

# Query Complexity of Tournament Solutions

Jahnvi Patel

Advisor: Dr. Neeldhara Misra  
SRIP, IIT Gandhinagar

May 2016

## Abstract

*Computational social choice is an interdisciplinary field that is concerned with the application of techniques developed in computer science, such as complexity analysis or algorithm design, to the study of social choice mechanisms, such as voting procedures or fair allocation of resources among stakeholders. On the other hand, it associates concepts of preference aggregation mechanism to a society of autonomous software agents. We know that a number of voting related problems are fixed parameter tractable(FPT) in terms of number of candidates. This project, in short, aims to survey some FPT and parametrised complexity results for problems arising in computational social choice such as the query complexity of tournament solutions.*

# 1 Introduction

Social choice theory is the study of collective decision processes and procedures. Many problems arise while aggregating the preferences of multi agent systems. Some of the most common issues are concerned about the criterion based on which the agents can select a suitable alternative. Another challenge is to aggregate many rankings into a single consensus ranking, reflecting an optimal compromise between various alternatives. For instance, the plurality rule may result in a completely different solution than a Condorcet winner. Therefore, for a finite alternative problem, it is necessary to map the preference relation to utility function and assign numerical values based on ranking. The Gibbard-Satterthwaite Theorem establishes the impossibility of devising a reasonable, general voting rule that is immune to strategic manipulation. Hence, it is important to devise an algorithm that takes into account the bribery involved in the voting process, strategic manipulation and is in agreement with the Arrow's Impossibility Theorem.

Parametrized Complexity is a branch of Theoretical Computer Science that aims to understand computational problems with respect to multiple input parameters, and to classify these problems according to their inherent difficulty. A parametrized problem is a problem with a certain feature of the input distinguished as the parameter. For example, for our election problems, the parameter will always be the number  $m$  of candidates in the election. A problem is fixed-parameter tractable (in FPT) if there exists an algorithm that, given an instance  $I$  with parameter  $k$ , can compute the answer to this problem in time  $f(k) \cdot |I|^{O(1)}$ , where  $f$  is a computable function and  $|I|$  is the length of the encoding of  $I$ . A voting rule  $R$  is a function that maps an election to a subset of alternatives, the set of winners. There are numerous voting rules in practice such as Borda's rule, plurality, anti-plurality and Condorcet extensions like Copeland, Maximin and Dodgson's rule that can be used to determine the score of an alternative.

This project basically aims to survey problems in the intersection area between computational social choice and parametrized complexity and formulate a conclusion on the query complexity of tournament solutions. This proposal includes the existing results and the road-map of questions to be explored.

# 2 Technical Preliminaries

We model a directed graph as a pair  $G = (V, E)$ , where  $V = \{1, 2, \dots, n\}$  is the set of vertices and  $E$  represents the set of edges  $(i, j)$  where  $i, j \in V$ . In order to obtain information about any property, we need to construct a decision tree at each node and has access to the results of queries so far, it further returns another query or returns an output. The complexity of a property is the worst-case number of queries that must be asked. Some terminologies regarding graphs and tournaments are briefly described as follows:

An evasive property of a digraph is a property such that one has to ask about all the edges in the graph  $(n(n-1))$  in the case of digraphs in order to test for

it. A tournament over  $N = \{1, \dots, n\}$  is a digraph with exactly one directed edge between every two vertices. In other words, a tournament is an orientation of a complete undirected graph or clique. In an election, the voters are usually assumed to hold linear preferences over the set of candidates. Candidate  $i \in N$  is said to dominate candidate  $j \in N$  if a majority of voters prefer  $i$  to  $j$ . This dominance relation is an irreflexive, asymmetric, and complete binary relation  $T$  on  $\{1, \dots, n\}$ ;  $i T j$  means that  $i$  dominates  $j$ . Such a binary relation on the set of candidates is known as a tournament. Crucially, this definition and the one given above are equivalent: there is a directed edge  $(i, j)$  in tournament  $T$  graph if and only if  $i T j$ . In the context of social choice theory, the vertices refer to candidates. In a tournament, a candidate that dominates every other candidate (that is, has an outdegree of  $(n - 1)$ ) is known as a Condorcet winner. By Pigeonhole Principle, a Condorcet winner, if exists, is unique.

