered one after another, in a predefined order, and the voters know the assignment to the earlier variables before expressing their preferences on later ones (see, e.g., [3–5]). This method, however, works reasonably well only if we can guarantee that there exists a common order over issues such that every agent can express her preferences unambiguously on the values of each issue at the time she is asked to report them.

☆ This paper is an invited revision of a paper which first appeared at the 18th International Conference on Principles and Practice of Constraint Programming, 2012.

E-mail addresses: lang@lamsade.dauphine.fr (J. Lang), Jerome.Mengin@irit.fr (J. Mengin), xialirong@gmail.com (L. Xia).

https://doi.org/10.1016/j.artint.2018.05.004
0004-3702/© 2018 Published by Elsevier B.V.

A third class of methods consists in using a language in which the agents can express their preferences over the space of combined alternatives for all issues simultaneously. If the language is expressive enough to allow for expressing any possible preference relation, then the aforementioned paradoxes are avoided. However, this expressivity implies significant elicitation and computation costs: in a multi-issue domain, the number of alternatives is exponential in the number of issues, therefore the number of possible preference relations is doubly exponential and whichever language we use to represent them, the input will be exponential in the worst case. One way around this problem is to use some compact preference representation language; this comes at the cost of some domain restriction, depending on the chosen language.

Therefore, when aggregating preferences on multiple interrelated issues, a choice, or a trade-off, must be made between (a) being prone to severe paradoxes, (b) requiring a heavy communication and computation burden or (c) imposing a domain restriction.

In this paper, we explore a particular domain restriction by focusing on *conditionally lexicographic preferences*. Standard lexicographic preferences are a simple model, where alternatives are first sorted according to one issue, considered to be the most *important* one; alternatives that have equal values for that issue are then sorted according to the second most important issue, and so on until the set of alternatives is totally ordered. The psychology literature shows evidence that lexicographic preferences are often an accurate model for human decisions [6,7]. Although very intuitive, this model is quite restrictive. However, it can be extended by allowing the preferences over the values of an issue, as well as the relative importance of two issues, to depend on the values of more important issues. The relative importance of issues is no longer a linear ordering, but a tree, with the most important issue at its root. A preference relation is *conditionally lexicographic* if it is defined by such a lexicographic preference tree. To the best of our knowledge, conditionally lexicographic preference trees were first introduced by Fraser [8,9], and have been considered in individual decision making (and especially in constraint satisfaction problems) by Wilson [10,11] and Wallace and Wilson [12], as well as, from the learning point of view, by Booth et al. [13], Brauning et al. [14,15] and by Liu and Truszczyński [16]. Note that Wilson [11] and Brauning et al. [14,15] propose to extend further the lexicographic preference model by allowing several variables to appear at the same importance level/node in the tree; we do not consider such trees in this paper.

Lexicographic preferences appear to be a reasonable assumption in many domains (see the work of Taylor [17] for a discussion on the plausibility of lexicographic preferences in various political contexts). However, it implies a strong domain restriction: for q binary variables, there are $(2^q)!$ preference relations and only 2^q (respectively $q! \times 2^q$) lexicographic preference relations if the importance order is fixed (respectively, not fixed). As explained above, domain restrictions are needed to escape strong paradoxes and significant communication cost, but the risk, by making a too strong domain restriction, is to be on the weak side regarding its plausibility, since only a tiny fraction of the whole set of preference relations falls in the class of lexicographic preference relations. (A similar problem occurs with separable preferences.) Conditionally lexicographic preference models still are a domain restriction, but a much weaker one than standard lexicographic preferences, as there are exponentially more conditionally lexicographic preferences than lexicographic preferences [13]. Therefore, this assumption can be seen as a reasonable trade-off between expressivity and complexity.

The aggregation of lexicographic preferences over combinatorial domains has received only little attention. Taylor [17] considers the aggregation of lexicographic preferences on a domain of two attributes with real-valued domains. Each voter has unconditional, single-peaked preferences on each of the two attribute domains. He shows that when voters all have the same importance order, there always exists a weak Condorcet winner, but that this is no longer the case if voters may diverge on the relative importance of the two attributes (and he characterizes situations where a weak Condorcet winner exists). Pattanaik [18] generalizes the former results on domains of more than two attributes, and establishes conditions for the existence of a socially best alternative for a wider class of voting rules including simple majority. Bhadury et al. [19] generalize Taylor's latter result to more than two attributes. Encarnacion [20] focuses on preference aggregation (with a preference relation as output) rather than voting: he also considers lexicographic preferences with a unconditional importance order, common to all agents, and single-peaked preferences on the domain of each variable, and shows that this domain restriction is Arrow-consistent.

More recently, Liu and Truszczyński [21] obtained results completing the results we obtained in our conference paper [22] (see Section 5 for more discussions) and used answer-set programming to encode and solve preference aggregation problems on combinatorial problems with conditionally lexicographic preferences (we will come back to this in more detail in Section 4).

Dividino et al. [23] make use of lexicographic preferences for preference aggregation in multi-attribute domains. A major difference with our work is that they use lexicographicity *for the preference aggregation phase* and not for defining each of the individual preferences. They start with a set of items (pieces of information on the web), each associated with a tuple of values corresponding to various criteria such as the time the item was posted, its source, etc. Each criterion corresponds to a weak order over items (for instance, for time, the more recent the better). Finally, given an importance order $\rhd$ on criteria, the global preference relation over items is defined lexicographically according to $\rhd$.

The generic problem of aggregating conditionally lexicographic preferences can be stated as follows. The set of alternatives is a multi-issue domain D composed of a finite set of binary issues.[1] We have a set of voters, each providing a

[1] The assumption that issues (or variables) are binary is made for the sake of simplicity, and because it holds in many practical domains, especially multiple referenda (see [24]). We discuss the extension of our results to non-binary domains in Subsection 9.3.

conditionally lexicographic preference relation over D under the compact and natural form of a lexicographic preference tree (LP-tree for short) [13], which we will define soon. Therefore, a (compactly represented) profile V consists of a collection of LP-trees. Finally, for a given voting rule r, we ask whether there is a simple way to compute the winner, namely $r(V)$, where 'simple' means that the winner should be computed efficiently and directly from V. This means that we must avoid producing the entire preference relations of every voter explicitly, which would require exponential space. For many cases where winner determination is computationally hard, we show that these problems can be efficiently converted to MAXSAT problems and can be solved by MAXSAT solvers.

The rest of the paper is organized as follows. Conditionally lexicographic preferences and their compact representation by LP-trees are defined and discussed in Section 2. In Section 3 we state the problem considered in this paper, namely the application of voting rules to profiles composed of conditionally lexicographic preferences. We will focus on three families of rules. First, k-approval rules in Section 4: we show that for many values of k, we can give a quite satisfactory answer to our question above, even for our most general models. Note that by 'satisfactory' we do not necessarily mean 'computable in polynomial time'. For instance, we will show that in some cases, there is a model-preserving translation between the winner determination problem and a maximum (weighted or unweighted) satisfiability problem. Since efficient MAXSAT solvers exist, in such cases we can consider preference aggregation as tractable to some extent. In Section 5 we focus on the Borda rule (and, to a lesser extent, to some other positional scoring rules sharing some properties with Borda): we show that winner determination can be solved in polynomial time for some of the simplest LP-tree models, and propose a translation into a weighted minimum satisfiability problem for some general models. We also provide a natural family of scoring rules for which similar translations exist. Then in Section 6 we consider the existence of a Condorcet winner, and show that even deciding whether a given alternative is a Condorcet winner is hard. In Section 7, we consider a particular setting where all voters have a common, possibly conditional, importance structure among issues, but can have varying, conditional preferences over the values of each issue. Section 8 is devoted to the specific case of LP-trees with fixed local preferences but possibly different structures. In Section 9 we discuss the application of our results to committee elections, as well as the importance of our domain restriction, and the extension of our results to variables with nonbinary domains. Finally, Section 10 concludes.

2. Conditionally lexicographic preferences and LP-trees

Let $\mathcal{I} = \{X_1, \ldots, X_q\}$ $(q \geq 2)$ be a set of *issues*, where each issue X_i takes a value in a binary *local domain* D_i, denoted as $\{0_i, 1_i\}$, or as $\{x_i, \overline{x_i}\}$, or, when there is no ambiguity, as $\{0, 1\}$. The set of alternatives is $D = D_1 \times \cdots \times D_q$, that is, an alternative is identified by its values on all issues. Alternatives are denoted by $\mathbf{d}$, $\mathbf{e}$, etc. For any $Y \subseteq \mathcal{I}$ we denote $D_Y = \prod_{X_i \in Y} D_i$. An element of D_Y is called a *partial alternative*. Let $L(D)$ denote the set of all linear orders over D.

We first give a high-level description of lexicographic preferences. Any two alternatives $\mathbf{d}$, $\mathbf{e}$ are compared by looking at the issues in sequence, according to their importance, until we reach an issue X such that the value of X in $\mathbf{d}$ is different from the value of X in $\mathbf{e}$. $\mathbf{d}$ and $\mathbf{e}$ are then ordered according to the *local preference* relation over the values of X. For such lexicographic preference relations we need both an *importance* relation between issues, and *local preference* relations over the domains of the issues. Both the importance between issues and the local preferences may be conditioned by the values of more important issues. Such lexicographic preference relations can be compactly represented by *Lexicographic Preference trees (LP-trees)* [13], formally defined in the upcoming subsection.

2.1. Lexicographic preference trees

An LP-tree $\mathcal{L}$ is composed of two parts:

1. A tree $\mathcal{T}$ where each node t is labelled by an issue, denoted by $\mathsf{Iss}(t) = X_i$, such that every issue appears once and only once on each branch; each non-leaf node either has two outgoing edges, labelled by the two values in the local domain (0_i and 1_i) respectively, or one outgoing edge, labelled by $\{0_i, 1_i\}$.
2. A *conditional preference table* $\mathsf{CPT}(t)$ for each node t, which is defined as follows. Let $\mathsf{Anc}(t)$ denote the set of issues labelling the ancestors of t. Let $\mathsf{Inst}(t)$ (respectively, $\mathsf{NonInst}(t)$) denote the set of issues in $\mathsf{Anc}(t)$ that have two (respectively, one) outgoing edge(s). There is a set $\mathsf{Par}(t) \subseteq \mathsf{NonInst}(t)$ such that $\mathsf{CPT}(t)$ is composed of the agent's local preferences over $D_{\mathsf{Iss}(t)}$ for all valuations of $\mathsf{Par}(t)$. That is, if $\mathsf{Iss}(t) = X_i$ then for every valuation $\mathbf{u}$ of $\mathsf{Par}(t)$, the CPT contains either $\mathbf{u} : 0_i \succ 1_i$ or $\mathbf{u} : 1_i \succ 0_i$.

An LP-tree $\mathcal{L}$ represents a linear order $\succ_{\mathcal{L}}$ over D as follows. Let X_i be the issue associated with the root of $\mathcal{L}$, and let $\succ_{X_i}$ denote the associated local preference relation. We denote by $\mathcal{L}_{X_i = x_i}$ (respectively $\mathcal{L}_{X_i = \overline{x_i}}$) the subtree of $\mathcal{L}$ corresponding to the branch $X_i = x_i$ (respectively $X_i = \overline{x_i}$). Let $\mathbf{d} = (d_1, \ldots, d_q)$ and $\mathbf{e} = (e_1, \ldots, e_q)$ be two different alternatives. Then $\mathbf{d} \succ_{\mathcal{L}} \mathbf{e}$ if either (1) $d_i \succ_{X_i} e_i$, or (2) $d_i = e_i$ and $\mathbf{d} \succ_{\mathcal{L}_{X_i = d_i}} \mathbf{e}$.

Moreover, to each alternative $\mathbf{d} \in D$ corresponds a single branch in the tree: at each node labelled with issue X_i with two outgoing edges, the branch corresponding to $\mathbf{d}$ follows the edge labelled with d_i. The *importance order* associated with $\mathbf{d}$ in $\mathcal{L}$, denoted by $\mathsf{IO}(\mathcal{L}, \mathbf{d})$, is the order in which issues appear along that branch. We use $\rhd$ to denote an importance order to distinguish it from agents' preferences $\succ$ over D. If in $\mathcal{L}$ each node has no more than one child, then all alternatives are associated with the same importance order $\rhd$, and we say that $\rhd$ is the importance order of $\mathcal{L}$.

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

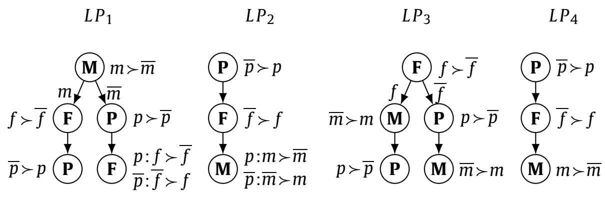

Fig. 1. Four LP-trees.

The *size* $|\mathcal{L}|$ of an LP-tree $\mathcal{L}$ is the number of its nodes plus the accumulated size of its conditional preference tables. Note that the size of $\mathcal{L}$ may be exponential in q even in the case of a linear structure.

Example 1. Suppose there are three binary issues to be decided among the inhabitants of a city.

1. Should the city build a metro?
2. Should the city centre be pedestrian?
3. Should there be a fee for cars entering the city?

The three issues are **M** (metro) with domain $\{m, \overline{m}\}$, **P** (pedestrian) with domain $\{p, \overline{p}\}$, and **F** (fee) with domain $\{f, \overline{f}\}$. There are four voters. Each voter has conditionally lexicographic preferences represented by the LP-trees LP_1, LP_2, LP_3 and LP_4 respectively as depicted in Fig. 1.

Consider the LP-tree LP_1. Let $\mathbf{P}_1$ be the node at the extremity of the left branch. We have $\mathsf{Iss}(\mathbf{P}_1) = \mathbf{P}$, $\mathsf{Anc}(\mathbf{P}_1) = \{\mathbf{M}, \mathbf{F}\}$, $\mathsf{Inst}(\mathbf{P}_1) = \{\mathbf{M}\}$, $\mathsf{NonInst}(\mathbf{P}_1) = \{\mathbf{F}\}$, and $\mathsf{Par}(\mathbf{P}_1) = \emptyset$ because the preferences over $\{p, \overline{p}\}$ do not depend on the value of **F**. On the other hand, if $\mathbf{F}_1$ is the node at the extremity of the right branch, then $\mathsf{Par}(\mathbf{F}_1) = \{\mathbf{P}\}$. The linear order represented by LP_1 is

$$mf\overline{p} \succ mfp \succ m\overline{f}\overline{p} \succ m\overline{f}p \succ \overline{m}pf \succ \overline{m}p\overline{f} \succ \overline{m}\overline{p}\overline{f} \succ \overline{m}\overline{p}f.$$

Moreover, $\mathsf{IO}(LP_1, mfp) = [\mathbf{M} \rhd \mathbf{F} \rhd \mathbf{P}]$ and $\mathsf{IO}(LP_1, \overline{m}\overline{f}\overline{p}) = [\mathbf{M} \rhd \mathbf{P} \rhd \mathbf{F}]$.

We say that an LP-tree has *unconditional* local preferences when the preference relation on the value of every issue is independent from the values of all other issues. In other words, each issue in the tree only contains unconditional preferences. This is the case for trees LP_3 and LP_4 but not for LP_1 nor LP_2. In LP_1, for instance, the preference between p and $\overline{p}$ depends on the value of **M**. We note that when the preferences are unconditional it is still possible that the importance relation be conditional, i.e. the tree may have branches.

Likewise, we say that the importance relation is *unconditional* when the order of the issues is the same in all branches of the tree (or, equivalently, when the tree has a single branch). This is the case of trees LP_2 and LP_4. Note that LP_4 has both unconditional importance and unconditional local preferences. See [13] for a discussion on various sub-classes of LP-trees.

A property that makes LP-trees appealing for social choice is that it is easy to compute the rank of alternatives:

Observation 1. Consider an alternative $\mathbf{d}$ and an LP-tree $\mathcal{L}$. For each issue X_i, let t_i be the node labelled with X_i in the branch of $\mathcal{L}$ corresponding to $\mathbf{d}$; we can define the level of X_i w.r.t. $\mathbf{d}$ and the local rank of $\mathbf{d}$ w.r.t. to X_i as follows:

- $\mathsf{level}(\mathcal{L}, \mathbf{d}, X_i)$ is the level of t_i in the tree, that is, the distance from t to the root plus one. The level of the root is 1 and the level of all leaves is q.
- $\Delta(\mathcal{L}, \mathbf{d}, X_i)$ is the local rank of the value of $\mathbf{d}$ for issue X_i in the local preference at t_i. Because issues are binary, $\Delta(\mathcal{L}, \mathbf{d}, X_i) = 0$ if $d_i \succ \overline{d_i}$ at node t_i (given the values of $\mathbf{d}$ for the issues that appear above t_i), and $\Delta(\mathcal{L}, \mathbf{d}, X_i) = 1$ otherwise.

Let $\mathsf{rank}(\mathcal{L}, \mathbf{d})$ be the rank of $\mathbf{d}$ in the linear order $\succ_{\mathcal{L}}$ defined by $\mathcal{L}$, then:

$$\mathsf{rank}(\mathcal{L}, \mathbf{d}) = 1 + \sum_{i=1}^{q} 2^{q - \mathsf{level}(\mathcal{L}, \mathbf{d}, X_i)} \Delta(\mathcal{L}, \mathbf{d}, X_i) \quad (1)$$

Example 2. Consider LP_1 defined in Example 1.

$$\mathsf{rank}(LP_1, mf\overline{p}) = 1 + 2^2 \times 0 + 2^1 \times 0 + 2^0 \times 0 = 1;$$
$$\mathsf{rank}(LP_1, \overline{m}f\overline{p}) = 1 + 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 1 = 8;$$
$$\mathsf{rank}(LP_1, m\overline{f}p) = 1 + 2^2 \times 0 + 2^1 \times 1 + 2^0 \times 1 = 4.$$

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

Note that rank$(\mathcal{L}, \mathbf{d})$ can be computed by a simple top-down traversal of $\mathcal{L}$ along the branch corresponding to $\mathbf{d}$, in time linear in the number of issues.

Conversely, Algorithm 1 computes the alternative with rank k in the linear order $\succ_{\mathcal{L}}$ corresponding to an LP-tree $\mathcal{L}$: the base 2 representation of $k-1$ is used to guide a top-down traversal of the tree from the root to a leaf. The running time is again linear in the number of issues.

