How Hard is it to Explain Preferences Using Few Boolean Attributes?

Clemens Anzinger, Jiehua Chen, Christian Hatschka, Manuel Sorge, Alexander Temper

Institute of Logic and Computation, TU Wien, Vienna, Austria
{jchen, chatschka, msorge}@ac.tuwien.ac.at

Abstract

We study the computational complexity of explaining preference data through Boolean attribute models (BAMs), motivated by extensive research involving attribute models and their promise in understanding preference structure and enabling more efficient decision-making processes. In a BAM, each alternative *has* a subset of Boolean attributes, each voter *cares* about a subset of attributes, and voters prefer alternatives with more of their desired attributes. In the BAM problem, we are given a preference profile and a number k, and want to know whether there is a Boolean k-attribute model explaining the profile.

We establish a complexity dichotomy for the number of attributes k: BAM is linear-time solvable for $k \leq 2$ but NP-complete for $k \geq 3$. The problem remains hard even when preference orders have length two. On the positive side, BAM becomes fixed-parameter tractable when parameterized by the number of alternatives m. For the special case of two voters, we provide a linear-time algorithm.

We also analyze variants where partial information is given: When voter preferences over attributes are known (BAM WITH CARES) or when alternative attributes are specified (BAM WITH HAS), we show that for most parameters BAM WITH CARES is more difficult whereas BAM WITH HAS is more tractable except for being NP-hard even for one voter.

1 Introduction

What patterns underlie sets of human (and other) preferences in their various applications? Preferences in practice are not arbitrary orderings; people prefer alternatives that better satisfy their underlying criteria or possess characteristics they desire, and the distributions of such criteria and their desirability are structured. For one example, in a speed-dating experiment, Fisman et al. (2006) found that on average the recruited women preferred intelligent partners from more affluent neighborhoods, men preferred physically attractive partners, and both men and women accepted partners from less densely populated areas. We study the problem of how to uncover such structure from the pure revealed preferences.

A parsimonious and interpretable model that captures the above dynamic between criteria and their desirability arises from binary characteristics. In this framework, each alternative possesses some subset of Boolean attributes (e.g., a restaurant either accepts credit cards or does not), and each individual cares about some subset of these attributes. An individual prefers alternative a to alternative b if and only if a possesses more of the attributes they care about than b does. For instance, if someone cares about credit card acceptance and vegan options and restaurant A has both while restaurant B has only vegan options, then A is preferred to B.

Given a preference profile $\mathcal{P}$ (a set of preferences of voters over alternatives), we are hence interested in a *Boolean attribute model (BAM)* for $\mathcal{P}$, that is, (i) a set of attributes, (ii) an assignment has of a subset of attributes to each alternative, and (iii) an assignment cares of a subset of attributes to each voter such that, for each voter v, all of its pairwise preferences $a \succ b$ are explained by a having more attributes that v cares about than b. For parsimony we aim to minimize the number of attributes in such models. We denote by BOOLEAN ATTRIBUTE MODEL (BAM) the computational problem where we are given a preference profile and an integer k and want to determine whether a Boolean attribute model with at most k attributes exists.

Boolean attribute models and natural continuous variants are frequent in the literature (see Fisman et al. (2006); Bhatnagar, Greenberg, and Randall (2008); Künnemann et al. (2019); Cheng and Rosenbaum (2023) for just a few examples). However, we are not aware of work that considers *computing* such attribute models (and checking how well they fit the data). Our goal here is to contribute towards this direction by studying the computational complexity of this problem, with the goal of identifying hard and tractable cases and thereby informing practical algorithm design.

Obtaining (Boolean) attribute models confers several benefits. First, it is intrinsically interesting to check whether practical preference profiles have attribute models with few attributes as this would contribute to our understanding of (human) preference formation and decision making. However, we currently lack the necessary algorithmic research.

Second, knowing that an application allows for parsimonious attribute models allows us to restrict the domains of preferences that we consider when understanding properties of decision-making processes and designing algorithms. This enables more scalable algorithms: For instance, computing stable matchings generally needs quadratic time in

Parameter	BAM		BAM WITH CARES		BAM WITH HAS	
In general	NPc	[T1]	NPc	[T5]	NPc	[T9]
n	? $O(m)$ if $n = 2$		?		NPh $(n \geq 1)$	[T9]
m	FPT	[T3]	NPh $(m \geq 3)$	[T5]	FPT	[T10]
k	NPh	[T1]	NPh $(k \geq 6)$	[T6]	FPT	[T11]
$n + m$	–	–	FPT	[T7]	–	–
$n + k$	FPT	[C1]	FPT	[C2]	–	–
$m + k$	–	–	FPT	[T8]	–	–

Table 1: Result overview; n is the number of voters, m the number of alternatives, and k an upper bound on the number of attributes. "-" means the FPT result follows from the result for the single parameter m. "NPh ($p \geq c$)" means that the corresponding problem remains NP-hard even if the parameter p has constant value c.

the number n of agents but if a Boolean k-attribute model is known, then they can be computed in $O(4^k \cdot n \cdot (k + \log n))$ time, that is, almost linear time for small k (Künnemann et al. 2019).

Third, if we know that certain applications admit small attribute models, then preference learning (Fürnkranz and Hüllermeier 2010, 2017), more specifically eliciting preferences from voters, becomes more tractable: Instead of having to ask voters for rankings of the (potentially many) alternatives, we can ask them which (of the few) attributes are important to them. For instance, Benabbou et al. (2016) provide an application of this principle to compute Borda winners efficiently. See also Feffer et al. (2023) for a recent survey of preference elicitation in the context of participatory machine learning.

Our contribution. In general, BAM is NP-hard, which we show via a reduction from computing proper graph colorings. Given this general hardness, we (a) investigate more closely what parameters $p \in \mathbb{N}$ of the input make the problem hard or tractable and (b) check how additional information given in the input affects the complexity. An overview of most of our results is given in Table 1.

As to (a), we obtain either the favorable fixed-parameter tractability (FPT), meaning that the problem is solvable in $f(p) \cdot |I|^{O(1)}$ time, where $|I|$ is the input size. In other words, there are efficient algorithms for small parameter values. Or we obtain NP-hardness even for constant values of the parameter p. As parameters p we start with the natural candidates: the number n of voters, the number m of alternatives, and the number k of attributes. For k, we give a dichotomy for BAM, showing NP-hardness for computing Boolean 3-attribute models (even for preference orders of length at most two), whereas for two attributes the problem is polynomial-time solvable via a reduction to 2-SAT. For parameter m, the problem turns out to be FPT. For $n = 2$ voters we show that the problem can be solved in linear time by deriving lower bounds on the needed number of attributes and showing that there is always an optimal solution that meets these lower bounds exactly. We leave the complexity wrt. parameter n as an open question, but when combining n with k we obtain FPT. All these results are given in Section 3.

