Elicitation for Preferences Single Peaked on Trees

Palash Dey
Indian Institute of Science, Bangalore
palash@csa.iisc.ernet.in

Neeldhara Misra
Indian Institute of Technology, Gandhinagar
mail@neeldhara.com

Abstract

Eliciting preferences of a set of agents over a set of items is a problem of fundamental interest in artificial intelligence in general and social choice theory in particular. Prior works on preference elicitation focus on unrestricted domain and the domain of single peaked preferences and show that the preferences in single peaked domain can be elicited by much less number of queries compared to unrestricted domain. We extend this line of research and study preference elicitation for single peaked preferences on trees which is a strict superset of the domain of single peaked preferences. We show that the query complexity crucially depends on the number of leaves, the path cover number, and the distance from path of the underlying single peaked tree, whereas the other natural parameters like maximum degree, diameter, pathwidth do not play any direct role in determining query complexity. We then investigate the query complexity for finding a weak Condorcet winner for preferences single peaked on a tree and show that this task has much less query complexity than preference elicitation. Here again we observe that the number of leaves in the underlying single peaked tree and the path cover number of the tree influence the query complexity of the problem.

1 Introduction

In multiagent systems, we often have scenarios where agents have to arrive at a consensus when choosing between multiple options. Typically, the agents have preferences over a set of items, and the problem of aggregating these preferences in a suitable manner is one of the most well-studied problems in social choice theory [Brandt *et al.*, 2015b]. There are many ways of expressing preferences over a set of alternatives. One of the most comprehensive ways is to specify a complete ranking over the set of alternatives. However, one of the downsides of this model is the fact that it can be expensive to solicit a preference when there are a large number of alternatives, and many agents are involved.

Since asking agents to provide their complete rankings is impractical, a popular notion is one of *elicitation*, where we ask agents simple *comparison queries*, such as if they prefer alternative X over Y. This naturally gives rise to the problem of *preference elicitation*, where we hope to recover the complete ranking (or possibly the most relevant part of the ranking) based on a small number of queries.

The paradigm of *voting* is a popular general methodology for aggregating preferences, where one devises "voting rules" for mapping a collection of preferences (which we refer to as votes) to either a *winning alternative* or a *consensus ranking*. Keeping in line with the terminology used in voting, we will refer to alternatives as candidates, and a collection of votes will be termed a preference profile. In the context of a fixed voting rule, we may also want to query the voters up to the point of determining the winner (or the aggregate ranking, as the case may be). Yet another refinement in this setting is when we have prior information about how agents are likely to vote, and we may want to determine which voters to query first, to be able to quickly rule out a large number of alternatives, as explored by [Conitzer and Sandholm, 2002].

When our goal is to elicit preferences that have no prior structure, one can demonstrate scenarios where it is imperative to ask each agent (almost) as many queries as would be required to determine an arbitrary ranking. However, in recent times, there has been considerable interest in voting profiles that are endowed with additional structure. The motivation for this is two-fold. The first is that in several application scenarios commonly considered, it is rare that votes are ad-hoc, demonstrating no patterns whatsoever. For example, the notion of *single-peaked preferences*, which we will soon discuss at length, forms the basis of several studies in the analytical political sciences [Hinich and Munger, 1997]. In his work on eliciting preferences that demonstrate the "single-peaked" structure, [Conitzer, 2009] argues that the notion of single-peakedness is also a reasonable restriction in applications outside of the domain of political elections.

The second motivation for studying restricted preferences is somewhat more technical, but is just as compelling. To understand why structured preferences have received considerable attention from social choice theorists, we must first take a brief detour into some of the foundational ideas that have shaped the landscape of voting rules as we understand them today. As it turns out, the axiomatic approach of social choice involves defining certain "properties" that formally capture the quality of a voting rule. For example, we would not want

a voting rule to be, informally speaking, a *dictatorship*, which would essentially mean that it discards all but one voter's input. Unfortunately, a series of cornerstone results establish that it is impossible to devise voting rules which respect some of the simplest desirable properties. Indeed, the classic works of Arrow [Arrow, 1950] and Gibbard-Satterthwaite [Gibbard, 1973; Satterthwaite, 1975] show that there is no straight forward way to simultaneously deal with properties like voting paradoxes, strategy-proofness, nondictatorship, unanimity etc. We refer to [Moulin, 1991] for a more elaborate discussion. Making the matter worse, many classical voting rules turn out to be computationally intractable.

This brings us to the second reason for why structured preferences are an important consideration. The notion of single-peakedness that we mentioned earlier is an excellent illustration (we refer the reader to Section 2 for the formal definition). Introduced by [Black, 1948], it not only captures the essence of structure in political elections, but also turns out to be extremely conducive to many natural theoretical considerations. To begin with, one can devise voting rules that are "nice" with respect to several properties, when preferences are single-peaked. Further, they are structurally elegant from the view of winner determination, since they always admit a *weak Condorcet winner* — a candidate which is not defeated by any other candidate in pairwise election — thus working around the Condorcet paradox which is otherwise a prominent concern in the general scenario. In a landmark contribution, [Brandt *et al.*, 2015a] show that several computational problems that are intractable in the general setting become polynomially solvable when we consider single-peaked preferences.