Algorithm 1: $FindAlternative(\mathcal{L}, k)$.

```
Let k − 1 = (k_{q−1}...k_0)_2 and L* = L;
for i = q − 1 down to i = 0 do
    Let X_i be the root issue of L*. Let the local preferences be x_i ≻ x̄_i;
    if k_i = 1 then
        L* ← L*_{X_i = x_i} ; d_i ← x_i ;
    else
        L* ← L*_{X_i = x̄_i} ; d_i ← x̄_i ;
    end
end
return d.
```

3. Applying voting rules to LP-trees

A *(voting) profile* V over a set of alternatives D is a collection of n votes $V_1, \dots, V_n$, each being a linear order on D. An (irresolute) *voting rule* r maps every profile V to a nonempty subset of D: $r(V)$ is the set of *co-winners* for V under r. A *scoring function* S is a mapping from $L(D)^n \times D$ to $\mathbb{R}$. Often, a voting rule r is defined in such a way that $r(V)$ is the set of alternatives maximizing some scoring function S_r. In particular, a *positional scoring rule* is defined by a *scoring vector* $\mathbf{v} = (v(1), \dots, v(m))$, where m is the number of alternatives (in this paper $m = 2^q$): for any vote $V_i \in L(D)$ and any $c \in D$, let $S_{\mathbf{v}}(V, c) = v(\text{rank}_V(c))$, where $\text{rank}_{V_i}(c)$ is the rank of c in V_i. Then for any profile $V = (V_1 \dots, V_n)$, let $S_{\mathbf{v}}(V, c) = \sum_{j=1}^{n} S_{\mathbf{v}}(V_j, c)$. The winner is the alternative maximizing $S_{\mathbf{v}}(V, \cdot)$. In particular, the *k-approval* rule App_k (with $k \le m$), is defined by the scoring vector $v(1) = \dots = v(k) = 1$ and $v(k+1) = \dots = v(m) = 0$. Let S^k_{App} denote scoring vector for k-approval. The *Borda* rule is defined by the scoring vector $(m-1, m-2, \dots, 0)$. Let S_{Borda} denote the scoring vector for Borda, that is, $S_{\text{Borda}}(V_i, c) = m - \text{rank}_{V_i}(c)$.

Let $N_V(c, d)$ denote the number of votes in V that rank c ahead of d. An alternative c is the *Condorcet winner* for a profile V if for every $d \neq c$, a (strict) majority of votes in V prefers c to d, that is, if $N_V(c, d) > n/2$. If for every $d \neq c$ we have $N_V(c, d) \ge n/2$, with equality for at least one d, then c is a *weak Condorcet winner*.

3.1. Voting restricted to conditionally lexicographic preferences

Applying a voting rule to profiles consisting of arbitrary preferences on multi-issue domains is highly unpractical, because the specification of such preferences requires exponential space if no domain restriction is made. Does it become significantly easier when we restrict to conditionally lexicographic preferences? This is the key problem addressed in this paper. Of course the answer depends on the voting rule used.

A *conditionally lexicographic profile* is a collection of n conditionally lexicographic preferences over D. As conditionally lexicographic preferences are compactly represented by LP-trees, we define an *LP-profile* V as a collection of n LP-trees.

Given an LP-profile V and a voting rule r, a naive way of finding the co-winners would consist in determining the n linear orders induced by the LP-trees and then applying r to these linear orders. However, this would be very inefficient, both in space complexity and time complexity. Therefore, we would like to know if it is feasible, and efficient, to compute the winners directly from the LP-trees. More specifically, we ask the following questions: (a) given a voting rule, how difficult is it to compute the co-winners (or, else, one of the co-winners) for various classes of LP-trees? (b) for score-based rules, how difficult is it to compute the score of the co-winners? Formally, for each voting rule r defined as the maximizer of scoring function S, we consider the following decision and function problems.

EVALUATION (for r):

Input LP-profile V, integer number T
Question Does there exist an alternative $\mathbf{d}$ such that $S(V, \mathbf{d}) > T$?

WINNER (for r):

Input LP-profile V
Output $r(V)$

Note that if EVALUATION is NP-hard and the score of an alternative can be computed in polynomial time, then WINNER cannot be in P unless P = NP: if WINNER were in P, then EVALUATION could be solved in polynomial time by computing a winner and its score. For the voting rules studied in this paper, if not mentioned specifically, EVALUATION is w.r.t. the score functions we present when defining these rules.

3.2. LP-trees and propositional logic

Some of the positive results in the sequel are obtained by translating LP-trees into some particular types of logical formulas. We briefly recall here some basic notions of propositional logic and some satisfiability problems that will be used in the sequel.

Given a set of propositional symbols $\mathcal{P}$, logical formulas can be built with the usual connectives $\wedge$ (conjunction), $\vee$ (disjunction) and $\neg$ (negation). In several translations, we will use one propositional symbol for each issue, and so we will use the same notation for an issue X_i and the corresponding propositional symbol, so that the set of propositional symbols will often be $\mathcal{I}$ itself.

The *literals* are the elements of $\mathcal{P}$ and their negations; that is, the set of literals is $\{X, \neg X \mid X \in \mathcal{P}\}$. A *clause* has the general form $C = l_1 \vee l_2 \vee \ldots \vee l_{|C|}$, where each l_i is a literal. When all the literals, except possibly one, are negated propositions, it is a *Horn* clause. A dual of a clause is a *cube*, which is a conjunction of literals.

A *valuation* of $\mathcal{P}$ assigns a Boolean value 0 (false) or 1 (true) to each symbol in $\mathcal{P}$. A valuation satisfies a clause if it makes at least one literal true, and SAT is the problem of deciding if a given set of clauses is satisfiable, that is, if there is at least one valuation that satisfies all clauses; such a valuation is called a *model* of the set of clauses. SAT is an NP-complete problem.

The MAXSAT problem is a generalization of SAT: given a set of clauses, we want to find a valuation maximizing the number of satisfied clauses. In the PARTIAL WEIGHTED MAXSAT problem, some clauses have an associated positive weight, the others are left unweighted; the goal is to maximize the sum of the weights of the satisfied weighted clauses, while satisfying all the unweighted clauses. These are NP-complete problems, but the 2016 MAXSAT Evaluation, part of the 19th International Conference on Theory and Applications of Satisfiability Testing, shows that some solvers are able to solve industrial benchmarks in a few minutes, with up to hundreds of thousands of variables, and millions of clauses (of size 3).[2] Some of the most competitive solvers, at the time of writing, are described in e.g. [25–27]. In the GENERALIZED MAXSAT problem, the input is not restricted to clauses: given a set of propositional formulas, we want to find a valuation that satisfies as many of the formulas as possible.

WEIGHTED MINSAT is another variant of SAT: given a set of clauses, each associated with a positive cost, we want to find a valuation with minimum cost, where the cost of a valuation is the sum of the weights of the clauses that it satisfies. Although it is NP-complete, this problem too can be solved using efficient solvers; some work by translating these instances into WEIGHTED MAXSAT instances (e.g. [28,29]), whereas more recent ones use branch-and-bound techniques directly on the MINSAT instances (see, e.g. [30,31]). [30] reports on experiments where random MINSAT instances with one hundred variables and up to five times more clauses were solved in a few minutes. According to [31], the performances of MINSAT solvers seem to be comparable to that of MAXSAT solvers, some particular problems being more efficiently solved with one or the other.

4. *k*-Approval

Recall that it is possible to compute, in time linear in the number of issues, the k-th ranked issue of any LP tree for any k (cf. Algorithm 1). Therefore, given an integer $k < 2^q$ and an LP profile V, we can compute the top k alternatives of all LP-trees in V, and store them in a table together with their k-approval scores. As we have at most kn such alternatives, constructing the table takes time in $O(knq)$. Hence we have the following result.

Proposition 2. *For any constant k, given an LP-profile V with n voters, the k-approval co-winners for V can be computed in time in* $O(knq)$.

Example 3. Let V be the profile of Example 1. The winners for 1-approval (that is, plurality) are $mf\overline{p}, \overline{m}\overline{f}\overline{p}$, $\overline{m}fp$ and $m\overline{f}\overline{p}$ – their 1-approval score is 1. The winners for 2-approval are $\overline{m}\overline{f}\overline{p}$ and $m\overline{f}\overline{p}$, with a 2-approval score of 2.

A similar result also holds for computing the $(2^q - k)$-approval co-winners for any constant k. However, unless q is very small, there is little practical interest in using $(2^q - k)$-approval for a fixed (small) value of k, since in practice, we expect $kn \ll 2^q$, so that almost every alternative would be a co-winner.

We now focus on k-approval voting for values of k that are defined via *proportions* of the number 2^q of alternatives. Let $M \le q$ be a positive integer and define $\mathsf{Rat}(M)$ as the set of all rational numbers of the form $j/2^M$ for some positive integer M and $j \in \{1, \ldots 2^M - 1\}$. Equivalently, $\mathsf{Rat}(M)$ is the set of all numbers α in the interval $(0, 1)$ with a finite base 2 representation of length at most M. For any $\alpha \in \mathsf{Rat}(M)$, *$\alpha$-proportion-approval* is another name for the $\alpha 2^q$ approval rule.

[2] The 2016 MAXSAT Evaluation webpage is at the address http://www.maxsat.udl.cat/16/index.html, with links to result tables and benchmarks.

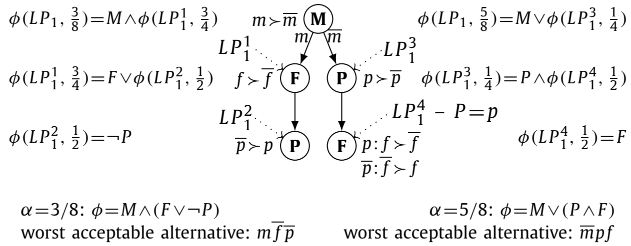

Fig. 2. LP tree LP_1 and its translations into formulas for 3-approval and 5-approval.

We first consider $\frac{1}{2}$-proportion-approval, or, equivalently, 2^{q-1}-approval. This rule requires to count the top 50% alternatives, which corresponds to the 'better' half of the tree. In this case, each voter only has to communicate her most important issue and its preferred value. Applying 2^{q-1}-approval here is therefore both intuitive and cheap in communication, and it turns out that the co-winners can be represented compactly and computed very easily.

Theorem 3. WINNER *for* 2^{q-1}*-approval and LP-profiles can be computed in time* $O(nq)$.

Proof. An alternative **d** is among the first half of alternatives in an LP-tree $\mathcal{L}_j$ if and only if the root issue of $\mathcal{L}_j$ is assigned to its preferred value. We build a table with the following $2q$ entries $\{1_1, 0_1, \ldots, 1_q, 0_q\}$: for every $\mathcal{L}_j$ we add 1 to the score of 1_i (resp. 0_i) if X_i is the root issue of $\mathcal{L}_j$ and the preferred value is 1_i (resp. 0_i). When this is done, for each X_i, we instantiate X_i to 1_i (resp. 0_i) if the score of 1_i is larger than the score of 0_i (resp. vice versa); if the scores are identical, we do not instantiate X_i. We end up with a partial instantiation, whose set of models (satisfying valuations) is exactly the set of co-winners. □

Example 4. Let V be the profile consisting of the 4 LP-trees of Example 1: **P** is the most important issue for two voters, both with preferred value $\overline{p}$; **M** is the most important issue for one voter with preferred value m, and **F** is the most important issue for one voter with preferred value f: therefore, the 4-approval winner is $mf\overline{p}$. If we add to this profile a fifth LP-tree with most important issue **M** and preferred value $\overline{m}$ then **M** will not be instantiated, and the two cowinners will be $mf\overline{p}$ and $\overline{m}f\overline{p}$.

While $\frac{1}{2}$-proportion-approval takes into account only the most important issue of each voter together with his preferred value, $\frac{1}{4}$- and $\frac{3}{4}$-proportion approval take into account the most important two issues of each voter; $\frac{1}{8}$-, $\frac{3}{8}$-, $\frac{5}{8}$- and $\frac{7}{8}$ take into account the most important three issues of each voter, and so on.

We now give a practical way to compute the α-proportion-approval co-winners, using a translation of the problem into an instance of GENERALIZED MAXSAT, with a one-to-one correspondence between the solutions of the voting problem and the solutions of its translation (which is usually referred to as a "model-preserving translation"). Using a reverse translation, we will then prove that, for any $\alpha \in \mathsf{Rat}(M) \setminus \{\frac{1}{2}\}$, α-proportion-approval is in fact NP-hard.

To illustrate the idea we first give the construction for two special cases: $\alpha = 1/4$ and $\alpha = 3/4$. Consider first the case $\alpha = 1/4$ and some LP-tree $\mathcal{L}_j$ whose top issue is X_{i_1} with preferred value $x_{i_1} \in \{0_{i_1}, 1_{i_1}\}$. Assume that the second most important issue given x_{i_1} is X_{i_2}, with preferred value x_{i_2}. Then, the $\frac{1}{4}2^q$ best alternatives are those that have precisely values x_{i_1} and x_{i_2} for X_{i_1} and X_{i_2} respectively. We can encode these alternatives by a formula $l_{i_1} \wedge l_{i_2}$, whose models are precisely the $\frac{1}{4}2^q$ best alternatives for $\mathcal{L}_j$: if $x_{i_1} = 1_{i_1}$ then we let $l_{i_1} = X_{i_1}$, and otherwise $l_{i_1} = \neg X_{i_1}$; l_{i_2} is defined similarly.

Consider now the case $\alpha = 3/4$ and some LP-tree $\mathcal{L}_j$ whose top issue is X_{i_1} with preferred value x_{i_1} and whose second most important issue *given* $\overline{x}_{i_1}$ is X_{i_2}, with preferred value x_{i_2} in this case: the $\frac{3}{4}2^q$ best alternatives are those for which $X_{i_1} = x_{i_1}$ and those for which $X_{i_1} = \overline{x}_{i_1}$ and $X_{i_2} = x_{i_2}$. These alternatives can be encoded as the models of the formula $l_{i_1} \vee l_{i_2}$, where l_{i_1} and l_{i_2} are the same as defined above for the case $\alpha = \frac{1}{4}$.

More generally, we define, for LP-tree $\mathcal{L}$ over a set $\mathcal{I}$ of q issues, and a fraction $\alpha \in \mathsf{Rat}(M)$, a formula $\phi(\mathcal{L}, \alpha)$ which characterizes the alternatives that are among the $\alpha 2^q$ best alternatives of $\mathcal{L}$. We note that, since $q \geq M$, $\alpha 2^q$ is an integer number. Let X_i be the root issue, let x_i^+ be the preferred value for X_i, and x_i^- be its less preferred value; let $l_i = X_i$ if $x_i^+ = 1_i$, and $l_i = \neg X_i$ if $x_i^+ = 0_i$, then:

$$\phi(\mathcal{L}, \alpha) = \begin{cases} l_i \wedge \phi(\mathcal{L}^+, 2\alpha) & \text{if } 0 < \alpha < 1/2 \\ l_i & \text{if } \alpha = 1/2 \\ l_i \vee \phi(\mathcal{L}^-, 2\alpha - 1) & \text{if } 1/2 < \alpha < 1 \end{cases}$$

where $\mathcal{L}^+$ (respectively $\mathcal{L}^-$) is the subtree of $\mathcal{L}$ corresponding to x_i^+ (respectively x_i^-), with the local preference tables being simplified by removing all conditional preferences where $X_i = x_i^-$ (respectively $X_i = x_i^+$).

The translations for LP tree LP_1 for two values of α are shown on Fig. 2.

Proposition 4. *Let $\alpha \in \mathrm{Rat}(M)$. Let V be an LP-profile over a set $\mathcal{I}$ of $q \geq M$ issues. Let $\Phi_V = \{\phi(\mathcal{L}_j, \alpha) \mid \mathcal{L}_j \in V\}$ be the multiset of formulae associated with V (note that the same formula can appear in several copies in Φ_V). Then the α-proportion-approval co-winners are the valuations (i.e., truth assignments) over $\mathcal{I}$ that maximize the number of formulas satisfied in Φ_V.*

Proof. The result follows from the fact that an alternative **d** is in the $\alpha 2^q$ best alternatives of some LP-tree $\mathcal{L}_j$ if and only if the corresponding valuation satisfies $\phi(\mathcal{L}_j, \alpha)$. □

In order to use state-of-the-art MAXSAT solvers to compute α-proportion-approval co-winners, we can easily transform Φ_V into a set of clauses of quadratic size: given some LP-tree $\mathcal{L}_j$, we introduce a new propositional variable L_j, and define a set of clauses Ξ_j in which every clause has the form $C \vee \neg L_j$, so that an interpretation can satisfy L_j and all the clauses in Ξ_j only if it satisfies all the constraints encoded in Ξ_j. The definition of Ξ_j is recursive, based on the formula $\phi(\mathcal{L}_j, \alpha)$, of the form $l_{i_1} \diamond_1 (l_{i_2} \diamond_2 (\ldots (l_{i_{M-1}} \diamond_{M-1} l_{i_M})\ldots))$:

- Ξ_j is initialized to $\{l_{i_M} \vee \neg L_j\}$;
- at the next stage, if $\diamond_{M-1}$ is a conjunction, then Ξ_j becomes $\{l_{i_{M-1}} \vee \neg L_j, l_{i_M} \vee \neg L_j\}$; if $\diamond_{M-1}$ is a disjunction, then Ξ_j becomes $\{l_{i_{M-1}} \vee l_{i_M} \vee \neg L_j\}$;
- more generally, at the $(M-k)$th stage, if $\diamond_{i_k}$ is a conjunction, then we add $l_{i_k} \vee \neg L_j$ to Ξ_j; otherwise, when $\diamond_{i_k}$ is a disjunction, then we add l_{i_k} as new disjunct to every clause in Ξ_j.

Example 5. For LP_1 and $\alpha = 3/8$, $\phi(LP_1, 3/8) = M \wedge (F \vee \neg P)$: the set of clauses is initialized to $\{\neg P \vee \neg L_1\}$; it then becomes $\{F \vee \neg P \vee \neg L_1\}$, and finally $\{M \vee \neg L_1, F \vee \neg P \vee \neg L_1\}$. For $\alpha = 5/8$, $\phi(LP_1, 5/8) = M \vee (P \wedge F)$ thus the set of clauses is initialized to $\{F \vee \neg L_1\}$, then becomes $\{P \vee \neg L_1, F \vee \neg L_1\}$, and finally $\{M \vee P \vee \neg L_1, M \vee F \vee \neg L_1\}$.

For an LP-profile V, we obtain a set of "hard" clauses $\Xi(V) = \bigcup_{\mathcal{L}_j \in V} \Xi_j$, and the PARTIAL WEIGHTED MAXSAT instance consists in finding valuations that satisfy all clauses in $\Xi(V)$, and a maximum number of the L_j's.

With this translation, the number of hard clauses for each voter is the number of 1's in α, and the size of the largest clause becomes one plus the number of 0's in α. It follows that the size of Ξ_j for each voter is in $O(M^2)$, and the translation, with n voters, is in $O(M^2 n)$; in terms of the parameters used in general to describe MAXSAT instances, the number of clauses is in $O(nM)$, their size is bounded by M, and the number of variables is q. Given the performances of the best current MAXSAT solvers, this approach may be applicable when there are hundreds of thousands of voters, for some $M \leq 10$.