As to (b), in some applications BAMs are partially given already, that is, we may already know (i) which attributes voters care about, leading to the problem BAM WITH CARES or (ii) which attributes the alternatives have, corresponding to problem BAM WITH HAS. We show that also both these problems are NP-hard and we analogously subject them to a parameterized complexity analysis. Here, generally BAM WITH CARES turns out to be harder than BAM (see Section 4) whereas BAM WITH HAS is easier, up to the parameter n, where BAM WITH HAS is NP-hard already for one voter (see Section 5).

Related work. Attribute models of preferences are fundamental and widely used in decision theory (Yu 1985; Siskos, Grigoroudis, and Matsatsinis 2016), social choice theory (Lang and Xia 2016), and machine learning (Labreuche 2011). Attribute models have also been used in McFadden's conditional logit model to statistically predict discrete choices of individuals (McFadden 1974) and as a decision-making heuristic of Tversky's elimination by aspects (Tversky 1972). Often the attributes involve larger domains but Boolean domains are simple, interpretable, and cover relevant basic cases (Lang and Xia 2009).

To relate to other works below it is useful to consider the following alternative view of BAMs. Implicitly, in a k-attribute BAM we define a utility function for each voter: We associate each alternative a (resp. each voter v) with a vector $\mathsf{has}(a) \in \{0,1\}^k$ (resp. a vector $\mathsf{cares}(v) \in \{0,1\}^k$). The utility of v for a is the scalar product of $\mathsf{has}(a)$ and $\mathsf{cares}(v)$, i.e., the weighted sum $\sum_{i \in [k]} \mathsf{has}(a)[i] \cdot \mathsf{cares}(v)[i]$. The so-defined utilities must rise strictly with voter's preferences. Thus, having a k-attribute BAM amounts to learning, for each voter, a utility function that is the sum of utilities for the individual attributes.

A further motivation of BAMs comes from explaining AI systems' decisions (Holzinger et al. 2020): Here, the input preferences may stem from black-box machine-learning models. We may even know the attributes (features) that each of the alternatives have and were taken into consideration by the models. The task is then to select the attributes that the models relied on in their decision, that is, what attributes they cared about (Labreuche 2011).

A similar set of tasks stems from multi-criteria decision making (Yu 1985) and preference learning. Herein, one or multiple (Auriau et al. 2024) decision makers reveal their preferences and we try to learn their utility functions. So-called UTA methods (UTilités Additives or additive utilities in English) (Siskos, Grigoroudis, and Matsatsinis 2016) are popular for this task: We assume that the utility functions are weighted sums over the attributes and we try to learn the underlying attributes and the utility gained from each attribute. In a setting with multiple decision makers we additionally want to cluster the input preferences and learn a utility function for each decision maker over his corresponding cluster (Auriau et al. 2024). In BAMs we also try to learn simple additive utility functions from revealed preferences but we do not cluster the preferences directly (although a clustering may be derived from voters with similar cares-mappings).

In social choice theory, voting in combinatorial domains (Lang and Xia 2016) is loosely related to BAMs. Here, we consider the alternatives to be tuples of attribute values; often the attributes are Boolean, corresponding to multiple-issue voting. The voters express (usually complete) preferences over alternatives. Generally, straightforward issue-wise (attribute-wise) voting rules then lead to undesirable outcomes and one tries to find better voting rules. In contrast, in BAMs we assume that voters can only approve or disapprove of each attribute and we may have incomplete preferences. Crucially though, in combinatorial voting it is usually assumed that an attribute model is given whereas we want to compute such models.

Finally, preferences with (Boolean) few-attribute models form a restricted preference domain. Such domains and how they can allow for more efficient decision-making procedures are a mainstay of social choice theory (Elkind, Lackner, and Peters 2022). The group-separable preference domain (Inada 1964, 1969) could be interpreted as having a sequence of binary attributes with exponentially decreasing weights and voters evaluate attributes positively or negatively. More closely related are the geometric preference under ℓ_p norm (Chen, Pruhs, and Woeginger 2017; Peters 2017; Chen et al. 2022) where the voters and alternatives are located in a geometric space and voters prefer an alternative that is closer to him (under ℓ_p norm). Another popular domain restriction are so-called single-peaked preferences. However, there is little evidence that real-world preferences are single-peaked (Sui, Francois-Nienaber, and Boutilier 2013; Przedmojski 2016) and thus researchers called for investigation into multi-(but low-)-dimensional variants in particular of single-peaked domains (Barberá, Gul, and Stacchetti 1993). In a sense BAMs constitute one of the simplest multi-dimensional restricted preference domains and thus may inform investigation into more sophisticated variants.

2 Preliminaries

(Full) proofs for results marked by ($\star$) are deferred to the full version of the paper (Anzinger et al. 2025). In this section, we present necessary concepts for our Boolean attribute model and collect some fundamental properties thereof. Given a non-negative integer $t \in \mathbb{N}$, let $[t]$ denote the set $\{1, \ldots, t\}$. We assume basic knowledge of parameterized complexity and refer to the textbook by Cygan et al. (2015) for more details.

Preference profiles and BAMs. A *preference profile* (*profile* in short) is a triple $\mathcal{P} = (\mathcal{C}, \mathcal{V}, \mathcal{R})$, consisting of a set $\mathcal{C}$ of m alternatives, a set $\mathcal{V} = \{v_1, \ldots, v_n\}$ of n voters, and a collection $\mathcal{R} = (\succ_{v_1}, \ldots, \succ_{v_n})$ of (possibly incomplete) *strict preference orders* such that each $\succ_i$, is a linear order of a subset of $\mathcal{C}$ and represents the preferences (aka. ranking) of voter v_i over some subset of $\mathcal{C}$, $i \in [n]$. For instance, for $\mathcal{C} = \{1, 2, 3, 4\}$, an incomplete preference order of a voter v can be $3 \succ 1 \succ 2$; we omit the subscript if it is clear from the context which voter we refer to. This means that voter v prefers 3 to 1, and 1 to 2. Alternative 4 is not ranked in his preference order. The *rank* of an alternative c for a voter v is the number of alternatives that v prefers to c, i.e., $r_v(c) := |\{b \mid b \succ_v c\}|$. $r_v(c)$ is undefined if v's preference order does not rank c. The *length* $|\succ_v|$ of v's preference order is the number of alternatives in $\succ_v$. For $3 \succ 1 \succ 2$, we infer that $r_v(2) = 2$ and $|\succ| = 3$.

