On the Tree Representations of Dichotomous Preferences

Yongjie Yang
Chair of Economic Theory, Saarland University, Saarbrücken, Germany
yyongjiec@gmail.com

Abstract

We study numerous restricted domains of dichotomous preferences with respect to some tree structures. Particularly, we study the relationships among these domains and the ones proposed by Elkind and Lackner [2015]. We also show that recognizing all the restricted domains proposed in this paper is polynomial-time solvable. Finally, we explore the complexity of winner determination for several important approval-based multiwinner voting rules when restricted to these domains.

1 Introduction

Preference domain restrictions have been widely studied in computational social choice recently. The main motivations of the study can be summarized as follows. First, domain restrictions have been a successful approach to circumventing many impossibility theorems. One of the classic examples is arguably the single-peaked domain, restricted to which the median voter rule is non-dictatorial yet strategy-proof [Black, 1948; Moulin, 1980], circumventing Gibbard and Satterthwaite’s impossibility theorem [Gibbard, 1973; Satterthwaite, 1975]. Second, many voting problems which are computationally hard in general become polynomial-time solvable when restricted to special domains [Brandt et al., 2015; Faliszewski et al., 2011]. Domain restrictions also offer many structural parameters for researchers to study voting problems from the parameterized complexity point of view. Many fixed-parameter tractability results have been established with respect to several domain restrictions [Bredereck et al., 2016; Cornaz et al., 2012; Yang, 2015; Yang and Guo, 2014]. Third, domain restrictions model many real-world applications, because, in many practical scenarios, the preferences of voters are subject to some combinatorial restrictions, resulting in a restricted domain of preferences. We refer to [Elkind et al., 2017] for a comprehensive discussion on domain restrictions.

Following up the work of Elkind and Lackner [2015], we study several domain restrictions on dichotomous preferences. In general, in our models, either candidates or votes are mapped into vertices in a (rooted, oriented) tree and, more importantly, for the former case all approved candidates of every vote induce a special structure, and in the latter case all votes approving a common candidate induce a specific structure. Domain restrictions based on tree structures have been studied previously. For instance, Demange [1983] extended the single-peaked domain proposed by Black [1948] to single-peaked preferences on a tree. Peters and Elkind [2016] studied single-peaked preferences on trees with further restrictions. Recently, a more general graph class, namely median graphs, has been used in the study of domain restrictions [Clearwater et al., 2015; Demange, 2012; Puppe and Slinko, 2019; Kung, 2015]. One common point of these models is that the trees or graphs involved are all undirected. These domains may not be able to model the scenarios where candidates or votes have domination or dependency relations. In this paper, we consider domain restrictions based on undirected trees, as well as rooted and oriented trees where each edge has a direction which may indicate the domination/dependency relations between the two vertices incident to the edge. We study inclusive relationships among domain restrictions defined in this paper and the ones studied in [Elkind and Lackner, 2015] (Section 3). In addition, we explore whether they can be recognized efficiently (Section 4). Finally, we study the complexity of WINNER DETERMINATION for many important approval-based multiwinner voting rules (Section 5).

2 Preliminary

We assume the familiarity of basics in graph theory and complexity theory. A graph \(G = (N, A)\) consists of a set of vertices \(N\) and a set of edges \(A\). A bipartite graph is a graph whose vertices can be partitioned into two subsets such that there are only edges between these two subsets. A path is a sequence \(u_1, \ldots, u_t\) of vertices such that there is an edge between \(u_i\) and \(u_{i+1}\) for each \(1 \leq i < t\). A comb is obtained from a path by, for every vertex \(u\) in the path, introducing one degree-1 vertex adjacent to \(u\). A tree is a connected graph without cycles. A star is a tree where there is a specific vertex, called the center, and every other vertex is of degree-1 and is adjacent to the center. A rooted tree is a tree with a specific vertex, called the root. An oriented tree is a tree where each edge has a direction. An arborescence is an oriented tree where there is exactly one vertex without incoming edges.