A natural question at this point is if the problem of elicitation becomes any easier — that is, if we can get away with fewer queries — by taking advantage of the structure provided by single-peakedness. It turns out that the answer to this is in the affirmative, as shown in a detailed study by [Conitzer, 2009]. The definition of single-peakedness involves an ordering over the candidates (called the *harmonious ordering* by some authors). The work of [Conitzer, 2009] shows that $\mathcal{O}(mn)$ queries suffice, assuming either that the harmonious ordering is given, or one of the votes is known. Dey and Misra [Dey and Misra, 2016b] show a query complexity bound of $\mathcal{O}(mn)$ for the domain of single crossing profiles and a large number of voters.

We now return to the theme of structural restrictions on preferences. As it turns out, the single peaked preference domain has subsequently been generalized to single peakedness on trees (roughly speaking, these are profiles that are single peaked on every path) [Demange, 1982; Trick, 1989]. This is a class that continues to exhibit many desirable properties of single peaked domains. For example, there always exists a weak Condorcet winner and further, many voting rules that are intractable in unrestricted domain are polynomial time computable if the underlying single peaked tree is "nice" [Yu *et al.*, 2013; Peters and Elkind, 2016]. We note the class of profiles that are single peaked on trees are substantially more general than the class of single peaked preferences. Note that the latter is a special case since a path is, in particular, a tree. Our work here addresses the issue of elicitation on profiles

that are single-peaked on trees, and can be seen as a significant generalization of the results in [Conitzer, 2009]. We now detail the specifics of our contributions.

Our Contributions. We study the query complexity for preference elicitation when the preference profile is single peaked on a tree. We provide *tight* connections between various parameters of the underlying tree and the query complexity for preference elicitation. Our broad goal is to provide a suitable generalization of preference elicitation for single peaked profiles to profiles that are single peaked on trees. Therefore, we consider various ways of quantifying the "closeness" of a tree to a path, and reflect on how these measures might factor into the query complexity of an algorithm that is actively exploiting the underlying tree structure.

We summarize our results for preference elicitation in Table 1, where the readers will note that most of the parameters (except diameter) chosen are small constants (typically zero, one or two) when the tree under consideration is a path. Observe that in some cases — such as the number of leaves, or the path cover number — the dependence on the parameter is transparent (and we recover the results of [Conitzer, 2009] as a special case), while in other cases, it is clear that the perspective provides no additional mileage (the non-trivial results here are the matching lower bounds).

In terms of technique, our strategy is to "scoop out the paths from the tree" and use known algorithms to efficiently elicit the preference on the parts of the trees that are paths. We then efficiently merge this information across the board, and that aspect of the algorithm varies depending on the parameter considered. The lower bounds typically come from trees that provide large "degrees of freedom" in reordering candidates, typically these are trees that don't have too many long paths (such as stars). The arguments are often subtle but intuitive.

We then study the query complexity for finding a weak Condorcet winner of a preference profile which is single peaked on a tree. Here, we are able to show that a weak Condorcet winner can be found with far fewer queries than the corresponding elicitation problem. In particular, we establish that a weak Condorcet winner can be found using $\mathcal{O}(mn)$ many queries for profiles that are single peaked on trees [Theorem 7], and we also show that this bound is the best that we can hope for [Theorem 10]. We also consider the problem for the special case of single peaked profiles. While [Conitzer, 2009] showed that $\Omega(mn)$ queries are necessary to determine the *aggregate ranking*, we show that only $\mathcal{O}(n \log m)$ queries suffice if we are just interested in (one of the) weak Condorcet winners. Moreover, we show this bound is tight under the condition that the algorithm does not interleave queries to different voters [Theorem 11] (our algorithm indeed satisfies this condition).

Finally, expressing the query complexity for determining a weak Condorcet winner in terms of a measure of closeness to a path, we show an algorithm with query complexity $\mathcal{O}(nk \log m)$ where k is the path cover number of $\mathcal{T}$ [Theorem 9] or the number of leaves in $\mathcal{T}$ [Corollary 6].