A *Boolean attribute model* for a profile $\mathcal{P} = (\mathcal{C}, \mathcal{V}, \mathcal{R})$ specifies for each alternative which attributes it has and for each voter which attributes he cares about. More specifically, let $\mathcal{M} = (\mathsf{AT}, \mathsf{has}, \mathsf{cares})$ be a triple, where AT denotes a set of attributes, and $\mathsf{has}\colon \mathcal{C} \to 2^{\mathsf{AT}}$ and $\mathsf{cares}\colon \mathcal{V} \to 2^{\mathsf{AT}}$ two attribute functions for the alternatives and voters, respectively. We say that an alternative $c \in \mathcal{C}$ has an attribute α if $\alpha \in \mathsf{has}(c)$ (or have, depending on grammatical necessity), and that c has ℓ attributes if $|\mathsf{has}(c)| = \ell$. A similar convention applies for the use of **cares**.

The *score* of a voter v for an alternative c under $\mathcal{M}$ is defined as $\mathrm{score}_v^{\mathcal{M}}(c) := |\mathsf{has}(c) \cap \mathsf{cares}(v)|$. We omit the superscript $\mathcal{M}$ from the score if it is clear from the context which $\mathcal{M}$ we refer to. We say that $\mathcal{M}$ *explains* the preference order of voter $v \in \mathcal{V}$ if for each two alternatives c and d with $c \succ_v d$ it holds that $\mathrm{score}_v(c) > \mathrm{score}_v(d)$. Note that we do not require the backward implication to hold since not every alternative is ranked. Accordingly, we say that $\mathcal{M}$ is a *Boolean attribute model* (*BAM* for short) for $\mathcal{P}$ if it explains the preference order of every voter.

Intuitively, the model aims to explain the voters' preferences by assigning each voter a set of attributes they care about, and assigning each alternative a set of attributes it possesses. Voters then prefer alternatives that possess more attributes they care about. If $|\mathsf{AT}| = k$, then $\mathcal{M}$ is also referred to as a *k-Boolean attribute model* (*k-BAM*).

Remark 1. *For brevity's sake, we sometimes also use the following vector representation for a k-BAM $\mathcal{M}$. The* $\mathsf{has}\colon \mathcal{C} \to \{0,1\}^k$ *(resp.* $\mathsf{cares}\colon \mathcal{V} \to \{0,1\}^k$*) function defines for each alternative $c \in \mathcal{C}$ (resp. each voter $v \in \mathcal{V}$) a binary vector of length k, where the attribute set is simply $[k]$ such that a 1 at coordinate z means the alternative* has *(resp. the voter* cares *about) attribute z.*

The score of alternative $c \in \mathcal{C}$ by a voter $v \in \mathcal{V}$ is given by the scalar product: $\mathrm{score}_v^{\mathcal{M}}(c) := \mathsf{has}(c) \cdot \mathsf{cares}(v)$.

Generally, our goal is to find a BAM that explains the voters' preferences with as few attributes as possible. We define the three decision problems that form the core of our work:

BAM
Input: A profile $\mathcal{P}$ and a non-negative integer k.
Question: Is there a k-Boolean attribute model for $\mathcal{P}$?

BAM WITH CARES (RESP. BAM WITH HAS)
Input: A preference profile $\mathcal{P} = (\mathcal{C}, \mathcal{V}, \mathcal{R})$ and a function $\mathsf{cares}\colon \mathcal{V} \to 2^{[k]}$ (resp. $\mathsf{has}\colon \mathcal{C} \to 2^{[k]}$).
Question: Is there a function $\mathsf{has}\colon \mathcal{C} \to 2^{[k]}$ (resp. $\mathsf{cares}\colon \mathcal{C} \to 2^{[k]}$) s.t. $(\mathsf{has}, \mathsf{cares})$ is a k-BAM for $\mathcal{P}$?

Clearly, our central problems are contained in NP:

Observation 1. *Checking whether* $(\mathsf{AT}, \mathsf{has}, \mathsf{cares})$ *explains a profile $\mathcal{P} = (\mathcal{C}, \mathcal{V}, \mathcal{R})$ is doable in polynomial time.*

This follows from the fact that one can determine the score of an alternative for a voter in polynomial time, and then iterate over every voter and every pair of alternatives to check if the voter's preference order is explained.

Fundamental properties. We now consider structural properties of BAMs. First, we provide some bounds on the number of attributes cared about by a voter (resp. possessed by an alternative).

Lemma 1 ($\star$). *For each k-BAM that explains a preference profile $\mathcal{P} = (\mathcal{C}, \mathcal{V}, \mathcal{R})$, the following holds.*
(i) For all $v \in \mathcal{V}$, it holds that $|\succ_v| - 1 \leq |\mathsf{cares}(v)| \leq k$.
(ii) For all $v \in \mathcal{V}$ and all c in $\succ_v$, it holds that $|\succ_v| - r_v(c) - 1 \leq |\mathsf{has}(c)| \leq k - r_v(c)$.

We can also observe that each k-BAM must be able to explain the difference in ranks of an alternative in two preference orders. This leads to the following lemma.

Lemma 2 ($\star$). *If $\mathcal{P} = (\mathcal{C}, \mathcal{V}, \mathcal{R})$ admits a k-BAM, then for all $c \in \mathcal{C}$, and $v, w \in \mathcal{V}$ with $r_v(c) \geq r_w(c)$, it holds that $|\succ_w| - r_w(c) + r_v(c) \leq k + 1$.*

Example 1. *We examine each pair of occurrences of an alternative in pairs of voters, and derive a lower bound for k. Consider the following example, with voters v and w:*

$$v\colon a \succ b \succ c \succ d \succ e, \text{ and } w\colon f \succ d \succ g \succ h \succ i.$$

For alternative d, we have $r_w(d) = 1$ and $r_v(d) = 3$. Lemma 2 implies that $k \geq |\succ_w| - 1 + 3 - 1 = 6$. Intuitively, alternative d needs to have *at least three attributes in order to explain voter w (since it is ranked higher than three other alternatives by w), but there should also be at least three attributes that d does not* have *in order to explain voter v (since three other alternatives are ranked higher than d).*

Finally, we show that a k-BAM always exists for large enough k.