We conclude this section by a proof that WINNER for α-proportion approval is NP-hard, which means that we may not expect much better than the above translation. In fact, our proof stems from the reverse translation. We will show that the model of a class of logical formulas can be encoded by α-proportion approval winners. Note that a formula $\phi(\mathcal{L}, \alpha)$ for some LP-tree $\mathcal{L}$ and some α always has the form

$$l_{i_1} \diamond_1 (l_{i_2} \diamond_2 (\ldots (l_{i_{M-1}} \diamond_{M-1} l_{i_M})\ldots)) \quad (2)$$

where the l_{i_j}'s are literals, each over a different variable, and where each $\diamond_i$ is a connective $\wedge$ or $\vee$, that only depends on the base 2 representation of α: let $\alpha = (0.\alpha_1 \ldots \alpha_M)_2$, then $\diamond_i$ is $\wedge$ if $\alpha_i = 0$, and $\diamond_i$ is $\vee$ if $\alpha_i = 1$. For any formula ϕ that has the form as in (2), we construct an LP-tree $\mathcal{L}(\phi)$ as follows:

- the importance structure is unconditional: $[X_{i_1} \triangleright X_{i_2} \triangleright \ldots \triangleright X_{i_{M-1}} \triangleright X_{i_M} \triangleright \text{Others}]$, where X_{i_j} is the issue corresponding to the literal l_{i_j}, and where 'Others' refers to all other issues.
- the local preference for issue X_{i_j} is unconditional and is $1_{i_j} \succ 0_{i_j}$ if $l_{i_j} = X_{i_j}$, or $0_{i_j} \succ 1_{i_j}$ if $l_{i_j} = \neg X_{i_j}$.

Then, given a set of formulas Φ having a common structure corresponding to some fraction $\alpha \in \mathrm{Rat}(M)$, $\alpha \neq 1/2$, we can define the profile $V = \{\mathcal{L}(\phi) \mid \phi \in \Phi\}$ of LP-trees with unconditional importance and unconditional local preferences: the α-proportion approval co-winners correspond to the valuations that satisfy a maximum number of formulas of Φ. We now prove that GENERALIZED MAXSAT remains NP-complete when restricted to this type of formulas.

Formally, given a set of propositional symbols $\mathcal{P}$ and a fraction $\alpha = (0.\alpha_1 \ldots \alpha_M)_2 \in \mathrm{Rat}(M)$, let $F(\mathcal{P}, \alpha)$ be the set of formulas of the form as in (2). We define MAX(α)SAT to be the problem that consists in finding a valuation satisfying a maximum number of formulas of a given set $\Phi \subseteq F(\mathcal{P}, \alpha)$, for a given set of propositional variables $\mathcal{P}$. In the rest of this section, we take $\mathcal{P} = \mathcal{I}$.

Lemma 5. *For any $\alpha \in \mathrm{Rat}(M) \setminus \{1/2\}$, MAX($\alpha$)SAT is NP-hard.*

Proof. By induction. The base cases are $\alpha = 3/4$ and $\alpha = 1/4$. Consider first $\alpha = 3/4 = (0.11)_2$: $F(\mathcal{P}, \frac{3}{4})$ is the set of 2-clauses (disjunctions of two literals). So MAX($\frac{3}{4}$)SAT is MAX2SAT, known to be NP-complete [32]. Consider next the case $\alpha = 1/4 = (0.01)_2$: $F(\mathcal{P}, \frac{1}{4})$ is the set of 2-cubes (conjunctions of two literals). Let $\Phi \subseteq F(\mathcal{P}, \frac{1}{4})$: then $\Phi' = \{\neg\phi \mid \phi \in \Phi\}$ is equivalent to a set of 2-clauses, and satisfying as many formulas of Φ as possible amounts to satisfying as few formulas of Φ' as possible. So MAX($\frac{1}{4}$)SAT is equivalent to MIN2SAT, known to be NP-complete too [33].

Let $\alpha = (0.\alpha_1 \ldots \alpha_M)_2 = \frac{1}{2}\alpha_1 + \frac{1}{4}\alpha_2 + \ldots + \frac{1}{2^M}\alpha_M$. Suppose first that $\alpha_1 = 0$. We give a polynomial reduction from MAX(2α)SAT to MAX(α)SAT. Note that $2\alpha = (\alpha_1.\alpha_2 \ldots \alpha_M)_2 = (0.\alpha_2 \ldots \alpha_M)_2$ has a base-2 representation of length strictly smaller than that of α. Let $\Phi \subseteq F(\mathcal{P}, 2\alpha)$, let $\mathcal{P}' = \mathcal{P} \cup \{X_0\}$, where X_0 is a new variable (not belonging to $\mathcal{P}$), and define $\Phi' = \{X_0 \wedge \phi \mid \phi \in \Phi\}$: then $\Phi' \subseteq F(\mathcal{P}', \alpha)$. Now, satisfying as many formulas of Φ as possible is equivalent to satisfying as many formulas of Φ' as possible, since only positive occurrences of X_0 appear in Φ'.

Finally, suppose that $\alpha_1 = 1$: we will prove that MAX($2\alpha - 1$)SAT can be reduced to MAX(α)SAT. Note that in this case $2\alpha - 1 = (0.\alpha_2 \ldots \alpha_M)_2$ has a base-2 representation of length strictly smaller than that of α. Let $\Phi \subseteq F(\mathcal{P}, 2\alpha - 1)$, let $\mathcal{P}' = \mathcal{P} \cup \{X_0\}$, where X_0 is a new variable – not in $\mathcal{P}$, and define $\Phi' = \{X_0 \vee \phi, \neg X_0 \vee \phi \mid \phi \in \Phi\} \subseteq F(\mathcal{P}', \alpha)$: Φ' is obtained from Φ by two copies of each $\phi \in \Phi$, and appending at the beginning a disjunction, one with X_0 and one with $\neg X_0$. Suppose that m is a valuation over $\mathcal{P}$ that satisfies μ formulas of Φ: let m' be an extension of m with some value for X_0, then m' satisfies $|\Phi| + \mu$ formulas of Φ'. Conversely, if a valuation m' over $\mathcal{P}'$ satisfies λ formulas of Φ', then $\lambda \geq |\Phi|$ because m' must satisfy one of X_0 and $\neg X_0$, and the restriction of m' to $\mathcal{P}$ satisfies $\lambda - |\Phi|$ formulas of Φ. So satisfying as many formulas of Φ as possible is equivalent to satisfying as many formulas of Φ' as possible.

A simple induction on the length of the base-2 representation of α then proves the result: the base cases correspond to a length 2 ($\alpha = 1/4$ or $3/4$). □

Theorem 6. *For any fraction $\alpha \in \mathsf{Rat}(M)$, $\alpha \neq 1/2$,* WINNER *for α-proportion-approval is* NP*-complete, even if all trees have unconditional importance and unconditional local preferences.*

Proof. We have seen that α-APPROVAL winner is equivalent to MAX(α)SAT, and we have just proved that the latter is NP-hard. The fact that these problems are in NP stems from the fact that GENERALIZED MAXSAT is in NP. □

5. Borda

In this section, we provide a translation of winner determination for the Borda rule into a WEIGHTED MINSAT problem. We also show that computing the Borda winner given an LP-profile is hard, except in the simple case where the preferences and the importance relations are unconditional.

Recall that the Borda score of an alternative $\mathbf{d}$ w.r.t. LP-tree $\mathcal{L}$ is $S_{\mathsf{Borda}}(\mathcal{L}, \mathbf{d}) = m - \mathsf{rank}(\mathcal{L}, \mathbf{d})$, with $m = 2^q$ and $\mathsf{rank}(\mathcal{L}, \mathbf{d}) = 1 + \sum_{i=1}^{q} 2^{q-\mathsf{level}(\mathcal{L}, \mathbf{d}, X_i)} \Delta(\mathcal{L}, \mathbf{d}, X_i)$. Hence, the Borda score of $\mathbf{d}$ for profile $V = (\mathcal{L}_1, \ldots, \mathcal{L}_n)$ is:

$$S_{\mathsf{Borda}}(V, \mathbf{d}) = \sum_{j=1}^{n} \left[2^q - 1 - \sum_{i=1}^{q} 2^{q-\mathsf{level}(\mathcal{L}_j, \mathbf{d}, X_i)} \Delta(\mathcal{L}_j, \mathbf{d}, X_i) \right]$$
$$= \sum_{j=1}^{n} \sum_{i=1}^{q} 2^{q-\mathsf{level}(\mathcal{L}_j, \mathbf{d}, X_i)} (1 - \Delta(\mathcal{L}_j, \mathbf{d}, X_i)).$$

The Borda winner is the alternative $\mathbf{d}$ that maximizes this score. As we shall see shortly, this optimization problem is hard in the general case. Let us start with a case in which the winner can be computed in polynomial time.

If the importance structure is unconditional, then $\mathsf{level}(\mathcal{L}_j, \mathbf{d}, X_i)$ does not depend on $\mathbf{d}$; let us write it more simply $\mathsf{level}(\mathcal{L}_j, X_i)$. It can be computed in polynomial time by a simple exploration of the tree $\mathcal{L}_j$. Similarly, if the local preferences are unconditional, then $\Delta(\mathcal{L}_j, \mathbf{d}, X_i)$ does not depend on the whole vector $\mathbf{d}$ either but only on d_i, thus we write $\Delta(\mathcal{L}_j, d_i) = \Delta(\mathcal{L}_j, \mathbf{d}, X_i)$. When, for every voter, the importance structure and the local preferences are unconditional, we can thus write:

$$S_{\mathsf{Borda}}(V, \mathbf{d}) = \sum_{j=1}^{n} \sum_{i=1}^{q} 2^{q-\mathsf{level}(\mathcal{L}_j, X_i)} (1 - \Delta(\mathcal{L}_j, d_i))$$

This means that we can choose in polynomial time the winning value for each issue independently: it is the value d_i that maximizes $\sum_{j=1}^{n} 2^{q-\mathsf{level}(X_i, \mathcal{L}_j)} (1 - \Delta(\mathcal{L}_j, d_i))$.

Theorem 7. *If, for every voter,* the importance structure and the local preferences are unconditional, *then* WINNER *for Borda can be computed in polynomial time.*

As we will see shortly, if we lift either one of the unconditionality conditions, computing the Borda winners becomes NP-hard. However, we first describe how this problem can be converted to a WEIGHTED MINSAT problem.

We consider one propositional variable X_i for each issue $X_i \in \mathcal{I}$. Let $\mathcal{L}$ be an LP-tree and t a node of $\mathcal{L}$. Let X_i be the issue associated with t, and l be the level of t in $\mathcal{L}$ (where level 1 corresponds to the root and level q to the leaves). Let v be the instantiation of $\mathsf{Inst}(t)$ in that branch, we can associate with it a conjunction V of literals as follows: for every

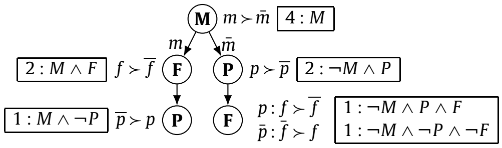

Fig. 3. LP tree LP_1 and its translation into weighted formulas.

variable X_i in $\mathsf{Inst}(t)$, V contains either X_i if v has value 1_i for X_i, or $\neg X_i$ if v has value 0_i for X_i. Then for each rule $\mathbf{u} : x \succ x'$ in the table associated with X_i at t, let U be the similar logical translation of $\mathbf{u}$; we define the formula $\psi_{t,\mathbf{u}}$, with weight 2^{q-l} for MINSAT, as follows:

- $2^{q-l} : U \wedge V \wedge X_i$ if $x = 1_i$; and
- $2^{q-l} : U \wedge V \wedge \neg X_i$ if $x = 0_i$,

As an example, LP tree LP_1 is recalled on Fig. 3 with the corresponding weighted formulas. Alternative $\bar{m}\bar{p}\bar{f}$ satisfies the formula $\neg M \wedge \neg P \wedge \neg F$, with weight 1: this alternative will be ranked $8 - 1 = 7$th; alternative $m\bar{f}\bar{p}$ satisfies M and $M \wedge \neg P$, with weight $4 + 1 = 5$, so it is ranked $8 - 5 = 3$rd.

Proposition 8. *Let $\mathcal{L}$ be an LP tree, let $\Psi(\mathcal{L}) = \{\psi_t \mid t \text{ node of } \mathcal{L}\}$, and let $\mathbf{d}$ be an alternative. The sum of the weights of the formulas of $\Psi(\mathcal{L})$ that are satisfied by $\mathbf{d}$ is the Borda score of $\mathbf{d}$ w.r.t. $\mathcal{L}$.*

Proof. The formulas that are satisfied by $\mathbf{d}$ are all on the branch corresponding to $\mathbf{d}$ in $\mathcal{L}$. Considering a node t_i on this branch labelled with variable X_i, the level of t_i is precisely $\mathsf{level}(\mathcal{L}, \mathbf{d}, X_i)$, so the weight is $2^{p-\mathsf{level}(\mathcal{L},\mathbf{d},X_i)}$, and the formula is satisfied if and only if the preferred value for X_i – given the values of $\mathbf{d}$ for the issues above t_i in this branch – is the value given by $\mathbf{d}$, so the overall weight is

$$\sum_{i=1}^{q} 2^{p-\mathsf{level}(\mathcal{L},\mathbf{d},X_i)}(1 - \Delta(\mathcal{L}, \mathbf{d}, X_i))$$

which is exactly the Borda score of $\mathbf{d}$ w.r.t. $\mathcal{L}$. □

Given an LP-profile $V = (\mathcal{L}_1, \ldots, \mathcal{L}_n)$, we consider the multiset of weighted formulas $\Psi(V) = \cup_j \Psi(\mathcal{L}_j)$ (note that there may be several copies of the same weighted formula) and we look for a valuation maximizing the sum of the weights of the formulas in $\Psi(V)$ that it satisfies. Since the formulas in $\Psi(V)$ are conjunctions of literals, $\Phi(V) = \{(w : \neg\phi) \mid (w : \phi) \in \Psi(V)\}$ is a set of weighted clauses, and a valuation minimizes the weights of the clauses satisfied in $\Phi(V)$ if and only if it maximizes the sum of the weights of the cubes satisfied in $\Psi(V)$, and this is an instance of WEIGHTED MINSAT.

On our example we get

$$\Phi(LP_1) = \left\{ \begin{array}{c} 4 : \neg M, \quad 2 : \neg M \vee \neg F, \quad 1 : \neg M \vee P, \\ 2 : M \vee \neg P, \quad 1 : M \vee \neg P \vee \neg F, \quad 1 : M \vee P \vee F \end{array} \right\}$$

Theorem 9. *For any profile V of LP-trees, there is a set of weighted clauses $\mathcal{C}$ of size polynomial in the size of V, such that the set of Borda co-winners for V is exactly the set of the valuations of $\mathcal{C}$ with the lowest.*

Some MINSAT solvers use a translation of MINSAT instances into MAXSAT ones. Using the same approach, we can also directly translate the WINNER problem for Borda into a WEIGHTED MAXSAT instance, but with more clauses: each weighted conjunction of literals $\phi = (w : l_1 \wedge \ldots \wedge l_k)$ that appears in $\Psi(V)$ is translated into a set of k weighted clauses $\Theta(\phi) = \{(w : l_1), (w : \neg l_1 \vee l_2), \ldots, (w : \neg l_1 \vee \neg l_2 \vee \ldots \vee l_k)\}$. Let $\Theta(\mathcal{L})$ denote the union of the $\Theta(\phi)$'s that we get for a tree $\mathcal{L}$. For $LP1$ we get:

$$\Theta(LP_1) = \left\{ \begin{array}{c} 4 : M, \quad 2 : M, \quad 2 : \neg M \vee F, \quad 1 : M, \quad 1 : \neg M \vee \neg P \\ 2 : \neg M, \quad 2 : M \vee P, \quad 1 : \neg M, \quad 1 : M \vee P, \quad 1 : M \vee \neg P \vee F \\ 1 : \neg M, \quad 1 : M \vee \neg P, \quad 1 : M \vee P \vee \neg F \end{array} \right\}$$

For a profile V, $\Theta(V)$ then denotes the union of the sets of weighted clauses obtained for each voter in V. To see why this MAXSAT formulation is equivalent to our Borda score maximization problem, consider an alternative $\mathbf{d}$, and a weighted conjunction $\phi = (w : l_1 \wedge \ldots l_k) \in \Psi(V)$:

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

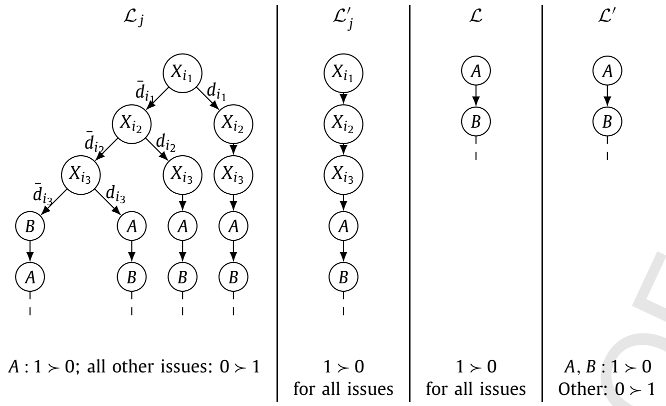

Fig. 4. Reduction of 3SAT to EVALUATION with Borda score and unconditional preferences.

- if $\mathbf{d} \models l_1 \wedge \ldots l_k$, then $\mathbf{d}$ satisfies all clauses in $\Theta(\phi)$; and
- if $\mathbf{d} \not\models l_1 \wedge \ldots l_k$, then there is exactly one clause in $\Theta(\phi)$ that is not satisfied by $\mathbf{d}$ (if j is the smallest index such that $\mathbf{d} \not\models l_j$, then $\mathbf{d}$ satisfies all clauses in $\Theta(\phi)$ except $\neg l_1 \vee \ldots \vee \neg l_{j-1} \vee l_j$).

Therefore, the "score" of $\mathbf{d}$ for $\Theta(V)$ is

$$\sum_{\substack{(w:C)\in\Psi(V)\\ \mathbf{d}\models C}} w|C| + \sum_{\substack{(w:C)\in\Psi(V)\\ \mathbf{d}\not\models C}} w(|C|-1) = \sum_{(w:C)\in\Psi(V)} w(|C|-1) + \sum_{\substack{(w:C)\in\Psi(V)\\ \mathbf{d}\models C}} w.$$

It is therefore a constant independent of $\mathbf{d}$, plus the "score" of $\mathbf{d}$ for $\Psi(V)$; thus maximizing the sum of the weights of the satisfied cubes in $\Psi(V)$ is equivalent to maximizing the sum of the weights of the satisfied clauses in $\Theta(V)$. Since $\Theta(V)$ contains at most q clauses of size at most q for each node in an LP-tree in V, its size is in $O(q^2|V|)$. Whether using a MAXSAT solver on this translation would be more efficient than using a recent MINSAT solver on $\Phi(V)$ remains to be tested.

We show next that we cannot expect a much better way of computing the Borda winner in the general case, since the problem is NP-complete.