To summarize, we remark that our results non-trivially

Parameter	Upper Bound	Lower Bound
Path width (w)	$\mathcal{O}(mn \log m)$ Observation 1	$\Omega(mn \log m)$ even for $w = 1, \log m$ Corollary 3
Maximum degree (Δ)	$\mathcal{O}(mn \log m)$ Observation 1	$\Omega(mn \log m)$ even for $\Delta = 3, m - 1$ Corollary 4
Path cover number (k)	$\mathcal{O}(mn \log k)$ Theorem 2	$\Omega(mn \log k)$ Corollary 2
Number of leaves (ℓ)	$\mathcal{O}(mn \log \ell)$ Corollary 1	$\Omega(mn \log \ell)$ Theorem 4
Distance from path (d)	$\mathcal{O}(mn + nd \log d)$ Theorem 3	$\Omega(mn + nd \log d)$ Theorem 6
Diameter (ω)	$\mathcal{O}(mn \log m)$ Observation 1	$\Omega(mn \log m)$ even for $\omega = 2, {}^m\!/_2$ Corollary 5

Table 1: Summary of query complexity bounds for PREFERENCE ELICITATION.

generalize earlier works on query complexity for preference elicitation in [Conitzer, 2009]. We believe revisiting the preference elicitation problem in the context of profiles that are single peaked on trees is timely, and that this work also provides fresh algorithmic and structural insights on the domain of preferences that are single peaked on trees.

Related Work. We have already mentioned the work of [Conitzer, 2009] addressing the question of eliciting preferences in single-peaked profiles, which is the closest predecessor to our work. Before this, Conitzer and Sandholm addressed the computational hardness for querying minimally for winner determination [Conitzer and Sandholm, 2002]. They also prove that one would need to make $\Omega(mn \log m)$ queries even to decide the winner for many commonly used voting rules [Conitzer and Sandholm, 2005] which matches with the trivial $\mathcal{O}(mn \log m)$ upper bound for preference elicitation in unrestricted domain based on sorting. Ding and Lin study preference elicitation under partial information setting and show interesting properties of what they call a deciding set of queries [Ding and Lin, 2013]. Lu and Boutilier provide empirical study of preference elicitation under probabilistic preference model [Lu and Boutilier, 2011b] and devise several novel heuristics which often work well in practice [Lu and Boutilier, 2011a].

2 Preliminaries

For a positive integer n, we denote the set $\{1, \ldots, n\}$ by $[n]$. Suppose we have a set $\mathcal{V} = \{v_i : i \in [n]\}$ of n *voters* each of which has a preference $\succ_i, i \in [n]$, alternatively called *vote*, which is a complete order over a set $\mathcal{C} = \{c_j : j \in [m]\}$ of m *candidates*. We denote the set of preferences over $\mathcal{C}$ by $\mathcal{L}(\mathcal{C})$. The tuple $(\succ_i)_{i \in [n]}$ of the preferences of all the voters is called a *profile*. If not mentioned otherwise, we use m and n to denote the number of candidates and the number of voters respectively. Let $\succ \in \mathcal{L}(\mathcal{C})$ be any complete order over $\mathcal{C}$. For a subset $\mathcal{X} \subseteq \mathcal{C}$ of candidates, we denote the restriction of an order $\succ$ to the subset of candidates $\mathcal{X}$ by $\succ (\mathcal{X})$. We denote the restriction of a profile $\mathcal{P} = (\succ_i)_{i \in [n]}$ to $\mathcal{X}$ by $\mathcal{P}(\mathcal{X}) = (\succ_i (\mathcal{X}))_{i \in [n]}$. We say that a candidate $x \in \mathcal{C}$ is placed at the ith position of a preference $\succ \in \mathcal{L}(\mathcal{C})$ if x is preferred over all but exactly $(i - 1)$ candidates in $\succ$. Given an n voters profile $\mathcal{P} = (\succ_i)_{i \in [n]}$, a candidate x is called a *Condorcet winner* of $\mathcal{P}$ if for every other candidate y, a *strict* majority of the voters prefer x over y; that is $|\{i \in [n] : x \succ_i y\}| > {}^n\!/_2$ for every $y \in \mathcal{C} \setminus \{x\}$. A candidate x is called a *weak Condorcet winner*

of an n voters profile $\mathcal{P} = (\succ_i)_{i \in [n]}$ if there does not exist any other candidate y whom a strict majority of the voters prefer over x; that is $|\{i \in [n] : y \succ_i x\}| \le {}^n\!/_2$ for every $y \in \mathcal{C} \setminus \{x\}$. A preference $\succ \in \mathcal{L}(\mathcal{C})$ over a set of candidates $\mathcal{C}$ is called *single peaked* with respect to an order $\succ' \in \mathcal{L}(\mathcal{C})$ if, for every candidates $x, y \in \mathcal{C}$, we have $x \succ y$ whenever we have either $c \succ' x \succ' y$ or $y \succ' x \succ' c$, where $c \in \mathcal{C}$ is the candidate at the first position of $\succ$. A profile $\mathcal{P} = (\succ_i)_{i \in [n]}$ is called single peaked with respect to an order $\succ' \in \mathcal{L}(\mathcal{C})$ if $\succ_i$ is single peaked with respect to $\succ'$ for every $i \in [n]$. Given a path $\mathcal{Q} = (x_1, x_2, \ldots, x_\ell)$ from a vertex x_1 to another vertex x_ℓ in a tree $\mathcal{T}$, we define the order induced by the path $\mathcal{Q}$ to be $x_1 \succ x_2 \succ \cdots \succ x_\ell$. Given a tree $\mathcal{T} = (\mathcal{C}, \mathcal{E})$ with the set of nodes as the set of candidates $\mathcal{C}$, a profile $\mathcal{P}$ is called single peaked on the tree $\mathcal{T}$ if $\mathcal{P}$ is single peaked on the order induced by every path of the tree $\mathcal{T}$; that is for every two candidates $x, y \in \mathcal{C}$, the profile $\mathcal{P}(\mathcal{X})$ is single peaked with respect to the order $\succ$ of the candidates $\mathcal{X}$ induced by the unique path from x to y in $\mathcal{T}$. We call the tree $\mathcal{T}$ the *underlying single peaked tree*. It is known (c.f. [Demange, 1982]) that there always exists a weakly Condorcet winner for a profile $\mathcal{P}$ which is single peaked on a tree $\mathcal{T}$.