Lemma 3 ($\star$). *For a profile with m alternatives and n voters, a k-BAM with $k \geq (m-1) \cdot m$ or $k \geq (m-1) \cdot n$ always exists.*

Remark 2. *Lemma 3 immediately implies fixed-parameter tractability for k-BAM wrt. m since a profile with m alternatives has at most $O(2^m \cdot m!)$ different (incomplete) preference orders.*

Moreover, for profiles with complete preferences, we have by Lemma 1(i) *and Lemma 3 that $m - 1 \leq k \leq m(m-1)$. Therefore, the parameters k and m are equivalent from the parameterized perspective. In other words, a voting problem is FPT wrt. m if and only if it is FPT wrt. k.*

3 BAM

In this section, we consider the first computational problem BAM. We first establish NP-completeness and then propose some parameterized and polynomial-time algorithms for some special cases.

General complexity. BAM is NP-hard by a reduction from the NP-complete problem below (Papadimitriou 1994).

3-COLORING
Input: An undirected graph $G = (U, E)$.
Question: Does G admit a *proper* 3*-coloring*, i.e., a function $\chi\colon U \to [3]$ s.t. no two adjacent vertices have the same value?

Theorem 1 ($\star$). BAM *is NP-complete; it remains NP-hard even if $k = 3$ and every preference order has length two.*

Proof. NP-membership follows directly from Observation 1. To show NP-hardness, we reduce from 3-COLORING, which results in an instance of BAM with $k = 3$ such that every preference order has length two.

Let $G = (U, E)$ be an instance of 3-COLORING. W.l.o.g, assume that no vertices have degree zero. We create an instance $I' = (\mathcal{P} = (\mathcal{C}, \mathcal{V}, \mathcal{R}), k = 3)$ as follows.

Alternatives. For each vertex $u \in U$, we add an alternative c_u to $\mathcal{C}$. Additionally, we add to $\mathcal{C}$ three groups of dummy alternatives containing in total seven dummies $D_1 := \{d_1^1, d_1^2, d_1^3\}$, $D_2 := \{d_2^1, d_2^2, d_2^3\}$, and $D_3 := \{d_3\}$. We will ensure that the alternatives in D_1 will each have a distinct attribute, the alternatives in D_2 will each have a distinct pair of attributes, while the single alternative in D_3 will have all three attributes.

Voters and their preferences.
– For each vertex $u \in U$, we add three voters v_u^1, v_u^2, v_u^3 with preference orders: $v_u^1\colon d_2^1 \succ c_u$, $v_u^2\colon d_2^2 \succ c_u$, and $v_u^3\colon d_2^3 \succ c_u$ (ensuring c_u has at most one attribute).
– For each edge $\{u, w\} \in E$, we add two voters $v_{u,w}$ and $v_{w,u}$ with preference orders $v_{(u,w)}\colon c_u \succ c_w$ and $v_{(w,u)}\colon c_w \succ c_u$. This will ensure simultaneously that c_u, c_w have exactly one attribute and they are distinct.
– We add 24 unnamed dummy voters in four groups V_1, V_2, V_3, V_4. These voters ensure that the dummy alternatives have the desired number of attributes as mentioned before. The first group V_1 has six voters with preference orders: $d_1^1 \succ d_1^2$, $d_1^2 \succ d_1^1$, $d_1^1 \succ d_1^3$, $d_1^3 \succ d_1^1$, $d_1^2 \succ d_1^3$, $d_1^3 \succ d_1^2$.
The second group V_2 has 9 voters with preference orders: $d_2^1 \succ d_1^1$, $d_2^1 \succ d_1^2$, $d_2^1 \succ d_1^3$, $d_2^2 \succ d_1^1$, $d_2^2 \succ d_1^2$, $d_2^2 \succ d_1^3$, $d_2^3 \succ d_1^1$, $d_2^3 \succ d_1^2$, $d_2^3 \succ d_1^3$.
The third group V_3 has 6 voters with preference orders: $d_2^1 \succ d_2^2$, $d_2^2 \succ d_2^1$, $d_2^1 \succ d_2^3$, $d_2^3 \succ d_2^1$, $d_2^2 \succ d_2^3$, $d_2^3 \succ d_2^2$.
The last group V_4 has 3 voters with preference orders: $d_3 \succ d_2^1$, $d_3 \succ d_2^2$, $d_3 \succ d_2^3$,

This concludes the construction, which can clearly be done in polynomial time. It remains to show the correctness, i.e., G has a proper 3-coloring if and only if the constructed profile $\mathcal{P}$ admits a 3-BAM. For the "only if" part, let $\chi\colon U \to [3]$ be a proper 3-coloring. It is straightforward to check that the following functions $\mathsf{has}\colon \mathcal{C} \to [3]$ and $\mathsf{cares}\colon \mathcal{V} \to [3]$ explain our profile.
– For each vertex $u \in U$, let $\mathsf{has}(c_u) := \{\chi(u)\}$.
– For each $j \in [3]$, let $\mathsf{has}(d_1^j) := \{j\}$ and $\mathsf{has}(d_2^j) := [3] \setminus \{j\}$. Let $\mathsf{has}(d_3) := [3]$.
– For each dummy voter $v \in \cup_{i \in [4]} V_i$, let x_v be the first-ranked alternative of the voter and $\mathsf{cares}(v) := \mathsf{has}(x_v)$.
– For each vertex $u \in U$, let $\mathsf{cares}(v_u^1) = \mathsf{cares}(v_u^2) = \mathsf{cares}(v_u^3) := [3]$.

Algorithm 1: BruteForceM

Input: A profile $\mathcal{P} = (\mathcal{C}, \mathcal{V}, \mathcal{R})$ with $m = |\mathcal{C}|$, $k \in \mathbb{N}$
Output: *yes* if $\mathcal{P}$ admits a k-BAM else *no*

1 **if** $k \geq m(m-1)$ **then return** *yes*;
2 **foreach** has $\subseteq {2^{[k]}}^m$ **do**
3   **foreach** $(v, S) \in \mathcal{V} \times 2^{[k]}$ **do**
4   **if** (has, S) explains v **then** cares$(v) \leftarrow S$;
5   **if** (has, cares) *is BAM for* $\mathcal{P}$ **then return** *yes*;
6 **return** *no*

– For each edge $\{u, w\} \in E$, let cares$(v_{(u,w)}) := \{\chi(u)\}$ and cares$(v_{(w,u)}) := \{\chi(w)\}$.