Theorem 10. EVALUATION *is* NP*-complete for Borda, even if a single one of the following restrictions holds:*

1. *the local preferences of all voters are unconditional; or*
2. *the importance structure is the same for all voters and is unconditional.*

Proof. EVALUATION is in NP, because the Borda score of each alternative can be computed in polynomial time. We prove NP-hardness in the two subcases 1 and 2 by two different reductions from 3SAT: in a 3SAT instance, we are given a formula F over binary variables $X_1, \ldots, X_q$, which is a conjunction of t 3-clauses, $F = C_1 \wedge \ldots \wedge C_t$. We are asked whether there exists a valuation of the variables $X_1, \ldots, X_q$ under which F is true.

We will now describe how we can reduce a 3SAT instance to two EVALUATION instances, one with a profile with unconditional preferences, the other with a profile with common, unconditional importance.

Reduction with unconditional local preferences. There are $q+2$ issues: $\mathcal{I} = \{A, B\} \cup \{X_1, \ldots, X_q\}$. We first define the following $2t+2$ LP-trees from which the profile is constructed.

- For each $j \leq t$, we define two LP-trees $\mathcal{L}_j$ and $\mathcal{L}'_j$ with the following structures. Suppose C_j contains variables $X_{i_1}, X_{i_2}, X_{i_3}$ ($i_1 < i_2 < i_3$), and $d_{i_1}, d_{i_2}, d_{i_3}$ are the valuations of $X_{i_1}, X_{i_2}, X_{i_3}$ that correspond to literals in C_j: for instance, if $C_j = X_1 \vee \neg X_3 \vee X_5$ then $d_{i_1} = 1, d_{i_2} = 0, d_{i_3} = 1$. In the importance order of $\mathcal{L}_j$, the first three issues are $X_{i_1}, X_{i_2}, X_{i_3}$. The fourth issue is A and the fifth issue is B if and only if $X_{i_1} = d_{i_1}$, $X_{i_2} = d_{i_2}$, or $X_{i_3} = d_{i_3}$; otherwise the fourth issue is B and the fifth issue is A. The other issues are ranked below in a fixed order, independent of j. The local preferences in $\mathcal{L}_j$ are $0 \succ 1$ for all issues except A, for which it is $1 \succ 0$. In $\mathcal{L}'_j$ the importance order is $[X_{i_1} \rhd X_{i_2} \rhd X_{i_3} \rhd A \rhd B \rhd \text{Others}]$, and the local preferences are $1 \succ 0$ for all issues. The trees are depicted on the left half of Fig. 4.
- There are two additional LP-trees $\mathcal{L}$ and $\mathcal{L}'$ with the same importance order $[A \rhd B \rhd \text{Others}]$; in $\mathcal{L}$ the local preferences are $1 \succ 0$ for all issues; in $\mathcal{L}'$ the local preferences are $1 \succ 0$ for A and B, and $0 \succ 1$ for all other issues.

Let $\mathbf{d} = \mathbf{d}'ab$ be an alternative, where $a \in \{0,1\}$ is its value for issue A, $b \in \{0,1\}$ its value for issue B, and $\mathbf{d}'$ its projection onto $\{X_1, \ldots, X_q\}$. Because the preferences concerning each of the X_i's are opposite in $\mathcal{L}_j$ and $\mathcal{L}'_j$, the Borda score of $\mathbf{d}$ for

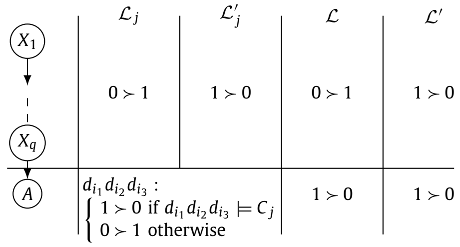

Fig. 5. Reduction of 3SAT to EVALUATION with Borda score and common, unconditional importance.

the profile $\{\mathcal{L}_j, \mathcal{L}'_j\}$ is a constant K_1 independent of j and $\mathbf{d}$, plus the score for issues A and B, ranked fourth and fifth out of $q+2$ issues. If we let $\mathbb{1}_j = 1$ if $\mathbf{d}' \models C_j$, and $\mathbb{1}_j = 0$ otherwise we have:

$$\begin{aligned}&S_{\text{Borda}}(\{\mathcal{L}_j, \mathcal{L}'_j\}, \mathbf{d}\,)\\&= K_1 + \mathbb{1}_j(a2^{q-2} + (1-b)2^{q-3}) + (1-\mathbb{1}_j)((1-b)2^{q-2} + a2^{q-3}) + a2^{q-2} + b2^{q-3}\\&= K_1 + 2^{q-3}[\mathbb{1}_j(2a+1-b) + (1-\mathbb{1}_j)(2(1-b)+a) + 2a+b]\end{aligned}$$

The Borda score of $\mathbf{d}$ for profile $\{\mathcal{L}, \mathcal{L}'\}$ is a constant K_2 plus the score for issues A and B, ranked first and second:

$$S_{\text{Borda}}(\{\mathcal{L}, \mathcal{L}'\}, \mathbf{d}\,) = K_2 + 2 \times (a2^{q+1} + b2^q) = K_2 + 2^{q+1}(2a+b).$$

We now describe the preference profile. It contains $4t$ LP-trees: $V = \{\mathcal{L}_j, \mathcal{L}'_j : 1 \le j \le t\} \cup V'$, where V' is composed of t copies of $\{\mathcal{L}, \mathcal{L}'\}$. Let $K = K_1 + K_2$, then:

$$\begin{aligned}&S_{\text{Borda}}(V, \mathbf{d}\,) =\\&tK + 2^{q-3}\sum_j[\mathbb{1}_j(2a+1-b) + (1-\mathbb{1}_j)(a+2-2b)) + 2a+b] + t2^{q+1}(2a+b).\end{aligned}$$

In particular, if $a = 0$ or $b = 0$, then $2a+1-b+2a+b \le 5$ and $a+2-2b+2a+b \le 5$, thus, independently from the value of $\mathbb{1}_j$, the term in the sum is bounded by 5 for every j; thus $S_{\text{Borda}}(V, \mathbf{d}) \le tK + 5t2^{q-3} + 2t2^{q+1}$. Or, equivalently: if $S_{\text{Borda}}(V, \mathbf{d}) > tK + 5t2^{q-3} + 2t2^{q+1}$, then it must be the case that $a = b = 1$.

Now, consider a threshold $T = tK + 5t2^{q-3} + 3t2^{q+1}$. Suppose first that $\mathbf{d}$ is such that $S_{\text{Borda}}(V, \mathbf{d}) \ge T > tK + 5t2^{q-3} + 2t2^{q+1}$. Then its values for A and B must both be 1, and $S_{\text{Borda}}(V, \mathbf{d}\,) = tK + 2^{q-3}\sum_j[2 \times \mathbb{1}_j + (1-\mathbb{1}_j) + 3] + 3t2^{q+1} = tK + 2^{q-3}\sum_j[4 + \mathbb{1}_j] + 3t2^{q+1}$. Now, since $S_{\text{Borda}}(V, \mathbf{d}\,) \ge T$, it must be the case that $\mathbb{1}_j = 1$ for every clause C_j, thus the projection of $\mathbf{d}$ onto D satisfies F; therefore F is satisfiable.

For the converse, suppose that F is satisfiable: let $\mathbf{d}'$ be a valuation that satisfies F, and consider the alternative $\mathbf{d}'1_A1_B$: $S_{\text{Borda}}(V, \mathbf{d}'1_A1_B) = T$, where $T = tK + 5t2^{q-3} + 3t2^{q+1}$.

Reduction with a common, unconditional importance structure. There are $q+1$ issues: $\mathcal{I} = \{X_1, \dots, X_q\} \cup \{A\}$. Let $\mathcal{O} = [X_1 \rhd X_2 \rhd \cdots \rhd X_q \rhd A]$ denote the fixed importance relation for the LP-trees. We construct a profile V consisting of $4t$ LP-trees with conditional importance relation $\mathcal{O}$, defined as follows:

- For each $j \le t$, V contains two LP-trees $\mathcal{L}_j$ and $\mathcal{L}'_j$ with unconditional local preferences for every issue except A: suppose C_j contains variables $X_{i_1}, X_{i_2}, X_{i_3}$ ($i_1 < i_2 < i_3$), in both LP-trees, $\text{Par}(A) = \{X_{i_1}, X_{i_2}, X_{i_3}\}$. The CPTs are defined as depicted on Fig. 5:
  - in $\mathcal{L}_j$, for every assignment $(d_{i_1}, d_{i_2}, d_{i_3})$ of $\{X_{i_1}, X_{i_2}, X_{i_3}\}$, the CPT entry for A is $d_{i_1}d_{i_2}d_{i_3} : 1 \succ 0$ if and only if C_j is satisfied by $(d_{i_1}, d_{i_2}, d_{i_3})$; the local preference is $0 \succ 1$ for every other issue.
  - in $\mathcal{L}'_j$, the CPT for A is the same as in $\mathcal{L}_j$, and the local preference is $1 \succ 0$ for every other issue.
- V contains $2t$ additional LP-trees: t copies of $\mathcal{L}$, where the local preference is $1 \succ 0$ for issue A, and $0 \succ 1$ for the other q issues; and t copies of $\mathcal{L}'$, where the local preferences are $1 \succ 0$ for all issues.

The additional $2t$ LP-trees are used to make sure that we only need to focus on alternatives whose A-component is 1. Let $\mathbf{d}' = (d_1, \dots, d_q)$ be a valuation of $X_1, \dots, X_q$. Then:

- $S_{\text{Borda}}(\{\mathcal{L}, \mathcal{L}'\}, \mathbf{d}'0_A) = \sum_{i=1}^{q} 2^{q+1-i}(1+0) = 2(2^q - 1)$: there are $q+1$ issues, issues $X_1, \dots X_q$ have ranks 1 to q, and whatever the value of d_i, it counts for 1 in one of the two trees, 0 in the other; 0_A is not the preferred value in $\mathcal{L}$ and $\mathcal{L}'$, so it does not add anything to the Borda score of $\mathbf{d}'0_A$.
- $S_{\text{Borda}}(\{\mathcal{L}, \mathcal{L}'\}, \mathbf{d}'1_A) = 2(2^q - 1) + 2$, because 1_A is the preferred value for A in $\mathcal{L}$ and $\mathcal{L}'$.
- $S_{\text{Borda}}(\{\mathcal{L}_j, \mathcal{L}'_j\}, \mathbf{d}'0_A) = 2(2^q - 1) + 2(1 - \mathbb{1}_j)$ (where again $\mathbb{1}_j = 1$ if $\mathbf{d}' \models C_j$, and $\mathbb{1}_j = 0$ otherwise).
- $S_{\text{Borda}}(\{\mathcal{L}_j, \mathcal{L}'_j\}, \mathbf{d}'1_A) = 2(2^q - 1) + 2 \times \mathbb{1}_j$.

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

Thus $S_{\text{Borda}}(V, \mathbf{d}'0_A) = 4t(2^q - 1) + 2t - 2K(F, \mathbf{d}')$, where $K(F, \mathbf{d}')$ is the number of clauses in F that are satisfied by $\mathbf{d}'$; whereas $S_{\text{Borda}}(V, \mathbf{d}'1_A) = 4t(2^q - 1) + 2t + 2K(F, \mathbf{d}')$. Hence there exists an alternative whose Borda score is at least $T = 4t(2^q - 1) + 4t$ if and only if the 3SAT instance is a *yes* instance. This completes the proof. □

Beyond Borda and k-approval. We end this section by discussing the generalization of the constructions of equivalent satisfiability problems for k-approval and Borda to other scoring rules. The common point of the two constructions is that the set of weights has the nice property that only a small number of formulas was needed to encode the scores obtained by the alternatives: more precisely, the number of formulas generated for a single LP-tree is at most the number of nodes of the LP-trees, and each formula has a size in $O(q)$, thus the total size of the construction is in $O(q.\sum_{j=1}^{n} |\mathcal{L}_j|)$, while the number of alternatives (2^q) can, in general, be exponentially larger.

- In the case of k-approval, this is due to the fact that the number of different weights in the scoring vector is very small (only two).
- In the case of Borda, this is due *more generally* to the fact that the weights are regular enough so that they can be generated by a succinct basis.

The construction can thus be generalized to scoring vectors that use a small number of weights, or more generally a "small basis". We give an example: $s = (s_1, \dots, s_{2^q})$ where

$$s_j = \begin{cases} 2(2^{q-2} - j + 1) & \text{if } j \leq 2^{q-2} \\ 1 & \text{if } 2^{q-2} + 1 \leq j \leq 2^{q-1} \\ 0 & \text{if } j > 2^{q-1} \end{cases}$$

The corresponding set of weighted formulas – assuming for simplicity that LP has an unconditional structure $X_1 \rhd X_2 \rhd X_3$ and local preferences $x_i \succ \overline{x_i}$ for every i – is

$$\{2^{q-2} : X_1 \wedge X_2 \wedge X_3; 2^{q-3} : X_1 \wedge X_2 \wedge X_4; \dots, 2 : X_1 \wedge X_2 \wedge X_q; 1 : X_1 \wedge X_2; 1 : X_1\}$$

For instance, for $q = 4$ we get $s = (8, 6, 4, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)$ and the set of weighted formulas corresponding to LP is

$$\{4 : X_1 \wedge X_2 \wedge X_3; 2 : X_1 \wedge X_2 \wedge X_4; \dots, 2 : X_1 \wedge X_2 \wedge X_q; 1 : X_1 \wedge X_2; 1 : X_1\}$$

6. Condorcet

We first observe that the existence of a Condorcet winner is not guaranteed for LP profiles:

Example 6. Consider the following profile with two issues X and Y, and three voters:

1: $X \rhd Y, x \succ \bar{x}, y \succ \bar{y}$: her preference relation is $xy \succ x\bar{y} \succ \bar{x}y \succ \bar{x}\bar{y}$.
2: $Y \rhd X, \bar{x} \succ x, y \succ \bar{y}$: her preference relation is $\bar{x}y \succ xy \succ \bar{x}\bar{y} \succ x\bar{y}$.
3: $Y \rhd X, \bar{x} \succ x, \bar{y} \succ y$: her preference relation is $\bar{x}\bar{y} \succ x\bar{y} \succ \bar{x}y \succ xy$.

It can be checked easily that this profile has no Condorcet winner.

For any type of LP profiles, deciding whether a given alternative $\mathbf{c}$ is a Condorcet winner is in coNP: a certificate that $\mathbf{c}$ is not a Condorcet winner is an alternative $\mathbf{d}$ that beats $\mathbf{c}$, which can be checked by comparing $\mathbf{d}$ and $\mathbf{c}$ for every LP-tree in the profile.

The following result shows that checking if a given alternative is a Condorcet winner is hard.

Theorem 11. *For LP profiles, deciding whether a given alternative is a Condorcet winner is* coNP*-hard, even if, for every voter, the local preferences are unconditional and the importance tree is unconditional.*

Proof. We prove the hardness using a reduction from the decision version of MAX HORN-SAT, known to be NP-complete [34]

Input a Horn formula $F = C_1 \wedge \dots \wedge C_t$ over variables $\{X_1, \dots, X_q\}$, and an integer number K.
Question Is there a valuation that satisfies more than K of the t clauses?

Let the set of issues be $\{c\} \cup \{X_1, \dots, X_q\}$. For a valuation $\mathbf{d}$ over $\{X_1, \dots, X_q\}$ we consider the completions $\mathbf{d}0_c$ and $\mathbf{d}1_c$ of $\mathbf{d}$, and we define the following LP-trees:

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

$\mathcal{L}_0$**:** the importance order is $[c \rhd \text{Others}]$, and the local preferences are $0 \succ 1$ for all issues. Note that $\mathbf{d}0_c \succ_{\mathcal{L}_0} \mathbf{d}1_c \succeq_{\mathcal{L}_0} \mathbf{1}$ for all $\mathbf{d}$.

$\mathcal{L}_j$**, if** $C_j = \neg X_1 \vee \cdots \vee \neg X_l$**:** the importance order is $[X_1 \rhd \cdots \rhd X_l \rhd c \rhd \text{Others}]$ and the local preferences are:
- $0 \succ 1$ for $X_1 \dots X_l$;
- $1 \succ 0$ for the other issues.

Note that if $\mathbf{d} \models C_j$, then $\mathbf{d}1_c \succ_{\mathcal{L}_j} \mathbf{d}0_c \succ_{\mathcal{L}_j} \mathbf{1}$, and if $\mathbf{d} \not\models C_j$, then $\mathbf{1} \succeq_{\mathcal{L}_j} \mathbf{d}1_c \succ_{\mathcal{L}_j} \mathbf{d}0_c$.

$\mathcal{L}_j$**, if** $C_j = \neg X_1 \vee \cdots \vee \neg X_l \vee X_{l+1}$**:** the importance order is $[X_1 \rhd \cdots \rhd X_{l+1} \rhd c \rhd \text{Others}]$, and local preferences $0 \succ 1$ for issues $c, X_1, \dots, X_l$, and $1 \succ 0$ for other issues; then
- $\mathbf{d}0_c \succ_{\mathcal{L}_j} \mathbf{d}1_c \succ_{\mathcal{L}_j} \mathbf{1}$ if $\mathbf{d} \models \neg X_1 \vee \cdots \vee \neg X_l$;
- $\mathbf{d}0_c \succ_{\mathcal{L}_j} \mathbf{1} \succeq_{\mathcal{L}_j} \mathbf{d}1_c$ if $\mathbf{d} \models X_1 \wedge \cdots \wedge X_l \wedge X_{l+1}$;
- $\mathbf{1} \succ_{\mathcal{L}_j} \mathbf{d}0_c \succ_{\mathcal{L}_j} \mathbf{d}1_c$ if $\mathbf{d} \not\models C_j$ (where the first preference comes from the facts that $\mathbf{1}$ and $\mathbf{d}0_c$ coincide on $X_1, \dots, X_l$ and that $1_{l+1} \succ 0_{l+1}$).

Let V be a profile containing two copies of each $\mathcal{L}_j$ for $1 \le j \le t$, and $2t - 4K$ copies of $\mathcal{L}_0$. Let $N(\mathbf{d})$ be the number of clauses satisfied by $\mathbf{d}$. The number of LP-trees in V that rank $\mathbf{d}0_c$ ahead of $\mathbf{1}$ is $2t - 4K + 2N(\mathbf{d})$; since V contains $4t - 4K$ LP-trees, $\mathbf{d}0_c \succ_V \mathbf{1}$ if and only if $2t - 4K + 2N(\mathbf{d}) > 2t - 2K$, that is:

$$\mathbf{d}0_c \succ_V \mathbf{1} \text{ if and only if } N(\mathbf{d}) > K \tag{3}$$

Hence, if there is a valuation $\mathbf{d}$ that satisfies more than K clauses, then $\mathbf{1}$ is not a Condorcet winner because $\mathbf{d}0_c \succ_V \mathbf{1}$.