Trees. The following definitions pertaining to the structural aspects of trees will be useful.

- The *pathwidth* of $\mathcal{T}$ is the minimum width of a *path decomposition* of $\mathcal{T}$ [Heinrich, 1993].

- A set of disjoint paths $\mathcal{Q} = \{Q_1 = (\mathcal{X}_1, \mathcal{E}_1), \ldots, Q_k = (\mathcal{X}_k, \mathcal{E}_k)\}$ is said to cover a tree $\mathcal{T} = (\mathcal{X}, \mathcal{E})$ if $\mathcal{X} = \cup_{i \in [k]} \mathcal{X}_i, \mathcal{E}_i \subseteq \mathcal{E}, \mathcal{X}_i \cap \mathcal{X}_j = \varnothing, \mathcal{E}_i \cap \mathcal{E}_j = \varnothing$ for every $i, j \in [k]$ with $i \ne j$. The *path cover number* of $\mathcal{T}$ is the cardinality of the smallest set $\mathcal{Q}$ of *disjoint* paths that cover $\mathcal{T}$.

- The *distance of a tree $\mathcal{T}$ from a path* is the smallest number of nodes whose removal makes the tree a path.

- The *diameter* of a tree $\mathcal{T}$ is the number of edges in the longest path in $\mathcal{T}$.

We also list some definitions of *subclasses* of trees (which are special types of trees, see also Figure 1).

- A tree is a *star* if there is a *center vertex* and every other vertex is a neighbor of this vertex.

- A tree is a *subdivision of a star* if it can be constructed by replacing each edge of a star by a path.
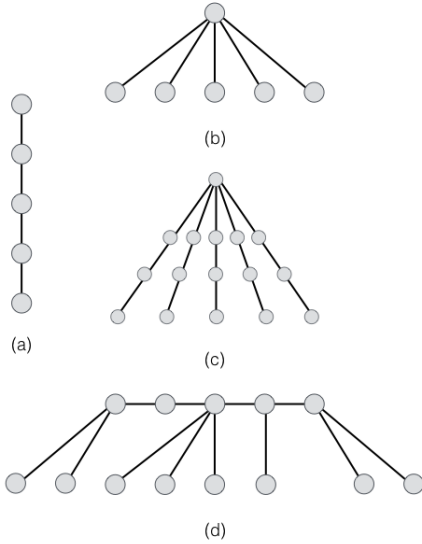
Figure 1: Depicting classes of trees: (a) a path, (b) a star, (c) a (balanced) subdivision of a star, (d) a caterpillar.

- A subdivision of a star is called *balanced* if there exists an integer ℓ such that the distance of every leaf node from the center is ℓ.
- A tree is a *caterpillar* if there is a *central path* and every other vertex is at a distance of one it.
- A tree is a *complete binary tree* rooted at r if every non-leaf node has exactly two children and there exists an integer h, called the height of the tree, such that every leaf node is at a distance of either h or $h-1$ from the root node r.

Problem Definitions and Known Results. Suppose we have a profile $\mathcal{P}$ with n voters and m candidates. For any pair of distinct candidates x and y, and a voter $\ell \in [n]$, we introduce the boolean-valued function QUERY($x >_\ell y$). The output of this function is TRUE if the voter ℓ prefers the candidate x over the candidate y and FALSE otherwise. We now formally state the two problems that we consider in this paper.

Definition 1 PREFERENCE ELICITATION
Given a tree $\mathcal{T} = (\mathcal{C}, \mathcal{E})$ and an oracle access to the function QUERY($\cdot$) *for a profile $\mathcal{P}$ which is single peaked on $\mathcal{T}$, find $\mathcal{P}$.*

Definition 2 WEAK CONDORCET WINNER
Given a tree $\mathcal{T} = (\mathcal{C}, \mathcal{E})$ and an oracle access to the function QUERY($\cdot$) *for a profile $\mathcal{P}$ which is single peaked on $\mathcal{T}$, find a weak Condorcet winner of $\mathcal{P}$.*

Suppose we have a set of candidates $\mathcal{C} = \{c_1, \dots, c_m\}$. We say that an algorithm $\mathcal{A}$ makes q queries if there are exactly q distinct tuples $(\ell, c_i, c_j) \in [n] \times \mathcal{C} \times \mathcal{C}$ with $i < j$ such that $\mathcal{A}$ calls QUERY($c_i >_\ell c_j$) or QUERY($c_j >_\ell c_i$). We call the number of queries made by an algorithm $\mathcal{A}$ its *query complexity*.