The details are deferred to the full version of the paper (Anzinger et al. 2025).

For the "if" part, let $\mathcal{M} = (\text{has}, \text{cares})$ be a 3-BAM for $\mathcal{P}$. We can show that the has function restricted to the "vertex" alternatives yields a proper 3-coloring by observing that no two dummy alternatives have the same subset of attributes and hence every vertex has a single distinct attribute. The proof is also deferred to the full version of the paper (Anzinger et al. 2025). □

Tractability results. We first show that determining a k-BAM can be done efficiently if all preference lengths ℓ are maximum possible, i.e., $\ell = k + 1$. This complements Theorem 1 where $\ell = k - 1$. The case with $\ell = k$ remains open.

Proposition 1 ($\star$). *If all preference orders in a preference profile are of length $k + 1$, then* BAM *can be solved in time $|\mathcal{P}|^{O(1)}$, i.e., polynomial time.*

The following result shows a dichotomy for the complexity of BAM wrt. the number of attributes. Theorem 1 shows that BAM is NP-hard if there are at least three attributes, while the next theorem shows that for two or less attributes the problem becomes tractable.

Theorem 2 ($\star$). *If $k \leq 2$,* BAM *is solvable in $O(n)$ time, i.e., linear time.*

Next, we consider the number m of alternatives as parameter. By Lemma 3 and due to the fact that there are at most $2^m \cdot m!$ many different preference orders, we immediately obtain a problem kernel wrt. m (that is, the instance size is bounded by $f(m)$), which yields FPT result for m by brute-force searching. Below, we provide an improved approach by only guessing the attribute set for each alternative.

Theorem 3. BAM *is solvable in time $2^{O(m^3)} \cdot |\mathcal{P}|^{O(1)}$, which is FPT wrt. m.*

Proof. The idea is to branch into all possible combinations of attribute subsets for the alternatives and for each branch check whether every voter can be explained by choosing an appropriate subset of attributes that he should care about. A pseudo-code can be found in Algorithm 1.

If $k \geq m(m-1)$, the algorithm is correct due to Lemma 3. Otherwise, since line 2 brute-force searches for all combinations has of the attribute subsets for the attributes, we will not miss a correct has function if the instance is a yes-instance. In Lines 3–6 we inspect for each voter whether there exists a subset of attribute that together with the branched has function can explain the voter's preference order. This is correct by the nature of brute forcing and by the fact that the cares functions of different preference orders do not affect each other.

It remains to check the running time: Line 1 runs in constant time. After that it is a triply nested for-loop with at most $(2^k)^m \cdot n \cdot 2^k \leq n \cdot 2^{m^3}$ iterations and a $|\mathcal{P}|^{O(1)}$ body. □

Now, we consider the combined parameter (n, k). By Lemma 1(ii), we infer that the sum of the lengths of the preference orders in a profile is at most $n \cdot (k+1)$. Since we can ignore alternatives that do not appear in any preference order, we can assume w.l.o.g. that $m \leq n \cdot (k+1)$. Together with Theorem 3, it follows:

Corollary 1. BAM *is solvable in time $2^{O((n \cdot (k+1))^3)} \cdot |\mathcal{P}|^{O(1)}$, which is FPT wrt. $n + k$.*

As for the single parameter n, our attempts at showing $W[1]$-hardness have been unsuccessful so far. We weakly conjecture that BAM is FPT wrt. n and provide a starting point for investigating the problem.

Theorem 4 ($\star$). *For $n = 2$ voters, one can determine the minimum number k of attributes of a BAM in $O(m)$ time, i.e., linear time, and compute a corresponding k-BAM in $O(m^2)$ time.*

Proof. Observe that with 2 voters, there are 3 different types of attributes, two being cared about by a single voter, and one by both. To determine the minimum k, we first compute a BAM where for each voter v and each alternative c, the score$_v(c)$ is tight, i.e., score$_v(c) = |\succ_v| - r_v(c) - 1$. We can compute in polynomial time the minimum number of attributes cared about by a single voter, and then the minimum number of attributes cared about by both in such a BAM. This directly yields the cares function for each voter. Afterwards, we compute the has function and show that the computed BAM is an optimal one.

Proof outline. We first define *attribute types* which play a central role. Afterwards we compute values that keep track of how many attributes of each type each alternative has. We show correctness of our procedure by showing that these values correspond to a BAM that minimizes the number of attributes.

The correctness proof is split into two parts. We first show that the computed values represent a BAM with some number k of attributes that explains the given profile $\mathcal{P}$. Then, we show that every BAM requires at least k attributes.

The procedure. Let $\mathcal{P} = (\mathcal{C}, \{u, w\}, \mathcal{R})$.

Attribute types and necessary definitions. We first define the (attribute) types for a given BAM $\mathcal{M} = (\mathsf{AT}, \text{has}, \text{cares})$. The *type* of an attribute α is *type*$(\alpha) := \{v \mid \alpha \in \text{cares}(v)\}$. W.l.o.g., we assume that each attribute is cared for by at least one voter in $\mathcal{M}$. For two voters there are hence three types; namely, $\{u\}$, $\{w\}$, and $\{u, w\}$. Define the set of attributes of each type as AT_u, AT_w, and $\mathsf{AT}_{u,w}$, respectively. Formally, $\mathsf{AT} = \mathsf{AT}_u \uplus \mathsf{AT}_w \uplus \mathsf{AT}_{u,w}$, where $\mathsf{AT}_u :=$

$\mathsf{cares}(u) \setminus \mathsf{cares}(w)$, $\mathsf{AT}_w := \mathsf{cares}(w) \setminus \mathsf{cares}(u)$, and $\mathsf{AT}_{u,w} := \mathsf{cares}(u) \cap \mathsf{cares}(w)$.