Conversely, assume there is an alternative $\mathbf{d}'$ that beats $\mathbf{1}$. If $\mathbf{d}'$ is of the form $\mathbf{d}0_c$ then $\mathbf{d}'$ satisfies more than K clauses because of equivalence (3) above. If $\mathbf{d}'$ is of the form $\mathbf{d}1_c$, then $2t - 4K + 2N_1(\mathbf{d}) > 2t - 2N_1(\mathbf{d})$, where $N_1(\mathbf{d})$ is the number of clauses in which $\mathbf{d}$ satisfies at least one negative literal: then $4N_1(\mathbf{d}) > 4K$, i.e., $N_1(\mathbf{d}) > K$, and since $N(\mathbf{d}) \geq N_1(\mathbf{d})$, again $\mathbf{d}$ satisfies more than K clauses.

This shows that $\mathbf{1}$ is *not* a Condorcet winner if and only if there is a valuation that satisfies more than K clauses, hence the coNP-completeness result. □

From this result we infer that deciding the EXISTENCE of a Condorcet winner for a profile consisting of LP-trees with fixed importance structure is in $\mathsf{NP}^{\mathsf{NP}} = \Sigma_2^{\mathsf{P}}$. Whether it is Σ_2^{P}-complete is an open problem.

This hardness result is likely to extend to the WINNER problem for most *Condorcet-consistent rules*, which elect the Condorcet winner whenever there exists one. This is left for further research (see however the conference version of our paper for some preliminary results).

7. Fixed importance LP-profiles

We now focus on the class of profiles with *fixed importance tree* (conditional or unconditional), *common to all voters*: LP-trees given by all voters have a common importance structure but may have different local preferences on the values of issues. Let us refer to such LP profiles as *FCI profiles*; if the fixed importance structure is unconditional then we say that the profile is *FUI*. This restriction is practically significant, as in many contexts we may assume that there is an *objective* relative importance of issues, common to all agents. (This can be compared to the assumption that all agents have common preferential dependency structure between variables, commonly made in voting on combinatorial domains [24].) We may wonder whether this restriction makes winner determination easier for some of the voting rules we consider. The existence of a Condorcet winner for FCI profiles was proven in [35]. We first define below a rule which is specifically tailored for FCI profiles, and then give some tractability results for other rules in this case.

7.1. The sequential majority winner and Condorcet-consistent rules

A rule which, for this domain restriction, is easy to compute and cheap in terms of communication, is defined in [3]: we choose a value for the issue at the root of the tree according to the majority rule (possibly with a tie-breaking mechanism if we have an even number of voters); then, we go down the branch corresponding to the winner for this issue, and choose a value for the issue at the next node using again the majority rule; and so on. The winner is called a *sequential majority winner*. When n is odd, there is a unique winning value at each step, and therefore, a unique sequential majority winner. When n is even, all winners obtained by choosing an arbitrary value in case of a tie are called *sequential majority cowinners*.

It was proven in [35] that for profiles composed of conditionally lexicographic preferences, and with an odd number of voters, the sequential majority winner is a Condorcet winner. This, together with the fact that the sequential majority winner can be computed in polynomial time, shows that the winner of any Condorcet-consistent rule applied to FCI profiles with an odd number of voters can be computed in polynomial time.[3]

[3] This result extends to profiles with an even number of voters, provided that the rule outputs all weak Condorcet winners when there exists at least one; in this case, the sequential majority cowinners are the weak Condorcet winners.

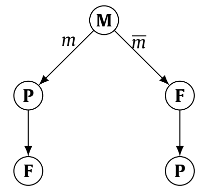

Fig. 6. The importance tree T_1 for Example 7.

Example 7. Consider the conditional importance tree T_I depicted on Fig. 6. Consider a profile V' with three voters, all having conditionally lexicographic preferences with the same importance tree T_I. To determine the sequential majority winner, we first look at the voters' preferences over the values of the most important issue, namely, **M**. Assume that two voters prefer $\overline{m}$ to m: the winning value is $\overline{m}$. Now, given that $\mathbf{M} = \overline{m}$, the second most important issue (for all voters) is **F**. We look at the voters' preferences over the values of **F**. Assume that two voters prefer f to $\bar{f}$: the winning value is f. Finally, we look at the voters' preferences over the values of **P**. Assume all voters prefer $\overline{p}$ to p: the winning value is $\overline{p}$, and finally the sequential Condorcet winner is $\overline{m} f \overline{p}$.

7.2. k-Approval for FCI profiles

We have seen that computing the winners for α-proportion approval is hard in general for $\alpha \in \mathsf{Rat}(M) \setminus \{\frac{1}{2}\}$, for any positive integer $M \le q$. It turns out that it is easier for FCI profiles, because in this case we can reduce the problem to computing the co-winners for j-approval, with $j = \alpha 2^M$, over a set of partial alternatives that correspond to the paths of length M from the root in the common importance tree T. Formally, this set $\mathcal{C}(T)$ of partial alternatives is defined as follows:

- for every node t of T at level M, labelled with issue X, let $\mathcal{C}(t) = \{x, \overline{x}\}$;
- if t is a node at a level $< M$, labelled with issue X, and t has two children, one child t' for value x, and another one t'' for value $\overline{x}$, let $\mathcal{C}(t) = \{x\mathbf{u} \mid \mathbf{u} \in \mathcal{C}(t')\} \cup \{\overline{x}\mathbf{u} \mid \mathbf{u} \in \mathcal{C}(t'')\}$;
- if t is a node at a level $< M$, labelled with issue X, and t has one child t', let $\mathcal{C}(t) = \{x\mathbf{u}, \overline{x}\mathbf{u} \mid \mathbf{u} \in \mathcal{C}(t')\}$.
- finally, $\mathcal{C}(T) = \mathcal{C}(\text{root})$.

For instance, for the tree T_1 depicted on Fig. 6 and for $M = 2$, $\mathcal{C}(T_1) = \{mp, m\overline{p}, \overline{m} f, \overline{m}\overline{f}\}$.

The set $\mathcal{C}(T)$ has the following properties:

- every complete alternative extends one and only one partial alternative in $\mathcal{C}(T)$; and for every partial alternative there are 2^{q-M} complete alternatives that extend it;
- given an LP-tree $\mathcal{L}$ with importance tree T, the order $\succ_{\mathcal{L}}$ induces a ranking $\succ_{\mathcal{L}}^{M}$ of the partial alternatives in $\mathcal{C}(T)$: let $\mathbf{u}, \mathbf{u}'$ be two different partial alternatives of $\mathcal{C}(T)$, let $\mathbf{d}, \mathbf{d}'$ be two complete alternatives that extend $\mathbf{u}$ and $\mathbf{u}'$ respectively, then $\mathbf{u} \succ_{\mathcal{L}}^{M} \mathbf{u}'$ if $\mathbf{d} \succ_{\mathcal{L}} \mathbf{d}'$;
- a partial alternative $\mathbf{u}$ is among the best j in $\mathcal{C}(T)$ w.r.t. $\succ_{\mathcal{L}}^{M}$ if and only if every complete alternative $\mathbf{d}$ that extends $\mathbf{u}$ is among the $j 2^{q-M} = \alpha 2^q$ best alternatives of $\succ_{\mathcal{L}}$.

This means that the $\alpha 2^q$ best alternatives for $\succ_{\mathcal{L}}$ are entirely defined by the best $\alpha 2^M$ partial alternatives in $\mathcal{C}(T)$ w.r.t. $\succ_{\mathcal{L}}^{M}$. Therefore:

Theorem 12. *Let $M \le q$, let $\alpha \in \mathsf{Rat}(M)$, and let $j = \alpha 2^M$. For FCI profiles,* WINNER *for α-proportion approval can be computed in time $O(jnM)$.*

Proof. The procedure proposed at the beginning of section 4 works here too: for all of the n LP-trees in the profile, compute their j best partial alternatives, and compute the j-approval scores of these. □

For instance, for the importance tree of Fig. 6, let $M = 2$ and $\alpha = 3/4$: each voter will add 1 to the scores of three of the four partial alternatives mp, $m\overline{p}$, $\overline{m} f$ and $\overline{m}\overline{f}$, and the partial winner will be the one with the highest score. The 3/4-proportion-approval co-winners will be the two alternatives that extend that partial winner.

This is efficient only if M is a constant: in this case, the running time is linear (in n).

7.3. Borda for FCI profiles

We already know, from Theorem 7, that for profiles with unconditional preferences and unconditional importance, the Borda winner can be computed in polynomial time. However, in the case of a *common, unconditional* importance tree and unconditional preferences, we have moreover an interesting characterization of the Borda (co)winner(s):

Theorem 13. *For FCI profiles with unconditional importance and unconditional preferences: when n is odd, the Borda winner is the sequential majority winner, and when n is even, the Borda cowinners are the sequential majority cowinners.*

Proof. Without loss of generality, assume that the common, unconditional importance order is $X_1 \rhd X_2 \rhd \ldots \rhd X_q$. Consider profile $V = (\mathcal{L}_1, \ldots, \mathcal{L}_n)$ where all the $\mathcal{L}_j$'s have that importance order. Let $\mathbf{x} = x_1 \ldots x_q$ denote a sequential majority cowinner. For every $i \in 1, .., q$, let k_i be the number of voters with local preference $x_i \succ \overline{x_i}$: because $\mathbf{x}$ is a sequential majority cowinner, we have $k_i \geq \frac{n}{2}$; and the Borda score of $\mathbf{x}$ for V can be expressed as $S_{\text{Borda}}(\mathbf{x}) = \sum_{i=1}^{q} 2^{q-i} k_i$. Moreover, for any other alternative $\mathbf{y} = y_1 \ldots y_q$, for every issue X_i, let k'_i be the number of voters who prefer y_i over $\overline{y_i}$: either $y_i = x_i$ and $k_i - k'_i = 0$, or $y_i = \overline{x_i}$ and $k'_i = n - k_i \leq \frac{n}{2}$ because $k_i \geq \frac{n}{2}$, thus $k_i - k'_i \geq 0$. Since $S_{\text{Borda}}(\mathbf{y}) = \sum_{i=1}^{q} 2^{q-i} k'_i$, we have

$$S_{\text{Borda}}(\mathbf{x}) - S_{\text{Borda}}(\mathbf{y}) = \sum_{i=1}^{q} 2^{q-i}(k_i - k'_i) \geq 0$$

and the difference can only be equal to 0 if $k_i - k'_i = 0$ for every issue, that is, if $\mathbf{y}$ too is a sequential cowinner.

When n is odd, there is a unique sequential majority winner $\mathbf{x}$, and for every other alternative $\mathbf{y}$ there is at least one issue X_i such that $y_i = \overline{x_i}$, and then $k'_i = n - k_i < \frac{n}{2}$; thus $S_{\text{Borda}}(\mathbf{x}) > S_{\text{Borda}}(\mathbf{y})$. Therefore, in this case, $\mathbf{x}$ is the unique Borda winner. □

This result is specific to LP-trees with fixed unconditional importance and unconditional preferences. Here is a counterexample for fixed unconditional importance and conditional preferences (note that we already know that in this case, the problem is NP-hard, and therefore we already knew that the Borda winner cannot always be the sequential majority winner).

Example 8. Consider three issues, common importance order $X_1 \rhd X_2 \rhd X_3$, and five LP-trees:

- LP_1, LP_2, LP_3 with unconditional preferences $x_1 \succ \overline{x_1}$, $x_2 \succ \overline{x_2}$, $x_3 \succ \overline{x_3}$;
- LP_4, LP_5: $\overline{x_1} \succ x_1$, $x_1 : \overline{x_2} \succ x_2$, $x_1 : \overline{x_3} \succ x_3$, $\overline{x_1} : x_2 \succ \overline{x_2}$, $\overline{x_1} : x_3 \succ \overline{x_3}$.

The sequential majority winner is $x_1x_2x_3$. It is ranked first in the first three votes and last in the last two, therefore its Borda score is 21. The Borda score of $\overline{x_1}x_2x_3$ (ranked 5th in the first three votes and first in the last two) is 23.

Here is now a counterexample for fixed conditional importance and unconditional preferences.

Example 9. Consider the conditional importance tree of Example 7, with three issues **M**, **P** and **F**. We have 19 voters, all preferring f to $\overline{f}$, and

- for 10 of them we have $m \succ \overline{m}$ and $p \succ \overline{p}$;
- for the other 9 we have $\overline{m} \succ m$ and $\overline{p} \succ p$.

The sequential majority winner is mpf. Its Borda score is 79 (it is ranked first by the voters of the first group and 7th by those of the second group); however, the Borda winner is $\overline{m}pf$, with Borda score 84 (it is ranked 5th by voters of the first group and 2nd by those of the second group).

Although, in the case of fixed conditional importance and unconditional preferences, the Borda winner is not always the sequential majority winner, still, it can be computed in time polynomial in the number of nodes of the importance tree, with a bottom-up algorithm that collects the optimal value for each issue in each branch with its contribution to the Borda score, and then chooses, at each binary node, the best value depending on the optimal contributions found in the two branches below it. Before giving the algorithm, we illustrate the method on an example.

Example 10. We consider the same profile as in Example 9. We start with the $\overline{m}$ branch: outcomes in this branch are of the form $\overline{m}xy$, where x is the value for **F** and y is the value for **P**. The Borda score of $\overline{m}xy$ is of the form $4\alpha + 2\beta + \gamma$, where α (respectively β, γ) is the number of voters whose preferred value is $\overline{m}$ for issue **M** (respectively, x for **F** and y for **P**). We can look for the values for **F** and **P** that maximize β and γ: starting at the leaf, there are 10 voters who prefer p, 9 who prefer $\bar{p}$, so we choose $y = p$, and get a local score of $1 \times 10 = 10$. Going up at the **F** node, all voters prefer f, so we choose $x = f$, and get a local score of 2×19, that we add to what has been scored at the leaf to get a current score of $38 + 10 = 48$: this is the best score that we can have for outcomes of the form $\overline{m}xy$, and it is reached for $\overline{m}fp$. Similarly, outcomes of the form mxy have a score of the form $4\alpha + 2\gamma + \beta$. We maximize again β and γ by choosing values f and p respectively, and get $2 \times 10 + 1 \times 19 = 39$ as optimal 'subscore' in the m branch: this is the highest we can get for outcomes of the form mxy, and is reached for mfp. Finally, the Borda score of $\overline{m}fp$ is $4 \times 9 + 48 = 84$, whereas that of mfp is $4 \times 10 + 39 = 79$, so the winner is $\overline{m}fp$.

We now describe the algorithm. Consider a profile V with common importance tree $\mathcal{T}$. For any node t at level l of $\mathcal{T}$ labelled with issue X, let $\mathcal{I}_t = \mathcal{I} \setminus (\mathsf{Anc}(t) \cup \{X\})$. For each voter $\mathcal{L}_j$, the subtree of $\mathcal{L}_j$ rooted at t is an LP-tree over $\mathcal{I}_t \cup \{X\}$, and completely orders the valuations of $\mathcal{I}_t \cup \{X\}$ – there are 2^{q-l+1} of them. Thus, for each value x of X, and each valuation $\mathbf{u}$ of $\mathcal{I}_t$, we can define their Borda score, w.r.t. to the profile that is composed of the subtrees rooted at t of the $\mathcal{L}_j$'s:

$$S_{\mathsf{Borda}}(t, V, x\mathbf{u}) = \sum_{Y \in \mathcal{I}_t \cup \{X\}} 2^{q-\mathsf{level}(t,\mathbf{u},Y)} \Big(\sum_{j=1}^{q} (1 - \Delta(\mathcal{L}_j, \mathbf{u}, Y)) \Big)$$

where

- $\mathsf{level}(t, \mathbf{u}, Y)$ denotes the level of variable Y in the branch of $\mathcal{T}$ going through t and continuing according to $\mathbf{u}$; and
- $\Delta(\mathcal{L}_j, \mathbf{u}, Y)$ denotes the rank, in the local preference of voter j, of the value for Y in $\mathbf{u}$.

If we let t'_x denote the node below t in the branch corresponding to the value x for X at t, then:

$$S_{\mathsf{Borda}}(t, V, x\mathbf{u}) = 2^{t-l} \sum_{j=1}^{q} (1 - \Delta(\mathcal{L}_j, x\mathbf{u}, X)) + S_{\mathsf{Borda}}(t'_x, V, \mathbf{u})$$

This shows that, when looking for the partial valuation $x\mathbf{u}$ that maximizes this score, we can, for every value x for X, look for the partial valuation $\mathbf{u}_x$ of $\mathcal{I}_t$ that maximizes $S_{\mathsf{Borda}}(t'_x, V, \mathbf{u}_x)$, and then look for the x that maximizes $S_{\mathsf{Borda}}(t, V, x\mathbf{u}_x)$. Algorithm 2 implements this approach.

Algorithm 2: $\mathsf{Borda}^+(t, V)$.

input: an importance tree $\mathcal{T}$ over a set $\mathcal{I}$ of q issues, a node t at level l of $\mathcal{T}$, labelled with X;
a FCI-UP profile V with n voters;
output: a pair consisting of a partial valuation of $\mathcal{I}_t$ that maximizes the partial score at t, and of this optimal partial score;

1. $k_x \leftarrow$ number of voters in V who prefer x to $\bar{x}$; $k_{\bar{x}} \leftarrow n - k_x$;
   $s_x \leftarrow 2^{q-l} k_x$; $s_{\bar{x}} \leftarrow 2^{q-l} k_{\bar{x}}$;
2. if t has a single child t':
   (a) $(\mathbf{u}', s_{\mathbf{u}'}) \leftarrow \mathsf{Borda}^+(t', V)$;
   (b) if $s_x \geq s_{\bar{x}}$: return $(x\mathbf{u}', s_x + s_{\mathbf{u}'})$;
   else: return $(\bar{x}\mathbf{u}', s_{\bar{x}} + s_{\mathbf{u}'})$;
3. else if t has two children t' and t'', corresponding to values x and $\bar{x}$:
   (a) $(\mathbf{u}', s_{\mathbf{u}'}) \leftarrow \mathsf{Borda}^+(t', V)$; $(\mathbf{u}'', s_{\mathbf{u}''}) \leftarrow \mathsf{Borda}^+(t'', V)$;
   (b) if $s_{\mathbf{u}'} + s_x \geq s_{\mathbf{u}''} + s_{\bar{x}}$: return $(x\mathbf{u}', s_{\mathbf{u}'} + s_x)$;
   else: return $(\bar{x}\mathbf{u}'', s_{\mathbf{u}''} + s_{\bar{x}})$;
4. else *(t is a leaf)*
   if $s_x \geq s_{\bar{x}}$: return (x, s_x); else: return $(\bar{x}, s_{\bar{x}})$;

Theorem 14. *For fixed, unconditional importance (FUI) profiles with unconditional preferences, the Borda winner can be computed in time in $O(nN)$, where N is the number of nodes of the importance tree.*

Proof. Algorithm 2 computes a winner with its score. It is called once at each node of the tree, and at each node it must count the number of times each value of the labelling issue is preferred in the profile. □

8. Fixed preferences

We study now the case where all agents have the same local, unconditional preferences. We can assume, without loss of generality that $1_i \succ 0_i$ for every agent and every issue X_i: issues can be seen as objects, and every agent has a preference for having an object rather than not, everything else being equal. For instance, if a voter has the unconditional importance order $X_1 \rhd X_2 \rhd X_3$ then the ordering over alternatives / sets of objects is $123 \succ 12 \succ 13 \succ 1 \succ 23 \succ 2 \succ 3 \succ \emptyset$ (where a subset S of $\{1, 2, 3\}$ represents an alternative that takes 1 on X_i if and only if $i \in S$. For example, 13 represents $1_1 0_2 1_3$).