We state some known results that we will appeal to later. The first observation employs a sorting algorithm like merge

sort to elicit every vote with $\mathcal{O}(m \log m)$ queries, while the second follows from the linear-time merge subroutine of merge sort ([Cormen, 2009]).

Observation 1 *There is a* PREFERENCE ELICITATION *algorithm with query complexity $\mathcal{O}(mn \log m)$.*

Observation 2 *Suppose $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}$ form a partition of $\mathcal{C}$ and $>$ is a ranking of the candidates in $\mathcal{C}$. Then there is a polynomial time algorithm that finds $>$ given $> (\mathcal{C}_1)$ and $> (\mathcal{C}_2)$ with query complexity $\mathcal{O}(|\mathcal{C}|)$.*

Theorem 1 *[Conitzer, 2009] There is a* PREFERENCE ELICITATION *algorithm with query complexity $\mathcal{O}(mn)$ for single peaked profiles.*

3 Results for PREFERENCE ELICITATION

In this section, we present our results for PREFERENCE ELICITATION for profiles that are single peaked on trees. Recall that we would like to generalize Theorem 1 in a way to profiles that are single peaked on trees. Since the usual single peaked profiles can be viewed as profiles single peaked with respect to a path, we propose the following measures of how much a tree resembles a path.

- *Leaves.* Recall any tree has at least two leaves, and paths are the trees that have exactly two leaves. We consider the class of trees that have ℓ leaves, and show an algorithm with a query complexity of $\mathcal{O}(mn \log \ell)$.
- *Path Cover.* Consider the notion of a *path cover number* of a tree, which is the smallest number of disjoint paths that the tree can be partitioned into. Clearly, the path cover number of a path is one; and for trees that can be covered with k paths, we show an algorithm with query complexity $\mathcal{O}(mn \log k)$.
- *Distance from Paths.* Let d be the size of the smallest set of vertices whose removal makes the tree a path. Again, if the tree is a path, then the said set is simply the empty set. For trees that are at a distance d from being a path (in the sense of vertex deletion), we provide an algorithm with query complexity $\mathcal{O}(mn \log d)$.
- *Pathwidth and Maximum Degree.* Finally, we note that paths are also trees that have pathwidth one, and maximum degree two. These perspectives turn out to be less useful: in particular, there are trees where these parameters are constant, for which we show that elicitation is as hard as it would be on an arbitrary profile, and therefore the easy algorithm from Observation 1 is actually the best that we can hope for.

For the first three perspectives that we employ, that seemingly capture an appropriate aspect of paths and carry it forward to trees, the query complexities that we obtain are tight — we have matching lower bounds in all cases. Also, while considering structural parameters, it is natural to wonder if there is a class of trees that are incomparable with paths but effective for elicitation. Our attempt in this direction is to consider trees of bounded diameter. However, again, we find that this is not useful, as we have examples to show that there exist trees of diameter two that are as hard to elicit as general profiles.

We remark at this point that all these parameters are polynomially computable for trees, making the algorithmic results viable. Also, for the parameters of pathwidth, maximum degree and diameter, we show lower bounds on trees where these parameters are large (such as trees with pathwidth $O(\log m)$, maximum degree $m-1$, and diameter $m/2$), which — roughly speaking — also rules out the possibility of getting a good inverse dependence. As a concrete example, motivated by the $O(mn)$ algorithm for paths, which have diameter m, one might wonder if there is an algorithm with query complexity $O(\frac{mn \log m}{\log \omega})$. This possibility, in particular, is ruled out. We are now ready to discuss the results in Table 1.

3.1 Algorithmic Results

We now present our main algorithmic results. We begin with generalizing the result of Theorem 1 to any single peaked profiles on trees whose path cover number is at most k. The idea is to partition the tree into k disjoint paths, use the algorithm from Theorem 1 on each paths to obtain an order of the candidates on each path of the partition, and finally merge these suborders intelligently. We now formalize this idea as follows.