Similarly, for each alternative c we partition the attributes it has into $\mathsf{has}_{u,w}$, has_u, and has_w. Formally, $\mathsf{has}(c) = \mathsf{has}_u(c) \uplus \mathsf{has}_w(c) \uplus \mathsf{has}_{u,w}(c)$, where

$$\mathsf{has}_u(c) := \mathsf{has}(c) \cap \mathsf{AT}_u, \quad \mathsf{has}_w(c) := \mathsf{has}(c) \cap \mathsf{AT}_w,$$
$$\mathsf{has}_{u,w}(c) := \mathsf{has}(c) \cap \mathsf{AT}_{u,w}.$$

Finally, for every alternative c we define some scores: $S_u(c) := |\mathsf{has}_u(c)|$, $S_w(c) := |\mathsf{has}_w(c)|$, $S_{u,w}(c) := |\mathsf{has}_{u,w}(c)|$. Clearly, $\text{score}_u(c) = S_{u,w}(c) + S_u(c)$ and $\text{score}_w(c) = S_{u,w}(c) + S_w(c)$.

Computing the necessary numbers of each attribute type. The formal computation steps are provided in Algorithm 2. We now additionally give an informal description and intuition. To compute a BAM, the main task is to compute the values $S_{u,w}(c)$, $S_u(c)$, and $S_w(c)$ for each alternative $c \in \mathcal{C}$ given above.

Intuitively, these values are lower bounds on the number of the corresponding attribute types that c requires in order to attain its ranks in u and w's preference orders. Using these values, we compute a (not necessarily unique) BAM, by determining $\max_{c\in\mathcal{C}} S_u(c)$ attributes of type AT_u, $\max_{c\in\mathcal{C}} S_w(c)$ attributes of type AT_w, and $\max_{c\in C} S_{u,w}(c)$ attributes of type $\mathsf{AT}_{u,w}$. Finally, for each alternative c and each type AT_T, we then arbitrarily assign $S_T(c)$ many attributes of type AT_T to alternative c.

Recall that for alternative c and voter v, $\text{score}_v(c)$ corresponds to the number of alternatives that is ranked lower than c by v, e.g., the last-ranked alternative has score zero. These values are computed in the for-loop in line 2.

The computation then proceeds as follows: In the for-loop in line 4, we first compute intermediary values, $t_{u,w}$, t_u, and t_w for alternatives c that are ranked by both voters. These values shall represent the least amount of attributes needed to ensure that c attains its ranks. In other words, we use as many attributes of type $\{u, w\}$ as possible, and fill the gap to the corresponding ranks with attributes of type $\{u\}$ and $\{w\}$, respectively. It is intuitive that these "filler" attributes are necessary since their role cannot be taken over by attributes of other types. Since the attribute types are used across the alternatives, in line 6 we compute the maximum over all filler attributes that only u resp. only v cares about, i.e., $M_u := \max_{c\in C} t_u(c)$ and $M_w := \max_{c\in C} t_w(c)$.

We remark that the $t_{u,w}$-value is not necessarily a lower bound on the number of attributes of type $\{u, w\}$ since we may reserve more filler attributes of type $\{u\}$ (resp. $\{w\}$) in line 6 and could "reuse" them. Intuitively, if we set the number of attributes of type $\{u\}$ and $\{w\}$ to M_u and M_w, respectively, they are already minimum possible. Thus it now remains to minimize the number of attributes of type $\{u, w\}$. The idea in the for-loop in line 7, is thus to determine for each relevant alternative c the number conv_c of attributes that can be reused. Then we reduce $t_{u,w}(c)$ accordingly to obtain our desired $S_{u,w}(c)$ value, which shall be the minimum number of attributes of type $\{u, w\}$ needed for c.

Finally, in the for-loops in lines 9 and 11, the alternatives ranked by only one voter will receive their values. Again, we do so by giving them exactly as many attributes as needed for their position. For alternatives $c \in \mathcal{C}$ ranked only by voter $v \in \{u, w\}$, we set $S_v(c) := \min\{M_v, |\succ_v| - r_v(c) - 1\}$ and $S_{u,w}(c) = r_v(c) - 1 - S_v(c)$. Informally, we assign already existing attributes from AT_v if possible, and then use attributes from $\mathsf{AT}_{u,w}$ for the rest and, if necessary, create new attributes in the set $\mathsf{AT}_{u,w}$. This concludes the informal description. Note that k is the total number of attributes used by a BAM corresponding to our computed values. Now that we have described the computational steps, it remains to show the correctness of the computed values. In other words, it remains to show that a BAM using a minimum number of attributes can be generated from the computed values.

Algorithm 2: `Value-Computation`

Input: A profile $\mathcal{P} = (\mathcal{C}, \{u, w\}, \mathcal{R})$
Output: Minimum k, and number of attributes M_T and score values $S_T(c)$ for all $\emptyset \subset T \subseteq \{u, w\}$ and all $c \in \mathcal{C}$.

1 Initialize all used variables as 0;
2 **foreach** $(c, v) \in \mathcal{C} \times \mathcal{V}$ **do**
3 $\quad \lambda_v(c) := \begin{cases} |\succ_v| - r_v(c) - 1, & \text{if } v \text{ ranks } c \\ 0, & \text{otherwise.} \end{cases}$
4 **foreach** $c \in C$ *that is ranked by both voters* **do**
5 $\quad t_{u,w}(c) := \min(\lambda_u(c), \lambda_w(c))$, $t_u(c) := \lambda_u(c) - t_{u,w}(c)$, $t_w(c) := \lambda_w(c) - t_{u,w}(c)$
6 $M_u := \max_{c\in\mathcal{C}} t_u(c)$ and $M_w := \max_{c\in\mathcal{C}} t_w(c)$
7 **foreach** $c \in C$ *that is ranked by both voters* **do**
8 $\quad \text{conv}_c := \min(M_w - t_w(c), M_u - t_u(c), t_{u,w}(c))$, $S_{u,w}(c) := t_{u,w}(c) - \text{conv}_c$, $S_u(c) := t_u(c) + \text{conv}_c$, and $S_w(c) := t_w(c) + \text{conv}_c$
9 **foreach** $c \in C$ *only ranked by* u **do**
10 $\quad S_w(c) := 0$, $S_u(c) := \min(\lambda_u(c), M_u)$, $S_{u,w}(c) := \lambda_u(c) - S_u(c)$
11 **foreach** $c \in C$ *only ranked by* w **do**
12 $\quad S_u(c) := 0$, $S_w(c) := \min(\lambda_w(c), M_w)$, $S_{u,w}(c) := \lambda_w(c) - S_w(c)$
13 $M_{u,w} := \max_{c\in\mathcal{C}} S_{u,w}(c)$; $k := M_{u,w} + M_u + M_w$;
14 **return** *k, M_T and $S_T(c)$ for all $c \in \mathcal{C}$ and $\emptyset \subset T \subseteq \{u, w\}$*

We defer the complete proof of correctness to the full version of the paper (Anzinger et al. 2025), but give a brief overview of the correctness proof. It is straightforward to verify that using the values a k-BAM can be found. One can simply generate M_T attributes of type $\emptyset \subset T \subseteq \{u, w\}$ and assign $S_T(c)$ many arbitrary attributes of type T to alternative c. To show that the computed value for k is indeed minimum, we show the existence of two types of alternatives:

- An alternative c that must have all attributes in $\mathsf{AT}_{u,w}$ and all attributes in either AT_u or AT_w.
- An alternative b that must have all attributes in AT_u (resp. AT_w) and none of the attributes in AT_w (resp. AT_u).