We consider in this section profiles where all preferences are unconditional, of the form $1_i \succ 0_i$; we call them *FP profiles*. Obviously, the best outcome for every agent is **1**, and applying any reasonable voting rule (more precisely, any voting rule that satisfies unanimity) will select this alternative. However, winner determination ceases to be trivial if we have a *constraint* on the set of feasible alternatives. Constraints can take various forms. In a multiwinner voting domain, the usual constraint is that the number of objects to be taken (that is, the number of elected candidates) is fixed to a constant. More general constraints can consist of lower and upper bounds on the cardinality of the number of objects taken, on their cumulated volume, or more generally, arbitrary constraints expressed in a succinct way as in binary aggregation with constraints [36]. For the sake of simplicity, the only constraint we are considering here is the simplest one: there is a *fixed number K of objects that we can take*, or equivalently, a fixed number of issues assigned to true. Let us denote by $\mathcal{I}[K]$ the set of all alternatives **d** with exactly K issues assigned to true (the *feasible* alternatives). Then we define the following problems, given a voting rule r for which the winner is determined by maximizing a score S:

EXACT-K WINNER DETERMINATION **(for** r / S**):**

> **Input** an FP LP-profile V, an integer $K \le q$
> **Output** find an alternative in $\mathcal{I}[K]$ maximizing $S(V, \cdot)$.

EXACT-K EVALUATION **(for** r / S**):**

> **Input** an FP LP-profile V, two integers K and T
> **Question** is there an alternative **d** in $\mathcal{I}[K]$ such that $S(V, \mathbf{d}) > T$?

Observation 15. If the voters all have the same importance structure, then for any voting rule satisfying unanimity, EXACT-K WINNER DETERMINATION and EXACT-K EVALUATION are in P.

This is straightforward from the fact that all voters have the same preferred set of K issues.

8.1. Fixed preferences and unconditional importance

An interesting sub-case is when we have unconditional importance for every voter. Let us refer to such LP profiles as *UI-FP profiles*. Each voter ranks the objects $X_1, \ldots, X_q$ according to how important they are for her, and we want to output the 'best' possible combination of objects. Since each voter only has to specify her (importance) ranking of the objects, a profile has the form $(\rhd_1, \ldots, \rhd_n)$, where $\rhd_i$ is the object ranking by voter i. We might then be tempted to proceed in two steps:

1. use a classical aggregation rule to aggregate the voters' rankings over issues into a collective ranking over issues;
2. select the first K issues in the resulting ranking.

If Step 1 can be done in time polynomial in the number of issues, then this procedure will be tractable. If EXACT-K WINNER DETERMINATION for voting rule r can be done in such a way using aggregation function f, we say that f *simulates* r. The question is, are there any interesting pairs (r, f) such that f simulates r? The next result answers positively for $r =$ Borda.

Theorem 16 *(*EXACT-K WINNER DETERMINATION *for* Borda *with UI-FP profiles). Let* $V = (\rhd_1, \ldots, \rhd_n)$ *be a UI-FP profile and* $\rhd_V$ *be the ranking of issues obtained by applying the scoring rule* $F_{\mathbf{s}}$ *with* $\mathbf{s} = (2^{q-1}, 2^{q-2}, \ldots, 1)$*. The top K issues in* $\rhd_V$ *form the alternative* $\mathbf{d} \in \mathcal{I}[K]$ *with maximal Borda score* $S_{\text{Borda}}(V, \mathbf{d})$*:* $F_{\mathbf{s}}$ *simulates* Borda.

Proof. For $\mathbf{d} \in \mathcal{I}[K]$, let $X_{i_1}, \ldots, X_{i_K}$ be the issues for which **d** has value 1. Then $S_{\text{Borda}}(V, \mathbf{d}) = \sum_{j=1}^{K} S_{\text{Borda}}(V, X_{i_j})$, where $S_{\text{Borda}}(V, X_{i_j}) = \sum_{j=1}^{n} 2^{q-\text{rank}(X_i, \rhd_j)}$ is the contribution of issue X_{i_j} to the Borda score of an alternative that assigns it to true. In order to find the alternative in $\mathcal{I}[K]$ with maximum Borda score, we only have to find the K issues X_{i_j} that maximize $S_{\text{Borda}}(V, X_{i_j})$, and they are the top K issues in $\rhd_V$. □

Example 11. Consider the FI LP-profile with three voters and five issues $V = (X_1 \rhd X_4 \rhd X_2 \rhd X_3 \rhd X_5,\ X_2 \rhd X_4 \rhd X_5 \rhd X_1 \rhd X_3,\ X_3 \rhd X_5 \rhd X_4 \rhd X_2 \rhd X_1)$. We have $S_{\text{Borda}}(V, \mathbf{d}, X_1) = 16 + 1 + 1 = 18$, $S_{\text{Borda}}(V, \mathbf{d}, X_2) = 4 + 16 + 2 = 22$, $S_{\text{Borda}}(V, \mathbf{d}, X_3) = 2 + 2 + 16 = 20$, $S_{\text{Borda}}(V, \mathbf{d}, X_4) = 8 + 8 + 4 = 20$, and $S_{\text{Borda}}(V, \mathbf{d}, X_5) = 1 + 4 + 8 = 13$, thus $X_2 \rhd_V X_3 \sim_V X_4 \rhd_V X_1 \rhd_V X_5$ and, for $K = 3$, the best feasible alternative is $0_1 1_2 1_3 1_4 0_5$.

This result also shows that the Borda rule applied to UI-FP profiles can be seen as a *best-K* multiwinner scoring rule [37]. The Borda rule is not the only rule which, applied to UI-FP profiles, can be seen as a best-K rule. For instance, if $q = 2$, then the scoring rule associated with scoring function S, applied to UI-FP profiles, corresponds to a best-K voting rule if and only if $\vec{s}$ has the form $(t_1 + t_2, t_1, t_2, 0)$.

A much wider family of multiwinner rules, including best-K, but also many others, is known as *committee scoring rules* [38]. We may wonder whether other such rules in this family can be obtained by applying some scoring rule to UI-FP profiles. The answer is yes. Taking $K = 2$ as an example, $\frac{1}{4}$-proportion approval, applied to UI-FP profiles, corresponds to the multiwinner rule that elects the pair of items appearing most frequently as the pair of preferred items in the votes. For instance, if the profile is $(a \rhd b \rhd c \rhd d,\ a \rhd c \rhd b \rhd d,\ b \rhd c \rhd a \rhd d,\ c \rhd d \rhd a \rhd b,\ d \rhd c \rhd b \rhd a)$, then the winner is $\{c, d\}$ since it appears twice as preferred pair of items while all other pairs appear at most once. This rule is known as the *perfectionist* rule [39]. This observation can of course be generalised to $\frac{1}{2^K}$-proportion approval for any K. Characterising the exact subset of committee scoring rules that can be recovered this way appears to be a more difficult question, which is left for further study.

We show next that exact-K winner determination for UI-FP profiles and for α-proportion-approval is hard. This implies that no simple aggregation rule, applied to the importance ordering of the voters, can simulate α-proportion-approval.

Theorem 17. *Let $\alpha \in \mathrm{Rat}(M)$, $\alpha \neq 1/2$.* EXACT-K WINNER DETERMINATION *for α-proportion-approval with UI-FP profiles is* NP*-hard.*

Proof. The proof is similar to that of Theorem 6, by reductions from some particular GENERALIZED MAXSAT problems. UI-FP profiles correspond to logical formulas of the type $F(\mathcal{I}, \alpha)$ containing only positive literals, since the local preferences are always $1 \succ 0$: let $F(\mathcal{I}, \alpha)^+$ be the set of such formulas. Let CONSTRAINED MAX$(\alpha)^+$SAT be the following problem: given a set S of formulas in $F(\mathcal{I}, \alpha)^+$ and an integer K, find a valuation that assigns exactly K variables to true, and satisfies a maximum number of formulas in S. We prove by induction on the length of the base 2 representation of α that this problem is NP-hard for every $\alpha \in \mathrm{Rat}(M)$, $\alpha \neq 1/2$.

The base case corresponding to $\alpha = 1/4$ is a generalization of the NP-hard problem BOUNDED DENSEST SUBGRAPH (given a weighted graph and an integer k, find a subset of at most k vertices that induces a densest subgraph / maximizes the number of "hit" edges, see e.g. [40]).[4] For the sake of completeness, here is a simple reduction from MAXIMUM CLIQUE: given a graph $G = (\mathcal{I}, E)$, the set of formulas S_E of $F(1/4, \mathcal{I})$ is defined as $\{X_i \wedge X_j \mid (i, j) \in E\}$; and G has a clique of size J if and only if there is a valuation of $\mathcal{I}$ that assigns J variables to true, and satisfies at least $J(J-1)$ formulas of S_E.

The base case corresponding to $\alpha = 3/4$, where formulas are of the form $X_i \vee X_j$, is a generalization of the NP-hard MAX K-VERTEX COVER problem (see e.g. [41–43] for a proof of NP-hardness and recent approximation algorithms for MAX K-VERTEX COVER): given a graph G and an integer K, find a set of K vertices that covers (i.e., contains one of the extremities of) a maximum number of edges of G.

For the induction cases, let $\alpha = (0.\alpha_1 \ldots \alpha_M)_2$. Suppose first that $\alpha_1 = 0$; then $2\alpha = (0.\alpha_2 \ldots \alpha_M)_2$, and let $\Phi \subseteq F(\mathcal{I}, 2\alpha)^+$. Let $\mathcal{I}' = \mathcal{I} \cup \{X_0\}$, where X_0 is a fresh variable, and let $\Phi' = \{X_0 \wedge \phi \mid \phi \in \Phi\}$. Φ' is a set of formulas of $F(\mathcal{I}', \alpha)^+$; and one can satisfy J formulas of Φ assigning exactly K variables to true if and only if one can satisfy J formulas of Φ' assigning exactly $K+1$ variables to true.

Now, suppose that $\alpha_1 = 1$. Let $\alpha' = 2\alpha - 1 = (0.\alpha_2 \ldots \alpha_{M-1}\alpha_M)_2$ and let $\Phi \subseteq F(\mathcal{I}, 2\alpha - 1)^+$. We introduce $N = 1 + K|\Phi|$ fresh variables $Y_1, \ldots, Y_N$, and define $\mathcal{I}' = \mathcal{I} \cup \{Y_1, \ldots, Y_N\}$ and $\Phi' = \{Y_i \vee \phi \mid \phi \in \Phi, 1 \leq i \leq N\}$. Φ' is a set of formulas of $F(\mathcal{I}', \alpha)^+$. We claim that there is a valuation m of $\mathcal{I}$ assigning exactly K variables in $\mathcal{I}$ to true and satisfying at least J formulas in Φ if and only if there exists a valuation m' of $\mathcal{I}'$ assigning exactly K variables in $\mathcal{I}$ to true and satisfying at least NJ formulas in Φ'. If $K = |\mathcal{I}|$, this is straightforward. Assume now $K < |\mathcal{I}|$. For a valuation m of $\mathcal{I}$, let $|m|$ be the number of variables of $\mathcal{I}$ assigned to true by m.

- Assume there is a valuation m of $\mathcal{I}$ such that $|m| = K$ and satisfying at least J formulas in Φ. Then the extension m' of m to $\mathcal{I}'$ evaluating every Y_i to 0 satisfies exactly K variables of $\mathcal{I}'$, and at least NJ formulas of Φ', since for every $\phi \in \Phi$ and $i \leq N$, $m \models \phi$ implies $m' \models Y_i \vee \phi$.
- Conversely, assume there is a valuation m' of $\mathcal{I}'$ such that the number of variables of $\mathcal{I}'$ satisfied by m' is K and satisfying at least NJ formulas in Φ'. Let m be the restriction of m' to $\mathcal{I}$, and let λ be the number of formulas in Φ satisfied by m. Note that, since m and m' satisfy $|m|$ variables in $\mathcal{I}$, the number of Y_i's satisfied by m' is $K - |m|$. Consider a formula $\phi \in \Phi$: if $m \models \phi$, then m' satisfies the N formulas of the form $Y_i \vee \phi$ in Φ'; and if $m \not\models \phi$, then m' still satisfies $(K - |m|)$ formulas $Y_i \vee \phi$. Therefore, since m satisfies λ formulas of Φ, and falsifies $|\Phi| - \lambda$ of them, the number of formulas of Φ' satisfied by m' is $\lambda N + (|\Phi| - \lambda)(K - |m|) \leq \lambda N + K|\Phi| < \lambda N + N$ since $N = 1 + K|\Phi|$. Since we assumed that m' satisfies at least NJ formulas in Φ', we must have $NJ < N(\lambda + 1)$, therefore $J \leq \lambda$: the restriction of m' to $\mathcal{I}$ satisfies at least J formulas of Φ. □

Example 12 and Proposition 18 show that no majoritarian aggregation rule, when applied to the importance orders of the voters, can simulate a Condorcet-consistent rule applied to the set $\mathcal{I}[K]$. An aggregation rule f is majoritarian if for any profile V, if the pairwise majority relation[5] $\succ_V^m$ is transitive then $f(V) = succ_V^m$. The notion of Condorcet winner is defined with respect to the set of feasible alternatives, that is, an alternative $\mathbf{d} \in \mathcal{I}[K]$ is a Condorcet winner for profile V if and only if for any $\mathbf{d}' \in \mathcal{I}[K]$, a majority of votes in V prefers $\mathbf{d}$ to $\mathbf{d}'$.

Example 12. There are four items A, B, C, D, and $K = 2$. There are five UI-FP LP-trees: $LP_1 = LP_2 = A \rhd B \rhd C \rhd D$, $LP_3 = LP_4 = D \rhd C \rhd B \rhd A$, and $LP_5 = B \rhd C \rhd D \rhd A$. The majority relation is $B \rhd C \rhd D \rhd A$ and its dominating two elements are B and C. However, $\{A, D\}$ majority-dominates $\{B, C\}$.

Proposition 18. *Given an FP-UI profile V, an integer K and an alternative $\mathbf{d} \in \mathcal{I}[K]$, checking whether $\mathbf{d}$ is a Condorcet winner is* coNP*-complete.*

[4] It is more general because the same pair can appear in several LP trees.
[5] The pairwise majority relation $\succ_V^m$ associated with profile V is defined by: for all alternatives $\mathbf{d}$ to $\mathbf{d}'$, $\mathbf{d} \succ_V^m \mathbf{d}'$ if a majority of votes in V prefer $\mathbf{d}$ to $\mathbf{d}'$.

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

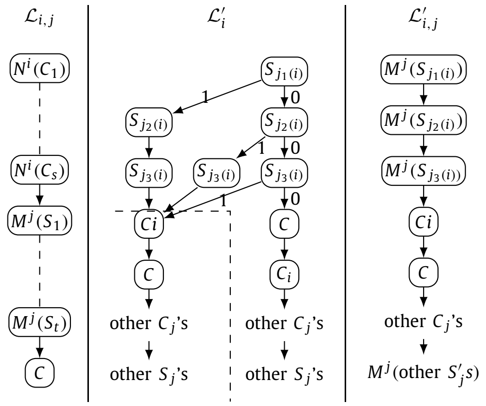

Fig. 7. Reduction of RESTRICTED X3C to K-EVALUATION for Borda with fixed preferences.

Proof. We give a reduction from HITTING SET to the complementary problem. An instance of HITTING SET consists of a collection of subsets $S_1, \dots, S_p$ of a set $S = \{1, \dots, s\}$, and an integer K; it is a "yes" instance if there exists a subset $S' \subseteq S$, $|S'| = K$, such that $S' \cap S_i \neq \emptyset$ holds for all i. With an instance of HITTING SET we associate the following instance of CONDORCET WINNER CHECKING FOR FP-UI PROFILES: the set of issues is $\mathcal{I} = \{X_1, \dots, X_s, Y_1, \dots, Y_K\}$, and the profile V is composed of $2p - 1$ FP-UI LP-trees:

- for $i = 1, \dots, p$, if $S_i = \{i_1, \dots, i_{|S_i|}\}$ then LP_i is defined by the unconditional preference relation $X_{i_1} \rhd \dots \rhd X_{i_{|S_i|}} \rhd Y_1 \rhd \dots \rhd Y_K \rhd$ Others;
- for $i = p + 1, \dots, 2p - 1$, LP_i is defined by the unconditional preference relation $Y_1 \rhd \dots \rhd Y_K \rhd$ Others

Suppose there is a hitting set S' of cardinality K for $\mathcal{S}$. Then the first p voters prefer $\{X_j, j \in S'\}$ to $\{Y_1, \dots, Y_K\}$, therefore $\{Y_1, \dots, Y_K\}$ is not a Condorcet winner.

Conversely, suppose $\{Y_1, \dots, Y_K\}$ is not a Condorcet winner. Then for some $C \in \mathcal{I}[K]$, $C \neq \{Y_1, \dots, Y_K\}$, at least p voters of V prefer C to $\{Y_1, \dots, Y_K\}$. But the last $p - 1$ voters have $\{Y_1, \dots, Y_K\}$ as their preferred committee of size K, therefore the other p must prefer C to $\{Y_1, \dots, Y_K\}$, which is the case only if C contains some X_{i_j} for all i; but then $\{i_j, i = 1, \dots, p\}$ is a hitting set of size $\leq K$ for $\mathcal{S}$, therefore there is a hitting set of size K for $\mathcal{S}$. □

8.2. FP profiles with conditional importance

When the importance order can be conditional, even the Borda rule becomes intractable.

Theorem 19. *K-EVALUATION for Borda with FP profiles is* NP*-hard.*

Proof. We prove NP-hardness by a reduction from RESTRICTED X3C, defined as follows: given a set $\mathcal{C} = \{C_1, \dots, C_s\}$, and a family $\mathcal{S} = \{S_1, \cdots, S_t\}$ of subsets of $\mathcal{C}$, with $|S_j| = 3$ for all j, and such that each $C_i \in \mathcal{C}$ appears in no more than three S_j's, we ask whether there is a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that each $c_i \in \mathcal{C}$ appears in exactly one $S_j \in \mathcal{S}'$ (which implies $|\mathcal{S}'| = s/3$); $\mathcal{S}'$ is called an exact cover of $\mathcal{C}$. RESTRICTED X3C is known to be NP-complete ([44]; problem [SP2] in [45]).