Theorem 2 *There is a* PREFERENCE ELICITATION *algorithm with query complexity $O(mn \log k)$ for profiles that are single peaked on trees with path cover number at most k.*

Proof: Since the path cover number is at most k, We can partition the tree $\mathcal{T} = (\mathcal{C}, \mathcal{E})$ into t disjoint paths $\mathcal{P}_i = (\mathcal{C}_i, \mathcal{E}_i), i \in [t]$, where t is at most k. We now show that we can elicit any preference $\succ$ which is single peaked on the tree $\mathcal{T}$ by making $O(m \log t)$ queries which in turn proves the statement. We first find the preference ordering restricted to $\mathcal{C}_i$ using Theorem 1 by making $O(|\mathcal{C}_i|)$ queries for every $i \in [t]$. This step needs $\sum_{i \in [t]} O(|\mathcal{C}_i|) = O(m)$ queries since $\mathcal{C}_i, i \in [t]$ forms a *partition* of $\mathcal{C}$. We next merge the t orders $\succ (\mathcal{C}_i), i \in [t]$, to obtain the complete preference $\succ$ by using a standard divide and conquer approach for t-way merging which makes $O(m \log t)$ queries [Hopcroft *et al.*, 1983]. Thus the query complexity of the algorithms is $O(m + m \log t) = O(m \log k)$. □

Towards our results on leaves, we show that the path cover of a tree with ℓ leaves is at least $\lfloor \ell/2 \rfloor$ and at most ℓ. The lower bound is easy since every path can account for at most two leaves. The upper bound comes from a careful partitioning of the nodes following maximal paths from leaves up to a root and using a careful marking scheme. Using the fact that a tree with ℓ leaves can be partitioned into at most ℓ paths, we can use the algorithm from Theorem 2 to obtain the following bound in terms of leaves.

Corollary 1 *There is a* PREFERENCE ELICITATION *algorithm with query complexity $O(mn \log \ell)$ for profiles that are single peaked on trees with at most ℓ leaves.*

Finally, if we are given a subset of vertices whose removal makes the given tree a path, then we have an elicitation algorithm that makes $O(mn + nd \log d)$. As before, we can determine the ordering among the candidates on the path with

$O(m - d)$ queries, and we determine the ordering among the rest in $O(d \log d)$ queries using Observation 1, and finally merge using Observation 2. This leads us to the following. (We defer the proofs of the results which are marked star to full version of the paper [Dey and Misra, 2016a].)

Theorem 3 *There is a* PREFERENCE ELICITATION *algorithm with query complexity $O(mn + nd \log d)$ for profiles that are single peaked on trees with distance d from path.*

3.2 Lower Bounds

We now turn to the lower bounds. Our first result is based on a counting argument, showing that the query complexity in terms of the number of leaves, given by Corollary 1, is tight up to constant factors. Indeed, let us consider a subdivision of a star with ℓ leaves and let t denote the distance from the center, so that we have a total of $t\ell + 1$ vertices. One can show that if the candidates are written out in level order, the candidates that are distance i from the star can be ordered arbitrarily within this ordering. This tells us that the number of possible preferences $\succ$ that are single peaked on the tree $\mathcal{T}$ is at least $(\ell!)^t$. We obtain the lower bound by using a decision tree argument, wherein we are able to show that it is always possible for an oracle answering the comparison queries to answer in such a way that the total space of possibilities decreases by at most a factor of half. Since the tree must entertain at least $(\ell!)^t$ leaves to account for all possibilities, we obtain the claimed lower bound.

Theorem 4 [⋆] *Let $\mathcal{T} = (\mathcal{C}, \mathcal{E})$ be a balanced subdivision of a star with ℓ leaves. Then any* PREFERENCE ELICITATION *algorithm for single peaked profiles on $\mathcal{T}$ has query complexity $\Omega(mn \log \ell)$.*

Since the path cover number of a subdivided star on ℓ leaves is at least $\ell/2$, we also have the following.

Corollary 2 *There exists a tree $\mathcal{T}$ with path cover number k such that any* PREFERENCE ELICITATION *algorithm for single peaked profiles on $\mathcal{T}$ has query complexity $\Omega(mn \log k)$.*

Mimicking the level order argument above on a generic tree with ℓ leaves, and using the connection between path cover and leaves, we obtain lower bounds that are functions of (n, ℓ) and (n, k), as given below. This will be useful for our subsequent results.

Theorem 5 [⋆] *Let $\mathcal{T} = (\mathcal{C}, \mathcal{E})$ be any arbitrary tree with ℓ leaves and path cover number k. Then any* PREFERENCE ELICITATION *algorithm for single peaked profiles on $\mathcal{T}$ has query complexity $\Omega(n\ell \log \ell)$ and $\Omega(nk \log k)$.*

The following results can be obtained simply by applying Theorem 5 on particular graphs. For instance, we use the fact that stars have $(m - 1)$ leaves and have pathwidth one to obtain the first part of Corollary 3, while appealing to complete binary trees that have $O(m)$ leaves and pathwidth $O(\log m)$ for the second part. These examples also work in the context of maximum degree, while for diameter we use stars and caterpillars with a central path of length $m/2$.