We study approval-based multiwinner voting. In this setting, an election is a tuple \(E = (C, V)\) where \(C\) is a set of candidates and \(V\) a multiset of votes. Each vote \(v \in V\) is a...
subset of $C$, representing the set of candidates approved by the corresponding voter. For each candidate $c \in C$, let $V(c)$ be the multiset of votes approving $c$, i.e., $V(c) = \{v \in V : c \in v\}$. A $k$-committee is a subset of $k$ candidates. A $k$-committee selection (multiwinner voting) rule maps each election $(C, V)$ to a $k$-committee, called the winners.

An election $E = (C, V)$ can be also represented by a bipartite graph with the vertex bipartition $(C, V)$. Moreover, there is an edge between $c \in C$ and $v \in V$ if and only if $v$ approves $c$, i.e., $c \in v$. We denote this graph by $G_E$ and call it the incidence graph of $(C, V)$.

Now we are ready to give the formal definitions of several restricted domains for dichotomous preferences. Generally speaking, in each of these domains candidates are mapped into vertices of a tree and approved candidates of each vote induce some specific structure. We refer to Figures 1–3 for illustrations of these concepts. Let $E = (C, V)$ be an election.

**α-Tree representation (α-TR)** We say that $E$ admits an α-TR if there is a rooted tree with vertex set $C \cup \{x\}$ where $x \not\in C$ is the root and, moreover, for every vote $v \in V$ there is a candidate $c \in C$ such that $v$ approves exactly the candidates in the path from $x$ to $c$, except $x$.

**β-Tree representation (β-TR)** We say that $E$ admits a β-TR if there is an oriented tree $T = (C, A)$ such that the approved candidates of each vote $v \in V$ induce an arborescence.

**β-Path-tree representation (β-PTR)** We say that $E$ admits a β-PTR if there exists an oriented tree $T = (C, A)$ such that the approved candidates of every vote $v \in V$ induce a directed path in $T$.

**Tree representation (TR)** We say that $E$ admits a TR if there exists a tree $T = (C, A)$ such that the approved candidates of every vote induce a subtree of $T$.

**Path-tree representation (PTR)** We say that $E$ admits a PTR if there exists a rooted tree with vertex set $C \cup \{x\}$ where $x \not\in C$ is the root and, moreover, for every vote $v \in V$ there exists an integer $i$, except $x$.

 PTR if there exists a tree $T = (C, A)$ such that the approved candidates of every vote induce a path in $T$.

Analogous to the above definitions, we can define tree representations of an election where votes are mapped into vertices of a tree and require that for each candidate $c$ the votes approving $c$ induce some specific subgraph. For each of the above concepts we add the letter "V" immediately after the hypen to denote such a tree representation. For instance, for β-TR, we say an election $(C, V)$ admits a β-VTR representation, if there is an oriented tree with vertex set $V$ such that for every candidate $c \in C$, votes in $V(c)$ induce an arborescence.

As we also study some domains proposed in [Elkind and Lackner, 2015], let’s also recall the definitions of these domains. For an integer $i$, let $[i] = \{1, 2, \ldots, i\}$.

**t-Partition (t-PART)** We say that $E$ is t-PART if there is a partition $(C_1, C_2, \ldots, C_t)$ of $C$ such that for every vote $v$ it holds that $v = C_i$ for some $i \in [t]$.

**Voter extremal interval (VEI)** We say that $E$ is VEI if there is an order $(v_1, v_2, \ldots, v_n)$ of $V$ such that for every candidate $c \in C$, there exists an integer $i \in [n]$ such that $V(c)$ is either $\{v_1, v_2, \ldots, v_i\}$ or $\{v_{i+1}, \ldots, v_n\}$.

**Voter interval (VI)** We say that $E$ is VI if there is a linear order over $V$ so that for every candidate $c \in C$, the votes approving $c$ are consecutive in this order.

**Candidate extremal interval (CEI)** We say that $E$ is CEI if there is an order $(c_1, c_2, \ldots, c_m)$ of $C$ such that for every vote $v \in V$ there exists an integer $i \in [m]$ such that either $v = \{c_j : 1 \leq j \leq i\}$ or $v = \{c_j : m \geq j \geq i\}$.

**Candidate interval (CI)** We say that $E$ is CI if there is an order of $C$ such that the approved candidates of every vote are consecutive in this order.

**Dichotomous uniformly Euclidean (DUE)** We say that $E$ is DUE if we can map votes and candidates into the real line such that the approved candidates of every vote is at most $r$ far from this vote, where $r$ is a fixed number.