Given such an instance of RESTRICTED X3C, we now describe how to reduce it to an instance of K-EVALUATION with FP LP-trees. Let $\mathcal{I} = \mathcal{S} \cup \mathcal{C} \cup \{C\}$, where C is a fresh issue.

We define first a set of LP-trees where issues in $\mathcal{C}$ are more important than issues in $\mathcal{S}$, themselves more important than C. Specifically, we define the importance order

$$\mathcal{O} = [C_1 \rhd C_2 \rhd \dots C_s \rhd S_1 \rhd S_2 \rhd \dots \rhd S_t \rhd C]$$

and consider these two cyclic permutations:

- $M = S_1 \to S_2 \to \dots \to S_t \to S_1$ is a permutation over $\mathcal{S}$;
- $N = C_1 \to C_2 \to \dots \to C_s \to C_1$ is a permutation over $\mathcal{C}$.

For $1 \leq j \leq t$ and $1 \leq i \leq s$, let $\mathcal{L}_{i,j}$ be the LP-tree with fixed preferences and unconditional importance order $N^i(M^j(\mathcal{O}))$; it is depicted on the left of Fig. 7. We define the profile $V = \{\mathcal{L}_{i,j} | 1 \leq j \leq t, 1 \leq i \leq s\}$.

Next, for every $i \leq s$, we define another set of LP-trees, again depicted on Fig. 7, as follows: let $\mathcal{S}(C_i)$ be the set of elements of $\mathcal{S}$ that cover C_i: recall that $1 \leq |\mathcal{S}(C_i)| \leq 3$. Let $\mathcal{S}'(C_i)$ be some subset of $\mathcal{S}$ of cardinality 3 that contains $\mathcal{S}(C_i)$ (therefore, $\mathcal{S}'(C_i)$ is obtained by adding 0, 1 or 2 S_j's to $\mathcal{S}(C_i)$). Denote by $S_{j_1(i)}, S_{j_2(i)}, S_{j_3(i)}$ the elements of $\mathcal{S}'(C_i)$, with $j_1(i) > j_2(i) > j_3(i)$. We define the following two importance orderings over $\mathcal{I}$:

- $\mathcal{O}_i = [S_{j_1(i)} \rhd S_{j_2(i)} \rhd S_{j_3(i)} \rhd C_i \rhd C \rhd \mathcal{C} \setminus \{C_i\} \rhd \mathcal{S} \setminus \{S_{j_1(i)}, S_{j_2(i)}, S_{j_3(i)}\}]$
- $\mathcal{O}'_i = [S_{j_1(i)} \rhd S_{j_2(i)} \rhd S_{j_3(i)} \rhd C \rhd C_i \rhd \mathcal{C} \setminus \{C_i\} \rhd \mathcal{S} \setminus \{S_{j_1(i)}, S_{j_2(i)}, S_{j_3(i)}\}]$

The only difference between $\mathcal{O}_i$ and $\mathcal{O}'_i$ is that C_i and C are swapped. Let $\mathcal{L}'_i$ be the LP-tree with the importance order $\mathcal{O}_i$ in all branches except in the branches with $S_j = 0$ for every $S_j \in \mathcal{S}(C_i)$, where the importance order is $\mathcal{O}'_i$. For $1 \le j < t$, let $\mathcal{L}'_{i,j}$ be the LP-tree with unconditional importance order $M^j(\mathcal{O}_i)$. For $1 \le i \le s$, let $V_i = \{\mathcal{L}'_i\} \cup \{\mathcal{L}'_{i,j} \mid 1 \le j < t\}$. We define the profile $V' = \cup_{i=1}^{s} V_i$, which contains $s \times t$ trees.

Now, let $sV + V'$ be the profile that contains V' and s copies of V. We now analyze the Borda score of alternative $\mathbf{d}$ for this profile. Recall that, since all trees have height $t + s + 1$, if an issue X is at level l in a tree $\mathcal{L}$, and if $\mathbf{d}(X) = 1$, it contribution to $S_{\text{Borda}}(\mathcal{L}, \mathbf{d})$ will be $2^{t+s+1-l}$. In the sequel, we denote by $S_{\text{Borda}}(W, \mathbf{d}, X)$ the contribution of issue X to the Borda score of $\mathbf{d}$ for some profile W.

- Consider first issue $S_j \in \mathcal{S}$: because of the symmetries, it will appear s times at each level between $s + 1$ and $s + t$ in the $\mathcal{L}_{ij}$'s; thus

$$\begin{aligned} S_{\text{Borda}}(V, \mathbf{d}, S_j) &= \mathbf{d}(S_j)s(2^t + \ldots + 2^1) \\ &= \mathbf{d}(S_j)s(2^{t+1} - 2). \end{aligned}$$

  Similarly, in every V_i, S_j will appear at each level between 1 and 3 and also at each level between $s + 5$ and $t + s + 1$; thus, since there are s V_i's in V':

$$\begin{aligned} S_{\text{Borda}}(V', \mathbf{d}, S_j) &= \mathbf{d}(S_j)s(2^{t+s} + 2^{t+s-1} + 2^{t+s-2} + 2^{t-4} + 2^{t-5} + \ldots + 1) \\ &= \mathbf{d}(S_j)s(7 \times 2^{t+s-2} + 2^{t-3} - 1). \\ &= \mathbf{d}(S_j)s2^{t+s+1}\left(\frac{7}{8} + \frac{1}{2^{s+4}} - \frac{1}{2^{t+s+1}} \right). \end{aligned}$$

  Therefore:

$$\begin{aligned} S_{\text{Borda}}(sV + V', \mathbf{d}, S_j) &= \mathbf{d}(S_j)s2^{t+s+1}\left[1 - \frac{1}{8} + \frac{1}{2^{s+4}} + \frac{s}{2^s} - \frac{(1+2s)}{2^{t+s+1}}\right] \\ &< \mathbf{d}(S_j)s2^{t+s+1}(1 - \epsilon) \end{aligned}$$

  for any ϵ in $]0, \frac{1}{8}[$ and for large enough values of s.

For issues in $\mathcal{C}$ and for C, the contributions in each V_i depend on whether at least one $S_j \in \mathcal{S}(C_i)$ has value 1 in $\mathbf{d}$ or not; therefore we define $\mathbb{1}_i(\mathbf{d}) = 0$ if $S_j = 0$ for every $S_j \in \mathcal{S}(C_i)$, and $\mathbb{1}_i(\mathbf{d}) = 1$ otherwise.

- Consider now issue C: it appears at level $t + s + 1$ in every $\mathcal{L}_{ij}$, so

$$S_{\text{Borda}}(V, \mathbf{d}, C) = \mathbf{d}(C)st.$$

  Furthermore, C appears at level 5 in each of the $s \times (t-1)$ $\mathcal{L}'_{ij}$, and at level 5 or 4 in each $\mathcal{L}_i$, depending on whether $\mathbb{1}_i(\mathbf{d}) = 1$ or 0; thus

$$\begin{aligned} S_{\text{Borda}}(V', \mathbf{d}, C) &= \mathbf{d}(C)\Big[s(t-1)2^{t+s-4} \\ &\qquad + \sum_{i=1}^{s}\left[\mathbb{1}_i(\mathbf{d})2^{t+s-4} + (1 - \mathbb{1}_i(\mathbf{d}))2^{t+s-3}\right]\Big] \\ &= \mathbf{d}(C)2^{t+s-4}\Big[st + \sum_{i=1}^{s}(1 - \mathbb{1}_i(\mathbf{d}))\Big]. \end{aligned}$$

  In particular, $S_{\text{Borda}}(sV + V', \mathbf{d}, C) \le \mathbf{d}(C)\left[s^2t + s(t+1)2^{t+s-4}\right] < S_{\text{Borda}}(sV + V', \mathbf{d}, S_j)$ for any $S_j \in \mathcal{S}$ such that $\mathbf{d}(S_j) = 1$ and for large values of t and s.
- Finally, consider issue $C_i \in \mathcal{C}$: C_i will appear t times at each level between 1 and s in the $\mathcal{L}_{ij}$'s; thus

$$S_{\text{Borda}}(V, \mathbf{d}, C_i) = \mathbf{d}(C_i)t(2^{t+s} + \ldots + 2^{t+1}) = \mathbf{d}(C_i)t2^{t+1}(2^s - 1).$$

  C_i is at level 4 or 5 in $\mathcal{L}_i$, depending on whether $\mathbb{1}_i(\mathbf{d}) = 1$ or 0; and at level 4 in each of the $t - 1$ $\mathcal{L}'_{ij}$; it also appears t times at some level between 6 and $s + 4$ in every $V_{i'}$ when $i' \neq i$. Thus

$$S_{\mathsf{Borda}}(V', \mathbf{d}, C_i) = \mathbf{d}(C_i)\big[\, (\mathbb{1}_i(\mathbf{d}) + t - 1)2^{t+s-3} + (1 - \mathbb{1}_i(\mathbf{d}))2^{t+s-4} + t \sum_{\substack{i'=1 \\ i' \neq i}}^{s} 2^{t+s+1-\mathsf{level}(\mathcal{L}'_{i'}, C_i)}\big]$$

where $\mathsf{level}(\mathcal{L}'_{i'}, C_i)$ denotes the level of issue C_i in i'; it is independent of $\mathbf{d}$ when $i' \neq i$.
In particular, $S_{\mathsf{Borda}}(sV + V', \mathbf{d}, C_i) > S_{\mathsf{Borda}}(sV, \mathbf{d}, C_i) = \mathbf{d}(C_i) st 2^{t+s+1}(1 - \frac{1}{2^s}) > \mathbf{d}(C_i) st 2^{t+s+1}(1 - \epsilon)$ for any fixed $\epsilon \in]0, \frac{1}{8}[$ and for large values of t and s.

From this analysis, we can see that for any $C_i \in \mathcal{C}$, $S_j \in \mathcal{S}$, for large values of t and s, we have that $S_{\mathsf{Borda}}(sV + V', \mathbf{d}, C_i) > S_{\mathsf{Borda}}(sV + V', \mathbf{d}, S_j) > S_{\mathsf{Borda}}(sV + V', \mathbf{d}, C)$. Hence, if we fix $K = s + t/3$, and look for alternatives $\mathbf{d}$ that have no more than K issues equal to 1, and that maximize $S_{\mathsf{Borda}}(sV + V', \mathbf{d})$, then such a $\mathbf{d}$ will have value 1 for all of the s issues in $\mathcal{C}$, and value 1 for a third of the t issues in $\mathcal{S}$; it will have value 0 for all remaining issues in $\mathcal{S}$ and for C. Moreover, its Borda score for $sV' + V$ will be:

$$\begin{aligned} S_{\mathsf{Borda}}(sV' + V, \mathbf{d}) = {} & \frac{st}{3} 2^{t+s+1}\big[1 - \frac{1}{8} + \frac{1}{2^{s-4}} + \frac{s}{2^s} - \frac{(1+2s)}{2^{t+s+1}}\big] \\ & + st2^{t+1}(2^s - 1) \\ & + 2^{t+s-4}\big(2t - 1 + \sum_{i=1}^{s} \mathbb{1}_i(\mathbf{d})\big) \\ & + t \sum_{i=1}^{s} \sum_{\substack{i'=1 \\ i' \neq i}}^{s} 2^{t+s+1-\mathsf{level}(\mathcal{L}'_{i'}, C_i)} \end{aligned}$$

Now, suppose that the instance of RESTRICTED X3C has an exact cover: then it is possible to define $\mathbf{d}$ with value 1 for all C_i's, value 1 for 1/3rd of the S_j's – those that are necessary to cover $\mathcal{C}$, and 0 for all the other issues. Because the selected S_j's form an exact cover of $\mathcal{C}$, $\sum_{i=1}^{s} \mathbb{1}_i(\mathbf{d}) = s$, and no alternative with only K issues with value 1 can have a higher Borda score. Let $S^{\max}_{\mathsf{Borda}}(sV' + V)$ denote this Borda score: if there exists an exact cover for the RESTRICTED X3C instance, then there is an alternative with no more than K issues with value 1 and whose Borda score is at least $S^{\max}_{\mathsf{Borda}}(sV' + V)$. For the converse, suppose that an alternative $\mathbf{d}$ has no more than K issues with value 1 and is such that $S_{\mathsf{Borda}}(sV' + V, \mathbf{d}) \geq S^{\max}_{\mathsf{Borda}}(sV' + V)$. Then there must be an alternative $\mathbf{d}'$ with a score at least as high and where all C_i's have value 1, and 1/3rd of the S_j's have value 1, and all other issues have value 0. But then, it must be the case that $\sum_{i=1}^{s} \mathbb{1}_i(\mathbf{d}') = s$: we have found an exact cover of $\mathcal{C}$. □

9. Discussion

9.1. Committee elections

Committee elections are a major application field of our work, which we have only hinted on in Section 8. Here we discuss in more detail how our work can be applied to this field.

A *committee* (or *multiwinner*) voting rule maps a profile into a set of winners, usually of fixed size (corresponding to the number of seats to be filled in the committee). There are two major trends in multiwinner elections; we discuss below the applicability of our setting to each of both. In the first trend, profiles consist of *approval* ballots: each voter approves a subset of candidates, of arbitrary size (see [46] for a survey of a number of such rules and [47] for a very recent reference). In the second trend, profiles consist of rankings over candidates (see [38] for a very recent reference). In both cases, preferences are *unconditional*: voters are not allowed to approve or disapprove a candidate conditionally on some other set of candidates being elected or not. A possible generalization of both approval-based and ranking-based multiwinner elections consists in having voters express LP-trees for which *(binary) variables correspond to candidates*. For instance, the profile of Example 1 could be reinterpreted this way: **M** (respectively **F**, **P**) now means that *Mary* (respectively *Francis*, *Patricia*) *is elected*; LP_1 now means that the most important for voter 1 is that Mary is elected (maybe because voter 1 is Mary herself!); if she is, then the second most important thing is that Francis is elected too, and if not, that Patricia is elected, and so on. Also, what is most important to voter 2 is that Patricia is *not* elected.

Although any of the different assumptions corresponding to the different sections of our paper are applicable to multiwinner elections, we feel that the one that applies most directly is Section 8, where voters cannot express negative preferences for a candidate. Expressing LP-trees in this case is a generalization of expressing rankings over candidates, as a ranking is just a UI-FP LP-tree. In Section 8 we have seen that some well-known multiwinner rules can be captured as voting rules on lexicographic preferences, such as some K-best rules and perfectionist rules, all belonging to the family of committee scoring rules [38]. Now, allowing for FP-CI LP-trees leads to a generalized form of ranking-based multiwinner elections, where the ranking over candidates is conditional.

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

Can approval-based multiwinner rules be captured in our framework? Not as such, because we would have to be able to allow *LP-trees with indifferences*. In such an extension, not only approval ballots are expressible (with all variables being equally important), but a voter could also express conditional or unconditional (dis)approvals. Note however that in order to be adapted to conditional approval balloting, the general model of aggregating LP-trees needs to be enriched by a cardinality constraint such as we did in Section 8.

9.2. How restrictive are conditionally lexicographic preferences?

When the set of alternatives is combinatorial, the number of alternatives is exponential, therefore the number of possible rankings of alternatives is doubly exponential. This leads to this classical tension between expressivity and communication complexity (see for instance the introductory section of [24]): if we don't make any restriction on expressible preferences, then we must accept an exponential communication complexity which makes the mechanism infeasible in practice; and if we do accept a domain restriction, then only an exponentially small proportion of all preference relations will be expressible. The question is to find a domain restriction that is meaningful enough in some classes of situations. Each domain restriction comes with some way of expressing preferences in a succinct way. The most common such languages, with their associated domain restrictions, are the following:

1. each voter specifies a ranking over the domain of each variable, independently of other variables (*separable* preferences), or a weight function mapping every value of each variable to a number, the weight of a tuple of values then being the sum of the weights obtained for each variable (*additively separable* preferences).
2. each voter specifies a ranking over single issues.
3. each voter specifies only her best tuple of values.

For each such way of expressing preferences, the preference relation over the whole domain is then determined using a *preference extension principle*. Interestingly, LP-trees are a combination of 1 and 2: 1 for the local preferences and 2 for the importance relation. As far as we know, this is the first time 1 and 2 are combined in the setting of preference aggregation over combinatorial domains.

Moreover, the different families of LP-trees offer a wide span of ways of trading off expressivity and communication complexity. If expressivity matters before all, then unrestricted LP-trees allow for considering nonseparable preferences, via the expression of conditional preferences, and a (possibly conditional) importance relation between variables.[6] Note that *1, 2 and 3* are all restricted to separable preferences, and thus disallow the expression of preferential dependencies between variables, which is a huge drawback. Very few methods exist for aggregating preferences over combinatorial domains with preferential dependencies. Among the most prominent ones we find, on the one hand, sequential voting, and on the other hand, voting with *conditional preference networks* (CP-nets) — see [24] for a review. But sequential voting, among other drawbacks, needs voters to interact again and again with the system, while direct, one-shot methods suffer from the fact that the preference relation induced from a CP-net is generally a partial order, which makes it sometimes impossible to apply a known voting rule (which needs a collection of complete orders). Admittedly, some voting rules can be naturally extended to partial orders, and more precisely to partial order obtained from CP-nets (see [48] for extensions of Copeland, maximin and Kemeny for such profiles); but this is not true for most positional scoring rules such as Borda or k-approval (which are the focus of our paper), for which no extension to partial orders is commonly agreed to be natural. Thus, aggregating LP-trees is, as far as we know, the first direct aggregation method ever that allows nonseparable preferences and makes it possible to apply any classical voting rule.

Of course, expressivity comes with high communication cost: in the worst case, the size of an LP-tree is exponential in the number of variables. Note however that this happens only in the worst case: if voters have few dependencies for their preference and importance relations, then the size of their LP-trees will remain small. This may still be a problem, because in some contexts, the amount of communication between the voters and the voting centre must be very small (as voters are unlikely to be willing to report their preferences if this takes them too long), and then moving to a further restriction might prove useful: with unconditional importance and unconditional preference (UI-UP), an LP-tree needs only polynomial space to be expressed and the communication complexity of aggregating LP-trees becomes polynomial. For the exact communication complexity of expressing various classes of LP-trees see [13].

Q5 One possible way of quantifying how drastic a domain restriction is consists in counting the number of preference relations that are expressible. This, of course, does not mean that expressing more preferences means being more plausible.[7] Moreover, this way of quantifying a domain restriction may look controversial: if it is too restrictive, it is not a

[6] Even more expressivity can be obtained by considering, as [14], a generalization of LP-trees were a node can contain several variables, deemed to have comparable importance; the extension of our algorithms to such extended LP-trees is left for further research. Note that, as suggested by a reviewer, it is possible to define a measure of how close a preference order is to be captured by efficient LP-trees: a single node can represent anything in space exponential in the number of issues; then the preference order can be decomposed by adding nodes; the "LP-tree-width" can then be the number of issues in the largest node.