Corollary 3 *There exist two trees $\mathcal{T}$ and $\mathcal{T}'$ with pathwidths one and $\log m$ respectively such that any* PREFERENCE ELICITATION *algorithm for single peaked profiles on $\mathcal{T}$ and $\mathcal{T}'$ respectively has query complexity $\Omega(mn \log m)$.*

Corollary 4 *There exist two trees $\mathcal{T}$ and $\mathcal{T}'$ with maximum degree $\Delta = 3$ and $m - 1$ respectively such that any* PREFERENCE ELICITATION *algorithm for single peaked profiles on $\mathcal{T}$ and $\mathcal{T}'$ respectively has query complexity $\Omega(mn \log m)$.*

Corollary 5 *There exists two trees $\mathcal{T}$ and $\mathcal{T}'$ with diameters $\omega = 2$ and $\omega = {}^{m}/_{2}$ respectively such that any* PREFERENCE ELICITATION *algorithm for profiles which are single peaked on $\mathcal{T}$ and $\mathcal{T}'$ respectively has query complexity $\Omega(mn \log m)$.*

Our final result, which again follows from Theorem 5 applied of caterpillar graphs with a central path of length $m - d$, shows that the bound in Theorem 3 is tight.

Theorem 6 [$\star$] *For any integers m and d with $1 \leq d \leq {}^{m}/_{4}$, there exists a tree $\mathcal{T}$ with distance d from path such that any* PREFERENCE ELICITATION *algorithm for profiles which are single peaked on $\mathcal{T}$ has query complexity $\Omega(mn + nd \log d)$.*

4 WEAK CONDORCET WINNER

We now show that we can find a weak Condorcet winner of profiles that are single peaked on trees using fewer queries than the number of queries needed to find the profile itself. First, note that if a Condorcet winner is guaranteed to exist, then it can be found using $\mathcal{O}(mn)$ queries — we pit an arbitrary pair of candidates x, y and use $\mathcal{O}(n)$ queries to determine if x defeats y. We push the winning candidate forward and repeat the procedure, clearly requiring at most m rounds. Now, if a profile is single peaked with respect to a tree, and there are an odd number of voters, then we have a Condorcet winner and the procedure that we just described would work. Otherwise, we simply find a Condorcet winner among the first $(n - 1)$ voters. It can be shown that such a winner is one of the weak Condorcet winners for the overall profile, and we therefore have the following upper bound.

Theorem 7 [$\star$] *There is a* WEAK CONDORCET WINNER *algorithm with query complexity $\mathcal{O}(mn)$ for single peaked profiles on trees.*

For the special case of single peaked profiles, we can do even better. Here we take advantage of the fact that a "median candidate" [Mas-Colell *et al.*, 1995] is guaranteed to be a weak Condorcet winner. We make $\mathcal{O}(\log m)$ queries per vote to find the candidates placed at the first position of all the votes using the algorithm in [Conitzer, 2009] and find a median candidate to show the following.

Theorem 8 [$\star$] *There is a* WEAK CONDORCET WINNER *algorithm with query complexity $\mathcal{O}(n \log m)$ for single peaked profiles (on a path).*

The next result uses Theorem 8 on paths in a path cover eliminating the case of even number of voters by the idea of setting aside one voter that was used in Theorem 7.

Theorem 9 [$\star$] *Let $\mathcal{T}$ be a tree with path cover number at most k. Then there is an algorithm for* WEAK CONDORCET WINNER *for profiles which are single peaked on $\mathcal{T}$ with query complexity $\mathcal{O}(nk \log m)$.*

Recalling that the number of leaves bounds the path cover number, we have the following consequence.

Corollary 6 *Let $\mathcal{T}$ be a tree with ℓ leaves. Then there is an algorithm for* WEAK CONDORCET WINNER *for profiles which are single peaked on $\mathcal{T}$ with query complexity $\mathcal{O}(n\ell \log m)$.*

We now state the lower bounds pertaining to WEAK CONDORCET WINNER. First, we show that any algorithm for single peaked profiles on stars has query complexity $\Omega(mn)$, showing that the bound of Theorem 7 is tight.

Theorem 10 *Any* WEAK CONDORCET WINNER *algorithm for single peaked profiles on stars must have query complexity $\Omega(mn)$.*