**Weakly single-crossing (WSC)** We say that $E$ is WSC if there is an order of $V$ such that for every pair of candidates $c$ and $c'$, the votes in each of $V_1 = V(c) \setminus V(c')$, $V_2 = V(c') \setminus V(c)$, and $V \setminus (V_1 \cup V_2)$ are consecutive with $V \setminus (V_1 \cup V_2)$ being between $V_1$ and $V_2$. 
3 Relations Among Different Domains

In this section, we investigate the relationships among the restricted domains proposed in this paper and the ones studied by Elkind and Lackner [2015]. Our findings are summarized in the following theorem.

Among these relations, we find that the relations among $\alpha$-TR, VI, and $\beta$-VPTR are particularly interesting. Recall that in $\alpha$-TR, candidates are mapped into vertices of a tree, but in VI votes are mapped into vertices in a path. Nevertheless, we prove that $\alpha$-TR is a special case of VI. The proof is based on the depth-first-search (DFS) traversal of the $\alpha$-TR tree. Further given that VI is a special case of $\beta$-VPTR, we are able to bridge the relations between candidates-mapped tree representations and votes-mapped tree representations. We prove similar relations among $\alpha$-TVP, CI, and $\beta$-PTR.

For two domain restrictions $A$ and $B$, $A \subset B$, which reads $A$ implies $B$, means that every $A$-election is a $B$-election but not necessarily the other way around.

**Theorem 1.** The following relations hold:

- $t$-PART $\subset \alpha$-TR $\subset \beta$-PTR $\subset PTR \subset \beta$-TR.
- $t$-PART $\subset \alpha$-TR $\subset \beta$-VPTR $\subset VPTR \subset VTR = \beta$-VTR.
- $\alpha$-TR $\subset CI \subset \beta$-PTR, and $\alpha$-TR $\subset VI \subset \beta$-VPTR.

Moreover, the relations are complete, in the sense that if a concept of restricted domain has no inclusive relation with another one shown above, there is no such a relation in general. (See Figure 4 for a graphical illustration.)

**Proof.** Let $E = (C, V)$ be an election. We only prove for the relations involving candidates-mapped representations.

- $t$-PART $\subset \alpha$-TR. Let $(C_1, \ldots, C_t)$ be a partition of $C$ such that every vote in $V$ is one of $C_1, \ldots, C_t$. We can construct a tree with vertex set $C \cup \{x\}$ such that $x$ is the root and each $C_i$, $1 \leq i \leq t$, induces a path in the tree with one of its endpoints being a child of $x$.

- $\alpha$-TR $\subset$ VI. Let $T$ be an $\alpha$-TR representation of $E$ with root $x$. For each vote $v \in V$, let $c_v \in C$ be the candidate so that the approved candidates in $v$ are exactly the candidates in the path from $x$ to $c_v$, except $x$. We shall show that ordering the votes according to the DFS traversal numbers of their corresponding candidates leads to a VI ordering of the election. (See page 603 in [Cormen et al., 2009] for the definition of DFS) For each candidate $c \in C$, let DFS($c$) be the DFS traversal number of $c$. Let $v_1, \ldots, v_n$ be an order of $V$ such that for every $i$, $1 \leq i \leq n$, it holds that DFS($c_{v_i}$) $<$ DFS($c_{v_{i+1}}$).

4 Complexity of Recognition

To exploit restricted domains to design algorithms, an important question is how efficiently we can determine whether an election falls into the category of some restricted domain. Many of our concepts are related to special hypergraphs whose recognition algorithms have been studied. We need the following notions for exposition.