[7] An example suggested by a reviewer: single-peakedness with respect to a fixed axis is certainly more plausible than the weird domain restriction obtained by fixing the best and the worst alternatives, although the latter can express more preference relations.

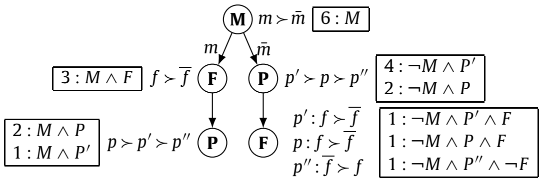

Fig. 8. LP tree LP'_1 and its translation into weighted formulas.

good restriction because too few preference relations can be expressed, but if it is not restrictive enough, then it is not good either because it induces a high communication complexity. Thus, in a sense, a reasonable domain restriction (at least for our purposes) should be neither too expressive nor too restrictive — and of course, it should also be *cognitively* reasonable.

Now, even in the highly restricted cases such as those considered in Sections 7 and 8, the number of expressible LP-trees is still larger than with the classical ways of expressing preferences in combinatorial domains reported in the above list, but exponentially smaller than the number of all preference relations. With fixed importance, there is still a doubly exponential number of expressible LP-trees due to the expression of conditional preferences, and with fixed preferences there is again a doubly exponential number of expressible LP-trees due to the expression of conditional importance. In contrast, with 1, 2 or 3 there is only a simply exponential number of expressible preferences. And furthermore, as said in the Introduction, the lexicographicity assumption, even in its most restricted forms, makes sense from a cognitive point of view.

9.3. Nonbinary domains

Although we restricted our study to the case of binary issues for the sake of clarity, let us briefly explain now why most results still hold if we allow non-binary issues. Note first that formula (1), that gives the rank of an alternative w.r.t. a given tree, can be generalized to non-binary domains, using products of the sizes of the domains D_i: given an alternative $\mathbf{d}$ and an LP-tree $\mathcal{L}$, if t_i is the node labelled with issue X_i in the branch of $\mathcal{L}$ corresponding to $\mathbf{d}$, then

- the local rank of the value of $\mathbf{d}$ for issue X_i in the local preference at t_i, still denoted $\Delta(\mathcal{L}, \mathbf{d}, X_i)$, is now a value between 0 (if d_i is the preferred value for X_i at node t_i, given the values of $\mathbf{d}$ for the issues that appear above t_i) and $|D_i| - 1$ (if d_i is the least preferred value for X_i at t_i);
- moreover, if we denote by $X_i \rhd_{\mathcal{L},\mathbf{d}} X_j$ the fact that X_i is above X_j in the branch of $\mathcal{L}$ corresponding to $\mathbf{d}$, then the number of alternatives that correspond to the leaves of $\mathcal{L}$ below t_i and to the left of $\mathbf{d}$ is $\Delta(\mathcal{L}, \mathbf{d}, X_i) \times \prod_{X_i \rhd_{\mathcal{L},\mathbf{d}} X_j} |D_j|$ (with the convention that, the empty product, when X_i is at a leaf, equals 1).

Thus

$$\mathsf{rank}(\mathcal{L}, \mathbf{d}) = 1 + \sum_{i=1}^{q} \Big(\Delta(\mathcal{L}, \mathbf{d}, X_i) \times \prod_{X_i \rhd_{\mathcal{L},\mathbf{d}} X_j} |D_j|\Big)$$

Therefore, computing, for a given LP-tree, the rank of a given alternative, or the alternative that has a given rank, can both be done, without significant increase in complexity, through a top-down traversal of the tree guided by the given alternative or the given rank. In a similar way, it is still possible to compute in polynomial time the Borda winner when importance and local preferences are both unconditional (Theorem 7), and to design a translation of winner determination into a WEIGHTED MINSAT instance – using weights that are not powers of 2 but that must again be computed through a traversal of each tree; to each issue with d values now correspond d propositional variables, and a polynomial number of "hard" clauses are added to enforce that only one of them can be true.

Consider for instance the tree LP_1, and assume that issue $\mathbf{P}$ now has three values p, p' and p''. Let LP'_1 be the LP-tree identical to LP_1 except that the values for $\mathbf{P}$ are ordered $p \succ p' \succ p''$ in the left branch, and $p' \succ p \succ p''$ in the right branch; it is depicted on Fig. 8. Then the formula corresponding to the root of LP_1 is still M, but now with a weight of $2 \times 3 = 6$; then, at the $\mathbf{F}$ node down the m branch, the formula is still $M \wedge F$ but with weight 3; there are now two formulas at the leaf of the m branch: $M \wedge P$ with weight 2, and $M \wedge P'$ with weight 1; alternative mfp' now satisfies M, $M \wedge F$ and $M \wedge P'$ and has a Borda score of 10 (mfp' is indeed the second best alternative for LP_1.

The α-proportion approval rule can still be defined, and the translation of WINNER DETERMINATION for this rule into a GENERALIZED MAXSAT instance can still be defined. For instance, if $\alpha = 0.8$, then assuming that we want the $\lfloor \alpha \times 12 \rfloor = 9 = (1/2 + 1/6 + 1/12) \times 12$ best alternatives, $\phi(LP'_1, \alpha) = M \vee (P' \vee (P \wedge F))$. Similarly, if $\alpha' = 0.45$, then assuming that we want the $\lfloor \alpha' \times 12 \rfloor = 5 = (0/2 + 1/4 + 2/12) \times 12$ best alternatives, $\phi(LP'_1, \alpha) = M \wedge (F \vee (P \vee P'))$. (Again there is one propositional variable for every value in the domain of every non-binary issue and hard clauses enforce that only one of them is true.) Note that the length of the decomposition of α depends on the order of the non-binary variables in the branches, so that the approach of Theorem 12 in the FCI case cannot work in this more general setting.

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

Finally, membership in NP of the problems we have studied here is not compromised when we consider non-binary issues, and hardness results obviously still hold.

10. Summary and future work

We first give three tables summarizing the results obtained for Borda, α-proportion approval, and checking whether an alternative is a Condorcet winner. In these tables, "CP" stands for "Conditional Preferences" (no restriction on the preference tables), "UP" means "Unconditional Preferences", whereas "FP" is an abbreviation for "Fixed Preferences": it is assumed that all voters prefer value 1 to 0 for every issue, and we studied the "constrained" case, where there is a bound on the number of issues that can have value 1. Similarly, "CI" and "UI" respectively stand for "Conditional Importance" (no restriction on the tree structure) and "Unconditional Importance" (all trees in the profile are linear). Finally, "FCI" and "FUI" indicate the restriction whereby all voters in the profile have the same tree structure.

- Borda:

	CP	UP	Constrained FP
CI	NP-hard (Theorem 10)	NP-hard (Theorem 10)	NP-hard (Theorem 19)
UI	NP-hard (Theorem 10)	P (Theorem 7)	P (Theorem 16)
FCI	NP-hard (Theorem 10)	P (Theorem 13)	P (Observation 15)
FUI	NP-hard (Theorem 10)	P (Theorem 7)	P (Observation 15)

- α-proportion approval ($\alpha \neq \frac{1}{2}$)[8]:

	CP	UP	Constrained FP
CI	NP-hard (Theorem 6)	NP-hard (Theorem 6)	NP-hard (Theorem 17)
UI	NP-hard (Theorem 6)	NP-hard (Theorem 6)	NP-hard (Theorem 17)
FCI	P (Theorem 12)	P (Theorem 12)	P (Observation 15)
FUI	P (Theorem 12)	P (Theorem 12)	P (Observation 15)

- Condorcet winner checking:

	CP	UP	Constrained FP
CI	coNP-hard (Theorem 11)	coNP-hard (Theorem 11)	coNP-hard (Theorem 18)
UI	coNP-hard (Theorem 11)	coNP-hard (Theorem 11)	coNP-hard (Theorem 18)
FCI	P [35]	P [35]	P (Observation 15)
FUI	P [3]	P [3]	P (Observation 15)

Our conclusions are partly positive, partly negative. On the one hand, there are voting rules for which the restriction to conditionally lexicographic preferences brings significant simplifications: this is the case, at least, for k-approval for some values of k. The Borda rule can be applied easily provided that both the importance relation and the local preferences are unconditional, which is a very strong restriction. The hardness of checking whether an alternative is a Condorcet winner suggests that Condorcet-consistent rules will be hard to apply as well, which is why we did not focus on them in more detail.

However, we have shown that some of these problems can be reduced to a compact MAXSAT problem. From a practical point of view, it is important to test the performance of MAXSAT solvers on these problems. Liu and Truszcziński [21] have proposed translations of winner determination and evaluation problems for the Borda and k-approval rules into Answer Set Programming, another general paradigm for solving hard combinatorial optimization problems. They showed that winners can be determined in a few minutes for these rules with 10 issues and 1000 voters, or 15 issues and 500 voters.[9] We believe that continuing studying preference representation and aggregation on combinatorial domains, taking advantages of developments in efficient satisfiability techniques, is a promising future work direction.

Although the terminology we used in the paper, as well as the choice of our examples, referred to preference aggregation, our approach can also be relevant to *epistemic* social choice, where the goal is no longer to aggregate preferences but to search for the most plausible state of affairs by aggregating plausibility orders. Here, the lexicographicity assumption makes sense when an agent considers a state of the world s more plausible than another one s' when s has a value considered more plausible than s' for her most important variable for which s and s' take different values.

[8] For $\alpha = \frac{1}{2}$, the problem is in P in all cases (Theorem 3).
[9] They also proved a hardness result for a variant of k-approval.

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

Acknowledgements

We thank all anonymous reviewers of AAAI-12, CP-12, COMSOC-12 and the Artificial Intelligence journal for their helpful comments and suggestions. This work has been partly supported by the ANR project CoCoRICo-CoDec. Lirong Xia acknowledges supports from NSF under Grant #1136996 to the Computing Research Association for the CIFellows Project, NSF under Grant #1453542 and #1716333, and ONR #N00014-17-1-2621.

References

[1] S. Brams, D. Kilgour, W. Zwicker, The paradox of multiple elections, Soc. Choice Welf. 15 (2) (1998) 211–236.
[2] D. Lacy, E. Niou, A problem with referenda, J. Theor. Polit. 12 (1) (2000) 5–31.
[3] J. Lang, L. Xia, Sequential composition of voting rules in multi-issue domains, Math. Soc. Sci. 57 (3) (2009) 304–324.
[4] G.D. Pozza, M.S. Pini, F. Rossi, K.B. Venable, Multi-agent soft constraint aggregation via sequential voting, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 172–177.
[5] S. Airiau, U. Endriss, U. Grandi, D. Porello, J. Uckelman, Aggregating dependency graphs into voting agendas in multi-issue elections, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 18–23.
[6] G. Gigerenzer, D. Goldstein, Reasoning the fast and frugal way: models of bounded rationality, Psychol. Rev. 103 (4) (1996) 650–669.
[7] M. Schmitt, L. Martignon, On the complexity of learning lexicographic strategies, J. Mach. Learn. Res. 7 (2006) 55–83.
[8] N.M. Fraser, Applications of preference trees, in: Proceedings of the IEEE Conference on Systems, Man and Cybernetics, 1993, pp. 132–136.
[9] N.M. Fraser, Ordinal preference representations, Theory Decis. 36 (1) (1994) 45–67.
[10] N. Wilson, An efficient upper approximation for conditional preference, in: Proceedings of the 17th European Conference on Artificial Intelligence, ECAI 2006, 2006, pp. 472–476.
[11] N. Wilson, Efficient inference for expressive comparative preference language, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, 2009, pp. 961–966.
[12] R.J. Wallace, N. Wilson, Conditional lexicographic orders in constraint satisfaction problems, Ann. Oper. Res. 171 (1) (2009) 3–25.
[13] R. Booth, Y. Chevaleyre, J. Lang, J. Mengin, C. Sombattheera, Learning conditionally lexicographic preference relations, in: Proceedings of the 19th European Conference on Artificial Intelligence, ECAI 2010, IOS Press, 2010, pp. 269–274.
[14] M. Bräuning, E. Hüllermeyer, Learning conditional lexicographic preference trees, in: J. Fürnkranz, E. Hüllermeyer (Eds.), Preference Learning: Problems and Applications in AI. Proceedings of the ECAI 2012 Workshop, 2012, pp. 11–15.
[15] M. Bräuning, E. Hüllermeier, T. Keller, M. Glaum, Lexicographic preferences for predictive modeling of human decision making: a new machine learning method with an application in accounting, Eur. J. Oper. Res. 258 (1) (2017) 295–306.
[16] X. Liu, M. Truszczynski, Learning partial lexicographic preference trees over combinatorial domains, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, AAAI Press, 2015, pp. 1539–1545.
[17] M. Taylor, The problem of salience in the theory of collective decision-making, Behav. Sci. 15 (5) (1970) 415–430.
[18] P.K. Pattanaik, Group choice with lexicographic individual orderings, Behav. Sci. 18 (2) (1973) 118–123.
[19] J. Bhadury, P.M. Griffin, S.O. Griffin, L.S. Narasimhan, Finding the majority-rule equilibrium under lexicographic comparison of candidates, Soc. Choice Welf. 15 (4) (1998) 489–508.
[20] J.J. Encarnacion, Group choice with lexicographic utility, Eur. J. Polit. Econ. 8 (3) (1992) 419–425.
[21] X. Liu, M. Truszczynski, Aggregating conditionally lexicographic preferences using answer set programming solvers, in: Algorithmic Decision Theory, Springer, 2013, pp. 244–258.
[22] J. Lang, J. Mengin, L. Xia, Aggregating conditionally lexicographic preferences on multi-issue domains, in: Milano [49], pp. 973–987.
[23] R.Q. Dividino, G. Gröner, S. Scheglmann, M. Thimm, Ranking RDF with provenance via preference aggregation, in: Knowledge Engineering and Knowledge Management – 18th International Conference, EKAW 2012, Galway City, Ireland, October 8–12, 2012, Proceedings, 2012, pp. 154–163.
[24] J. Lang, L. Xia, Voting in combinatorial domains, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A.D. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016, pp. 197–222.
Q6 [25] A. Abramé, D. Habet, AHMAXSAT: description and evaluation of a branch and bound Max-SAT solver, J. Satisf. Boolean Model. Comput.
[26] R. Martins, V.M. Manquinho, I. Lynce, Open-wbo: a modular maxsat solver, in: C. Sinz, U. Egly (Eds.), Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing, SAT 2014, in: Lecture Notes in Computer Science, vol. 8561, Springer, 2014, pp. 438–445.
[27] C. Luo, S. Cai, W. Wu, Z. Jie, K. Su, CCLS: an efficient local search algorithm for weighted maximum satisfiability, IEEE Trans. Comput. 64 (7) (2015) 1830–1843.
[28] A. Kügel, Natural max-sat encoding of min-sat, in: Y. Hamadi, M. Schoenauer (Eds.), Revised Selected Papers from the 6th International Conference on Learning and Intelligent Optimization, in: Lecture Notes in Computer Science, vol. 7219, Springer, 2012, pp. 431–436.
[29] Z. Zhu, C.M. Li, F. Manyà, J. Argelich, A new encoding from minsat into maxsat, in: Milano [49], pp. 455–463.
[30] C.M. Li, Z. Zhu, F. Manyà, L. Simon, Optimizing with minimum satisfiability, Artif. Intell. 190 (2012) 32–44.
[31] J. Argelich, C.M. Li, F. Manyà, Z. Zhu, MinSAT versus MaxSAT for optimization problems, in: Proceedings of the 19th International Conference on Principles and Practice of Constraint Programming, CP 2013, in: Lecture Notes in Computer Science, vol. 8124, Springer, 2013, pp. 133–142.
[32] M.R. Garey, D.S. Johnson, L.J. Stockmeyer, Some simplified np-complete graph problems, Theor. Comput. Sci. 1 (3) (1976) 237–267.
[33] R. Kohli, R. Krishnamurti, P. Mirchandani, The minimum satisfiability problem, SIAM J. Discrete Math. 7 (2) (1994) 275–283.
[34] B. Jaumard, B. Simeone, On the complexity of the maximum satisfiability problem for Horn formulas, Inf. Process. Lett. 26 (1) (1987) 1–4.
[35] L. Xia, V. Conitzer, Strategy-proof voting rules over multi-issue domains with restricted preferences, in: Internet and Network Economics – 6th International Workshop, WINE, Proceedings, 2010, pp. 402–414.
[36] U. Grandi, U. Endriss, Lifting integrity constraints in binary aggregation, Artif. Intell. 199 (2013) 45–66.
[37] P. Faliszewski, P. Skowron, A. Slinko, N. Talmon, Committee scoring rules: axiomatic classification and hierarchy, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, 2016, pp. 250–256.
[38] E. Elkind, P. Faliszewski, P. Skowron, A. Slinko, Properties of multiwinner voting rules, Soc. Choice Welf. 48 (3) (2017) 599–632.
[39] P. Faliszewski, P. Skowron, A.M. Slinko, N. Talmon, Multiwinner analogues of the plurality rule: axiomatic and algorithmic perspectives, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA, 2016, pp. 482–488.
[40] G. Kortsarz, D. Peleg, On choosing a dense subgraph (extended abstract), in: Proceedings of the 34th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1993, pp. 692–701.
[41] D. Marx, Parameterized complexity and approximation algorithms, Comput. J. 51 (1) (2008) 60–78.
[42] F.D. Croce, V.T. Paschos, Efficient algorithms for the max k-vertex cover problem, J. Comb. Optim. 28 (3) (2014) 674–691.
[43] É. Bonnet, V.T. Paschos, F. Sikora, Parameterized exact and approximation algorithms for maximum k-set cover and related satisfiability problems, RAIRO Theor. Inform. Appl. 50 (3) (2016) 227–240.

[44] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Proceedings of a Symposium on the Complexity of Computer Computations, in: The IBM Research Symposia Series, Plenum Press, New York, 1972, pp. 85–103.
[45] M. Garey, D. Johnson, Computers and Intractability, W. H. Freeman and Company, 1979.
[46] D.M. Kilgour, Approval balloting for multi-winner elections, in: Handbook on Approval Voting, 2010, pp. 105–124.
[47] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, T. Walsh, Justified representation in approval-based committee voting, Soc. Choice Welf. 48 (2) (2017) 461–485.
[48] V. Conitzer, J. Lang, L. Xia, Hypercubewise preference aggregation in multi-issue domains, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 158–163.
[49] M. Milano (Ed.), Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming, CP 2012, Lecture Notes in Computer Science, vol. 7514, Springer, 2012.

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004

Sponsor names

Do not correct this page. Please mark corrections to sponsor names and grant numbers in the main text.

ANR, *country*=France, *grants*=CoCoRICo-CoDec

NSF, *country*=United States, *grants*=1136996

NSF, *country*=United States, *grants*=1453542, 1716333

ONR, *country*=United States, *grants*=N00014-17-1-2621

Please cite this article in press as: J. Lang et al., Voting on multi-issue domains with conditionally lexicographic preferences, Artif. Intell. (2018), https://doi.org/10.1016/j.artint.2018.05.004