We then show that in order to explain the rankings of these alternatives for the voters, we require at least k attributes. We show this by lower-bounding the score of alternative/voter-pairs and lower-bounding the number of attributes that are in $\mathsf{has}(b) \setminus \mathsf{has}(c)$ and $\mathsf{has}(c) \setminus \mathsf{has}(b)$, respectively. □

4 BAM with Cares

In this section we study the restriction on the BAM problem, where the cares-function is given, and the question is whether a has-function exists that completes a valid k-BAM. Unfortunately, for all single parameters, except for n, this problem remains NP-hard even for constant parameter value. We complement this by providing FPT-algorithms for all 2-parameter combinations.

General complexity. The next two theorems show that BAM WITH CARES is NP-complete. In both cases we reduce from the well-known NP-complete 3-SAT problem.

Theorem 5 ($\star$)**.** BAM WITH CARES *is NP-complete.* BAM WITH CARES *remains NP-hard even if* $m = 3$.

Using a similar reduction, we show that even for a constant number of attributes the problem remains NP-hard.

Theorem 6 ($\star$)**.** BAM WITH CARES *remains NP-hard even if* $k = 6$.

Tractability results. We now consider the two parameter combinations. For (n, m), we obtain FPT result via integer linear programming (ILP).

Theorem 7. BAM WITH CARES *can be solved in time* $(2^n \cdot m)^{O(2^n \cdot m)} \cdot |\mathcal{P}|^{O(1)}$*, which is FPT wrt.* $n + m$ *.*

Proof. We solve this problem using an ILP with $O(2^n \cdot m)$ variables and $2^n \cdot m + n \cdot m$ constraints. It is known that ILPs can be solved in FPT time wrt. the number of variables (Frank and Tardos 1987). The intuition behind the ILP is to group the attributes according to types. Similarly to Theorem 4, an attribute's type is defined by the subset of voters that care about it. For each attribute type and each alternative we create a variable that stores how many attributes of that type the alternative has. We ensure the preference profile is explained by adding constraints for each voter v and each pair of alternatives that are consecutive in v's preference order. We now describe the variables and constraints in the ILP:

- For every $T \subseteq \mathcal{V}$ and every $a \in \mathcal{C}$, we add a variable $x_{T,a}$. We compute the number of attributes of type T, $m_T = |\{\alpha \in \mathsf{AT} \mid \forall v \in T\colon \alpha \in \mathsf{cares}(v) \wedge \forall v \in \mathcal{V} \setminus T\colon \alpha \notin \mathsf{cares}(v)\}|$ and add the constraint, $x_{T,a} \leq m_T$.
- For every voter $v \in \mathcal{V}$, we add constraints in the following way. Let $v\colon a_1 \succ \cdots \succ a_{m'}$ be the preferences of the voter. We add the following constraint for each $i \in [m' - 1]\colon \sum_{v \in T} x_{T,a_i} \geq 1 + \sum_{v \in T} x_{T,a_{i+1}}$.

Correctness can be checked straightforwardly: The value of each variable corresponds exactly to the number of attributes of that type that alternative a has. If there exists a k-BAM with the given cares, then setting the variables according to the BAM satisfies the constraints. On the other hand, if the ILP has a feasible solution $x_{T,a}$, then we can add $x_{T,a}$ attributes for each type T to alternative a. As the number of variables of each type does not exceed the existing attributes of that type and the constraints added for each $i \in [m' - 1]$ must be satisfied, this leads to a valid k-BAM containing cares. Since ILP feasibility can be checked in $O(p^{2.5p+o(p)} \cdot L)$ time, where L is the size of the ILP and p the number of variables (Frank and Tardos 1987), it follow that our running time is $O((2^n \cdot m)^{2.5 \cdot (2^n \cdot m) + o(2^n \cdot m)}) \cdot |\mathcal{P}|^{O(1)} = (2^n \cdot m)^{O(2^n \cdot m)} \cdot |\mathcal{P}|^{O(1)}$. □

Using Lemma 1(ii), we can then derive the following result from Theorem 7.

Corollary 2 ($\star$)**.** BAM WITH CARES *is solvable in* $(2^n \cdot n \cdot (k+1))^{O(2^n \cdot n \cdot (k+1))} \cdot |\mathcal{P}|^{O(1)}$ *time, i.e., FPT wrt.* $n + k$*.*

Finally, we can get an FPT algorithm for the parameter $m + k$ by brute-forcing through all possible has functions.

Theorem 8 ($\star$)**.** BAM WITH CARES *can be solved in time* $(2^k)^m \cdot |\mathcal{P}|^{O(1)}$*, which is FPT wrt.* $m + k$*.*

5 BAM with Has

We now study the restriction on the BAM problem, where the has-function is given, and the question is whether a cares-function exists that yields a valid k-BAM. We show that the problem is already hard even if there is only one voter, but is FPT wrt. the other parameters.

General complexity. We show hardness by giving a reduction from the NP-complete RESTRICTED EXACT 3-SET COVER problem (Gonzalez 1985).

Theorem 9 ($\star$)**.** BAM WITH HAS *is NP-complete.* BAM WITH HAS *remains NP-hard even if* $n = 1$.

Tractability results. For the parameter m, we can run a separate ILP using $O(2^m)$ many variables to compute the $\mathsf{cares}(v)$ for each voter v since the cares-functions do not affect each other. This immediately yields FPT result for m.