A hypergraph $\mathcal{G} = (N, A)$ is a tuple where $N$ is the set of vertices and $A$ the hyperedges each of which is a subset of $N$. A hypergraph can be represented by a bipartite graph where we have $N$ as the vertex set on one side and on the other side we create one vertex $a(e)$ for each hyperedge $e \in A$ which is adjacent to all vertices included in $e$. A hypergraph $(N, A)$ is a tree-hypergraph if there is a tree $T$ with vertex set $N$ such that every hyperedge in $A$ induces a subtree of $T$. Particularly, such a tree $T$ is referred to as a tree-support of $\mathcal{G}$ in the literature. Given a hypergraph $\mathcal{G} = (N, A)$, it has been shown that determining whether it is a tree-hypergraph can be done in polynomial time. More importantly, if $\mathcal{G}$ is a tree-hypergraph, a tree-support of $\mathcal{G}$ can be constructed in $O(n^2 \cdot (m + \log n))$ time where $n = |N|$ and $m$ is the number of hyperedges [Slater, 1978; Klemz et al., 2014; Korach and Stern, 2003; Johnson and Pollak, 1987; Buchin et al., 2011]. It is easy to see that an election $E = (C, V)$
admits a TR (resp. VTR), if and only if the hypergraph corresponding to the bipartite graph $G_E$ with $V$ (resp. $C$) being considered as the hyperedge set is a tree-hypergraph. This results in the following corollary.

**Corollary 1.** Determining whether an election admits a TR (resp. VTR) can be done in polynomial time and, moreover, a TR (VTR) representation can be constructed in $O(|C|^2 \cdot (|V| + \log |C|))$ (resp. $O(|V|^2 \cdot (|C| + \log |V|))$) time.

Brandes et al. [2012] considered path-based tree-supports of hypergraphs and showed that determining whether a hypergraph admits a path-based tree-support can be solved in $O(n^3m)$ time, where $n$ and $m$ are respectively the number of vertices and the number of hyperedges in the hypergraph. Recall that a path-based tree-support of a hypergraph $G = (N, A)$ is a tree-support $T$ of $G$ such that every hyperedge in $A$ induces a path of $T$. Similar to the discussion above, we can obtain the following corollary.

**Corollary 2.** Recognizing whether an election $(C, V)$ is PTR (resp. VPTR) can be done in polynomial time and, if so, a PTR (resp. VPTR) representation can be constructed in $O(|C|^3 \cdot |V|)$ (resp. $O(|V|^3 \cdot |C|)$) time.

A hypergraph $G$ is called a directed path hypergraph if there is an oriented tree such that every hyperedge of $G$ induces a directed path in this tree. Such a tree is called a dipath-based tree-support of $G$. Chaplick et al. [2010] investigated the relation between path-based tree-supports and dipath-based treesupports of hypergraphs, and proved that given a path-based tree-support $T$ of a directed path hypergraph, it is always possible to orient the edges in $T$ in some way to obtain a dipath-based tree-support of $G$. More importantly, any algorithm for constructing a path-based tree-support can be adapted to construct a dipath-based tree-support without increasing the time complexity in the worst case. Note that any hypergraph which has a dipath-based tree-support must have a path-based tree-support as well. Hence, we have the following result.

**Corollary 3.** Determining whether an election admits a $\beta$-PTR (resp. $\beta$-VPTR) can be done in polynomial time, and if so, a $\beta$-PTR (resp. $\beta$-VPTR) representation can be constructed in $O(|C|^3 \cdot |V|)$ (resp. $O(|V|^3 \cdot |C|)$) time.

Finally, we study a polynomial-time algorithm to recognize whether an election admits an $\alpha$-TR or $\alpha$-VTR.

**Theorem 2.** Determining whether an election $(C, V)$ admits an $\alpha$-TR (resp. $\alpha$-VTR) can be done in polynomial time, and, if so, an $\alpha$-TR tree can be constructed in $O(p^2 \cdot m^2)$ time, where $m = |C|$ and $p = \sum_{v \in V} |v|$.  