Proof: Let $\mathcal{T}$ be a star with center vertex c. We now design an oracle that will "force" any WEAK CONDORCET WINNER algorithm $\mathcal{A}$ for single peaked profiles on $\mathcal{T}$ to make $\Omega(mn)$ queries. For every voter v, the oracle maintains a set of *"marked"* candidates which can not be placed at the first position of the preference of v. Suppose the oracle receives a query to compare two candidates x and y for a voter ℓ. If the order between x and y for the voter ℓ follows from the answers the oracle has already provided to all the queries for the voter ℓ, then the oracle answers accordingly. Otherwise it answers $x >_{\ell} y$ if y is unmarked and marks y; otherwise the oracle answers $y >_{\ell} x$ and marks x. Notice that the oracle marks at most one unmarked candidate every time it is queried. We now claim that there must be at least ${}^{n}/_{10}$ votes which have been queried at least ${}^{m}/_{4}$ times. If not, then there exists $n - {}^{n}/_{10} = {}^{9n}/_{10}$ votes each of which has at least $m - {}^{m}/_{4} = {}^{3m}/_{4}$ candidates unmarked. In such a scenario, there exists a constant N_0 such that for every $m, n > N_0$, we have at least two candidates x and y who are unmarked in at least $(\lfloor {}^{n}/_{2} \rfloor + 1)$ votes each. Now if the algorithm outputs x, then we put y at the first position in at least $(\lfloor {}^{n}/_{2} \rfloor + 1)$ votes and at the second position in the rest of the votes and this makes y the (unique) Condorcet winner. If the algorithm does not output x, then we put x at the first position in at least $(\lfloor {}^{n}/_{2} \rfloor + 1)$ votes and at the second position in the rest of the votes and this makes x the (unique) Condorcet winner. Hence the algorithm fails to output correctly in both the cases contradicting the correctness of the algorithm. Also the resulting profile is single peaked on $\mathcal{T}$ with center at y in the first case and at x in the second case. Therefore the algorithm $\mathcal{A}$ must have query complexity $\Omega(mn)$. $\square$

Our concluding result uses an intricate adversary argument, and shows that the query complexity for WEAK CONDORCET WINNER for single peaked profiles in Theorem 8 is essentially optimal, provided that the queries to different voters are not interleaved, as is the case with our algorithm.

Theorem 11 *Any* WEAK CONDORCET WINNER *algorithm for single peaked profiles which does not interleave the queries to different voters has query complexity $\Omega(n \log m)$.*

References

[Arrow, 1950] Kenneth J Arrow. A difficulty in the concept of social welfare. *The Journal of Political Economy*, pages 328–346, 1950.

[Black, 1948] Duncan Black. On the rationale of group decision-making. *The Journal of Political Economy*, pages 23–34, 1948.

[Brandt *et al.*, 2015a] Felix Brandt, Markus Brill, Edith Hemaspaandra, and Lane A Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *Journal of Artificial Intelligence Research (JAIR)*, pages 439–496, 2015.

[Brandt *et al.*, 2015b] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia. Handbook of computational social choice, 2015.

[Conitzer and Sandholm, 2002] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 392–397, 2002.

[Conitzer and Sandholm, 2005] Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *Proceedings of the 6th ACM conference on Electronic Commerce (EC)*, pages 78–87. ACM, 2005.

[Conitzer, 2009] Vincent Conitzer. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research (JAIR)*, 35:161–191, 2009.

[Cormen, 2009] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[Demange, 1982] Gabrielle Demange. Single-peaked orders on a tree. *Mathematical Social Sciences*, 3(4):389–396, 1982.

[Dey and Misra, 2016a] Palash Dey and Neeldhara Misra. Elicitation for preferences single peaked on trees. https://arxiv.org/abs/1604.04403, 2016.

[Dey and Misra, 2016b] Palash Dey and Neeldhara Misra. Preference elicitation for single crossing domain. In *IJCAI 2016, Proceedings of the 25nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[Ding and Lin, 2013] Ning Ding and Fangzhen Lin. Voting with partial information: what questions to ask? In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 1237–1238. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[Gibbard, 1973] Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica: Journal of the Econometric Society*, pages 587–601, 1973.

[Heinrich, 1993] Katherine Heinrich. Path decomposition. *Le matematiche*, 47(2):241–258, 1993.

[Hinich and Munger, 1997] Mel vin J Hinich and Michael C Munger. *Analytical politics*. Cambridge University Press, 1997.

[Hopcroft *et al.*, 1983] John E Hopcroft, Jeffrey David Ullman, and Alfred Vaino Aho. *Data structures and algorithms*, volume 175. Addison-Wesley Boston, MA, USA:, 1983.

[Lu and Boutilier, 2011a] Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 287–293, 2011.

[Lu and Boutilier, 2011b] Tyler Lu and Craig Boutilier. Vote elicitation with probabilistic preference models: Empirical estimation and cost tradeoffs. In *Algorithmic Decision Theory*, pages 135–149. Springer, 2011.

[Mas-Colell *et al.*, 1995] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. *Microeconomic theory*, volume 1. Oxford university press New York, 1995.

[Moulin, 1991] Hervi Moulin. *Axioms of cooperative decision making*. Number 15. Cambridge University Press, 1991.

[Peters and Elkind, 2016] Dominik Peters and Edith Elkind. Preferences single-peaked on nice trees. In *Proceedings of the 30th International Conference on Artificial Intelligence (AAAI)*, 2016.

[Satterthwaite, 1975] Mark Allen Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[Trick, 1989] Michael A Trick. Recognizing single-peaked preferences on a tree. *Mathematical Social Sciences*, 17(3):329–334, 1989.

[Yu *et al.*, 2013] Lan Yu, Hau Chan, and Edith Elkind. Multiwinner elections under preferences that are single-peaked on a tree. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 425–431. AAAI Press, 2013.