Theorem 10 ($\star$)**.** BAM WITH HAS *can be solved in time* $(2^m)^{O(2^m)} \cdot |\mathcal{P}|^{O(1)}$*, which is FPT wrt.* m*.*

Finally, we can branch over all 2^k possibilities of assigning attributes for each of the voters, giving the following result.

Theorem 11 ($\star$)**.** BAM WITH HAS *can be solved in time* $2^k \cdot |\mathcal{P}|^{O(1)}$*, which is FPT wrt.* k*.*

6 Conclusion and Open Questions

On the positive side, we found tractability results if the number m of alternatives is small or if the number n of voters and the number k of attributes are small. On the negative side, we had to leave the tractability status with respect to n open and the practically interesting parameterization by k turned out to be intractable. Since the latter parameterization is related to graph coloring, it seems most promising to consider additional, structural parameters of the instance or solution to obtain more islands of tractability. For instance, based on a BAM we can define a tri-partite graph consisting of the voters, alternatives, and attributes, representing which voters rank which alternatives, which voters care about which attributes and which alternatives have which attributes. Perhaps if we restrict the structure of this graph, the problem becomes more tractable?

Acknowledgments

The authors are supported by the Vienna Science and Technology Fund (WWTF) [10.47379/ VRG18012]. We would like to thank the reviewers for their helpful comments.

References

Anzinger, C.; Chen, J.; Hatschka, C.; Sorge, M.; and Temper, A. 2025. How Hard is it to Explain Preferences Using Few Boolean Attributes? Technical report, arXiv.

Auriau, V.; Belahcène, K.; Malherbe, E.; and Mousseau, V. 2024. Learning Multiple Multicriteria Additive Models from Heterogeneous Preferences. In *Proceedings of the 8th International Conference Algorithmic Decision Theory (ADT 2024)*, 207–224.

Barberá, S.; Gul, F.; and Stacchetti, E. 1993. Generalized Median Voter Schemes and Committees. *Journal of Economic Theory*, 61(2): 262–289.

Benabbou, N.; Diodoro, S. D. S. D.; Perny, P.; and Viappiani, P. 2016. Incremental Preference Elicitation in Multi-attribute Domains for Choice and Ranking with the Borda Count. In *Proceedings of the Scalable Uncertainty Management - 10th International Conference, SUM 2016*, 81–95.

Bhatnagar, N.; Greenberg, S.; and Randall, D. 2008. Sampling stable marriages: why spouse-swapping won't work. In *Proceedings of the 19th annual ACM-SIAM symposium on Discrete algorithms, SODA 2008*, 1223–1232.

Chen, J.; Nöllenburg, M.; Simola, S.; Villedieu, A.; and Wallinger, M. 2022. Multidimensional Manhattan Preferences. In *Proceedings of the 14th Latin American Symposium (LATIN 2022)*, 273–289.

Chen, J.; Pruhs, K. R.; and Woeginger, G. J. 2017. The One-Dimensional Euclidean Domain: Finitely Many Obstructions Are Not Enough. *Social Choice and Welfare*, 48(2): 409–432.

Cheng, C. T.; and Rosenbaum, W. 2023. Stable Matchings with Restricted Preferences: Structure and Complexity. *ACM Transactions on Economics and Computation*, 10(3): 13:1–13:45.

Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer.

Elkind, E.; Lackner, M.; and Peters, D. 2022. Preference Restrictions in Computational Social Choice: A Survey. *CoRR*, abs/2205.09092.

Feffer, M.; Skirpan, M.; Lipton, Z. C.; and Heidari, H. 2023. From Preference Elicitation to Participatory ML: A Critical Survey & Guidelines for Future Research. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2023*, 38–48.

Fisman, R.; Iyengar, S. S.; Kamenica, E.; and Simonson, I. 2006. Gender Differences in Mate Selection: Evidence From a Speed Dating Experiment. *The Quarterly Journal of Economics*, 121(2): 673–697.

Frank, A.; and Tardos, É. 1987. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1): 49–65.

Fürnkranz, J.; and Hüllermeier, E., eds. 2010. *Preference Learning*. Springer.

Fürnkranz, J.; and Hüllermeier, E. 2017. Preference Learning. In *Encyclopedia of Machine Learning and Data Mining*, 1000–1005. Springer.

Gonzalez, T. F. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science*, 38: 293–306.

Holzinger, A.; Saranti, A.; Molnar, C.; Biecek, P.; and Samek, W. 2020. Explainable AI Methods - A Brief Overview. In *Proceedings of the Workshop xxAI - Beyond Explainable AI Held in Conjunction with the International Conference on Machine Learning (xxAI@ICML 20)*, volume 13200 of *Lecture Notes in Computer Science*, 13–38. Springer.

Inada, K. 1964. A note on the simple majority decision rule. *Econometrica*, 32: 525–531.

Inada, K. 1969. The simple majority rule. *Econometrica*, 37: 490–506.

Künnemann, M.; Moeller, D.; Paturi, R.; and Schneider, S. 2019. Subquadratic Algorithms for Succinct Stable Matching. *Algorithmica*, 81(7): 2991–3024.

Labreuche, C. 2011. A general framework for explaining the results of a multi-attribute preference model. *Artificial Intelligence*, 175(7-8): 1410–1448.

Lang, J.; and Xia, L. 2009. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57(3): 304–324.

Lang, J.; and Xia, L. 2016. Voting in Combinatorial Domains. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*, 197–222. Cambridge University Press.

McFadden, D. 1974. Conditional logit analysis of qualitative choice behavior. Technical Report 105-142, Frontiers in Econometric, New York.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison-Wesley.

Peters, D. 2017. Recognising Multidimensional Euclidean Preferences. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence,*, 642–648.

Przedmojski, T. 2016. *Algorithms and Experiments for (Nearly) Restricted Domains in Elections*. Master's thesis, TU Berlin.

Siskos, Y.; Grigoroudis, E.; and Matsatsinis, N. F. 2016. UTA Methods. In Greco, S.; Ehrgott, M.; and Figueira, J. R., eds., *Multiple Criteria Decision Analysis: State of the Art Surveys*, 315–362. Springer New York.

Sui, X.; Francois-Nienaber, A.; and Boutilier, C. 2013. Multi-Dimensional Single-Peaked Consistency and Its Approximations. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, 375–382.

Tversky, A. 1972. Elimination by aspects: A theory of choice. *Psychological Review*, 79(4): 281–299.

Yu, P.-L. 1985. *Multiple-Criteria Decision Making*. Springer US.