**Proof.** We only give the proof for $\alpha$-TR. Let $E = (C, V)$ be an election and $G_E$ its incidence graph. Let $m = |C|$ and $p = \sum_{v \in V} |v|$. We first calculate all connected components of $G_E$ which can be done in $O(m + mp)$ time. If $E$ admits an $\alpha$-TR, we shall construct a rooted tree $T$ where candidates in each connected component induce a subtree rooted at a child of the root of $T$. Note that for each connected component we have a subelection $(C', V')$ where $C' \subseteq C$, $V' \subseteq V$, and $C' \cup V'$ is the vertex set of this connected component. Then, it suffices to separately determine if each of these subelections admits an $\alpha$-TR. Particularly, if all subelections admit $\alpha$-TRs, we can get an $\alpha$-TR representation of $(C, V)$ by merging all the roots of these trees. So, let’s now only focus on one connected component $H$ of $G_E$, and let $(C', V')$ be the subelection corresponding to $H$. Observe that if there exists no candidate $c \in C'$ who is approved by all votes in $V'$, $(C', V')$ does not admit an $\alpha$-TR; so does not $(C, V)$. Otherwise, let $c_{\pi(1)}, \ldots, c_{\pi(z)}$ be the candidates in $C'$ each of which is approved in all votes in $V'$. We fix an order $c_{\pi(1)}, \ldots, c_{\pi(z)}$ of these candidates and accordingly construct a path. Then, we remove the candidates $c_{\pi(1)}, \ldots, c_{\pi(z)}$ from $H$, possibly leading to several connected components in $H$, for which we iteratively determine whether the subelection with respect to each of these connected components admits an $\alpha$-TR. If this is the case, similar to the above procedure, we merge all roots of these $\alpha$-TR trees together with $c_{\pi(z)}$. The algorithm can be implemented in $O(p^2m^2)$ time.  

5 Multiwinner Determination

In this section, we explore how different domain restrictions shape the complexity of WINNER DETERMINATION for some approval-based multiwinner voting rules, and how to exploit structures of restricted elections to design tractability algorithms. Let $(C, V)$ be an election and $k$ an integer.

**Chamberlin-Courant approval voting (CCAV)** The score of a committee $w$ is the number of votes intersecting $w$, i.e., $\{|v \in V : v \cap w \neq \emptyset\}$. This rule selects a $k$-committee $w \subseteq C$ with the maximum score.

**Proportional approval voting (PAV)** The score of a committee $w$ received from a vote $v$ is defined as $PAV(v, w) = 1 + \sum_{j \in C} \frac{1}{\left|\frac{v}{v \cap w}\right|}$ if $v \cap w \neq \emptyset$, and is 0 if $v \cap w = \emptyset$. This rule selects a $k$-committee with the maximum total score $\sum_{v \in V} PAV(v, w)$.

**Minimax approval voting (MAV)** The score of a committee $w$ is defined as $\max_{v \in V} (|v \setminus w| + |w \setminus v|)$, and this rule selects a $k$-committee with the minimum score.

Recall that in the WINNER DETERMINATION problem for a rule $\varphi \in \{MAV, CCAV, PAV\}$ (WD-$\varphi$), we are given an election and a rational number $R$, and the question is whether there is a $k$-committee with score at least (resp. most) $R$ for PAV and CCAV (resp. MAV).

It has been shown that in general elections WD-MAV, WD-CCAV, and WD-PAV are all NP-hard [Aziz et al., 2015; LeGrand, 2004; Betzler et al., 2013], which sparks much study of these problems in restricted elections. In particular, Peters [2018] proved that WD-PAV and WD-CCAV are polynomial-time solvable when restricted to CI elections.

As $\alpha$-VTR is a subdomain of CI, the polynomial-time solvability directly transfers to $\alpha$-VTR elections. In addition, WD-CCAV restricted to VI elections is also polynomial-time solvable [Elkind and Lackner, 2015]. However, whether WD-PAV restricted to VI elections is polynomial-time solvable remained as an open question so far. We are not able to resolve
this open question, but based on the dynamic-programming technique we show that this problem is polynomial-time solvable when restricted to a subdomain of VI, namely the $\alpha$-TR domain. In fact, we show this result for a large class of rules.

An a-PAV rule is characterized by a non-increasing vector $a = (a_1, \ldots, a_k)$ of $k$ non-negative real numbers. The score of a $k$-committee $w$ from a vote $v$ is $\sum_{i=1}^{[w,n]} a_i$ if $v \cap w \neq \emptyset$ and is 0 otherwise. This rule selects a $k$-committee with the maximum total score. Therefore, CCAV and PAV are $(1, 0, \ldots, 0)$-PAV and $(1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$-PAV, respectively.

**Theorem 3.** WD-a-PAV restricted to $\alpha$-TR elections is polynomial-time solvable.

**Proof Sketch.** Let $(C, V)$ be an $\alpha$-TR election. Due to Theorem 2, we can construct an $\alpha$-TR tree $T$ of this election in polynomial time. Let $x$ denote the root of $T$. For a candidate $c$, let $P(c)$ be the set of all candidates in the unique path from $x$ to $c$ except $x$ and $c$. Let $T_c$ be the subtree rooted at $c$, $V(T_c)$ the multiset of votes approving at least one candidate in $T_c$, and $C(T_c)$ the set of candidates in $T_c$. A crucial observation is that there exists an optimal $k$-committee such that if some candidate $c$ is in the committee then all candidates in $P(c)$ are also in this committee. Based on this observation, we develop a dynamic-programming algorithm as follows. For each candidate $c$ and a non-negative integer $i \leq |C(T_c)|$, let $sc(c, i)$ denote the a-PAV score of an optimal $(|P(c)| + i)$-committee consisting of exactly $i$ candidates from $T_c$ with respect to the subelection $(P(c) \cup C(T_c), V(T_c))$. Updating $sc(c, i)$, $i \geq 1$, can be reduced to a variant of the classic Knapsack problem. In particular, let $c_1, \ldots, c_t$ be the children of $c$. Note that for any $c_x$ and $c_y$ such that $1 \leq x \neq y \leq t$, it holds that $V(c_x) \cap V(c_y) = \emptyset$. The question is then to find a vector $(i_1, i_2, \ldots, i_t)$ of non-negative integers such that (1) $\sum_{j=1}^{t} i_j = i - 1$; (2) $i_j \leq |C(T_{c_j})|$ for every $1 \leq j \leq t$; and (3) $\sum_{j=1}^{t} sc(c_j, i_j)$ is maximized among all vectors satisfying the first and second conditions. Such a vector can be found in polynomial time using a similar dynamic programming algorithm for the Knapsack problem [Kellerer, 2016]. Given such a vector $(i_1, i_2, \ldots, i_t)$, let

$$sc(c, i) = \sum_{j=1}^{t} sc(c_j, i_j) + \left| V(T_c) \setminus \bigcup_{x \in [t]} V(T_{c_x}) \right| \cdot \frac{|P(c)|+1}{2}.$$

The score of an optimal $k$-committee is then $sc(x, k)$ (regarding $x$ as a candidate not approved by anyone).

The proof is adapted from the reductions in [Aziz et al., 2015; LeGrand, 2004] by introducing a candidate being the center of the star and being approved by all votes.

For CCAV, the above adaption does not work, because in this case if a candidate is approved in all votes, then any $k$-committee including this candidate is a winning $k$-committee. Nevertheless, via a completely different reduction, we show that WD-CCAV remains NP-hard even when restricted to star $\beta$-PTR elections, a special case of PTR.

In a graph, we say a vertex covers an edge if this vertex is one of the endpoints of this edge.

**Theorem 5.** WD-CCAV restricted to star $\beta$-PTR elections is NP-hard.

**Proof.** Let $(G = (A \cup B, F), p, \ell)$ be a CCAV instance where $G$ is a bipartite graph with the bipartition $(A, B)$. Let $n = |F|$ be the number of edges. For each vertex $u \in A \cup B$, we create a candidate $c(u)$. In addition, we create a candidate $d$. For each edge $e = (u, u')$ where $u \in A$ and $u' \in B$, we create a vote $v(e)$ which approves $d$, $c(u)$, and $c(u')$. Moreover, for each vertex $u \in A \cup B$, we create a multiset $V(u)$ of $n$ votes each of which approves only $c(u)$. It is easy to see that the election admits a star $\beta$-PTR, where there is an arc from every candidate $c(u), u \in A$, to the candidate $d$, and an arc from $d$ to every candidate $c(u), u \in B$. We complete the construction by setting $k = p$ and $R = \ell + p \cdot n$. (⇒) Assume that there exists an $S \subseteq A \cup B$ of $p$ vertices in $G$ which covers at least $\ell$ edges. Let $w = \{c(u) : u \in S\}$. We claim that $w$ has CCAV score at least $R$. First, $w$ intersects all $p \cdot n$ votes in $\bigcup_{u \in S} V(u)$. In addition, if an edge $e = (u, u')$ where $u \in A, u' \in B$, is covered by some vertex in $S$, at least one of $c(u)$ and $c(u')$ is in $w$; hence, $w$ intersects the corresponding vote $v(e)$. Since $S$ covers at least $\ell$ edges, $w$ intersects at least $\ell$ votes corresponding to these covered edges. In total, $w$ intersects at least $p \cdot n + \ell = R$ votes. ($\Leftarrow$) Suppose that the constructed election has a $k$-committee $w$ of score at least $R$. Observe that the candidate $d$ cannot be in $w$, since otherwise $w$ can intersect at most $n + (k-1) \cdot n = k \cdot n - 1$ of $R$ votes. So, $w \subseteq \{c(u) : u \in A \cup B\}$. Let $S = \{u \in A \cup B : c(u) \in w\}$ be the set of vertices corresponding to $w$. Clearly, $w$ intersects all $k \cdot n$ votes in $\bigcup_{u \in S} V(u)$. As $R = k \cdot n + \ell$, this implies that $w$ intersects at least $\ell$ edge-votes in $G$. Due to the construction, vertices in $S$ cover all edges corresponding to these edge-votes.

We show that the polynomial-time solvability for VI elections does not extend to VTR elections with even specific underlying tree structures.
Table 1: Entries marked with $\leftrightarrow$ (→) mean that the results are the same as the one immediately on the left (right) side.

<table>
<thead>
<tr>
<th></th>
<th>$\alpha$-TR</th>
<th>$\beta$-PTR</th>
<th>PTR</th>
<th>TR</th>
<th>$\alpha$-VTR</th>
<th>$\beta$-VTR</th>
<th>VPTR</th>
<th>VTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCAV</td>
<td>P [Thm. 3]</td>
<td>NP-h (star) [Thm. 5]</td>
<td>$\leftrightarrow$</td>
<td>$\leftrightarrow$</td>
<td>P [Peters, 2018]</td>
<td>$\rightarrow$</td>
<td>P [Thm. 7]</td>
<td>NP-h [Thm. 6] (star, comb)</td>
</tr>
</tbody>
</table>

We fill the gaps by the following theorems.

**Theorem 7.** WD-CCAV and WD-MAV restricted to star VTR elections are polynomial-time solvable.

**Proof.** We give only the proof for CCAV. Let $(C, V)$ be the election in a given instance. Observe that if there are two candidates $c, c' \in C$ such that $V(c) \subseteq V(c')$, removing $c$ does not change the answer to the instance. Hence, hereinafter, we assume that no such two candidates exist. Let $v_0 \in V$ denote the center and $v_1, \ldots, v_{n-1}$ the leaves in the VTR representation of $(C, V)$. Observe that if $v_0$ does not approve any candidate, all $k$-committees have the same score under the above assumption. Otherwise, let $C'$ be the set of candidates approved by three votes (one of them is $v_0$). Under the above assumption, if $|C'| < k$, any optimal $k$-committee contains $C'$ and any arbitrary $k - |C'|$ candidates from $C \setminus C'$; we are done. If $|C'| \geq k$, we solve the problem by reducing it to the PARTIAL EDGE COVER (PEC) problem which is polynomial-time solvable [Plesnik, 1999]. In particular, for each vote $v_i$, $1 \leq i \leq n - 1$, we create a vertex, denoted still by $v_i$ for simplicity. Then, for every candidate $c \in C'$ such that $V(c) = \{v_0, v_i, v_j\}$ where $1 \leq i \neq j \leq n - 1$, we create an edge between $v_i$ and $v_j$. Now, the question is equivalent to finding a subset of $k$ edges that cover as many vertices as possible. This is exactly a PEC instance.

**Theorem 8.** WD-MAV restricted to star $\beta$-PTR elections is polynomial-time solvable.

6 Conclusion

We studied several restricted domains of dichotomous preferences, where candidates/votes are mapped into vertices of a (rooted, oriented) tree, and votes/candidates induce some specific structures. Particularly, we studied the relations among these domains and the ones in [Elkind and Lackner, 2015] (Figure 4), the complexity of recognizing them, and the complexity of WINNER DETERMINATION restricted to these domains (Table 1). Except DUE, all domains shown in Figure 4 can be recognized in polynomial time. The complexity of determining whether an election satisfies DUE remained open.

**References**