In analyzing algorithms, we refer to the point in time when  $t$  queries have been submitted at time  $t$ . Given a tournament  $T$ ,  $T^{(t)}$  denotes the subset of the binary relation  $T$  based on the queries up to time  $t$ . In other words,  $T^{(t)}$  is a directed graph, with the set of vertices  $N$ , and a subset of size  $t$  of the edges of  $T$ . Informally,  $T^{(t)}$  is the part of the tournament  $T$  revealed to the algorithm at time  $t$ . At a certain time  $t$ , we say that  $i \in N$  is in if, based on the queries so far, the tournament can be completed in a way that  $i$  becomes a Condorcet winner; that is,  $i$  does not have any incoming edges in  $T^{(t)}$ . A candidate that is not in is out. Finally, we define two notions of score. The score of candidate  $i$  at time  $t$ , denoted  $s_i^t$ , is the number of candidates  $i$  beats based on the first  $t$  queries, i.e.  $i$ 's outdegree in  $T^{(t)}$ . The knockout score (k-score) of candidate  $i$  at time  $t$ , denoted  $\bar{s}_i^t$ , is the number of other candidates that  $i$  ousted; that is, the number of candidates  $j$  such that at some time  $t \leq t$  when  $j$  was in the algorithm queried  $i, j$ , and received the answer  $i T j$ .

Some of the known results are described briefly in the next section.

### 3 Overview of Known Results

The property of existence of a vertex with outdegree  $n - 1$  is evasive in digraphs. Also, it is known that  $2n - \lfloor \log(n) \rfloor - 2$  queries are sufficient to determine whether a Condorcet winner exists in a given tournament  $T$ . This is in contrast to the evasiveness of this property “Is there a candidate with outdegree  $n - 1$ ?” in general digraphs. The algorithm that is followed for determining the Condorcet winner is as follows:

The algorithm constructs a binary tree with  $n$  leaves that is almost complete, i.e. a tree of height  $D$ , such that for all levels  $d \leq D - 1$ , the number of leaves at depth  $d$  is  $2^d$ . The tree has to be full, that is each node has exactly 0 or 2 children. Each leaf of the tree is labeled by a distinct candidate  $i \in N$ . At each step, the algorithm submits the query  $i, j$ , where  $i$  and  $j$  are two sibling leaves in the tree. The father of  $i$  and  $j$  is labeled with the winner ( $i$  if  $i T j$  and  $j$  if  $j T i$ ), and the two leaves are trimmed (so the father becomes a leaf). This is continued until the tree becomes a singleton  $i_0$ . The algorithm

matches  $i_0$  against every candidate that was not matched against  $i_0$  previously. If  $i_0$  prevails in every competition, the algorithm returns true. If  $i_0$  loses some competition, the algorithm returns false.

Calculation of complexity, in this query model, of the problem statement “Given a tournament  $T$ , is there a Condorcet winner?” , i.e. determining a vertex with  $n - 1$  outdegree is  $2n - \lfloor \log(n) \rfloor - 2$  in contrast to the same property of general digraph which takes  $n(n - 1)$  time to compute.

## 4 Road-map of Questions

The algorithm specified above, however, does not cater to the practical problems that arise naturally in elections with incompletely specified votes, partially completed sports competitions, and more generally in any scenario where the outcome of some pairwise comparisons is not yet fully known. Therefore, there is a need to study the problem of computing possible and necessary winners for partially specified weighted and unweighted tournaments. Given any such solution concept  $S$ , possible winners of a partial tournament  $G$  are defined as alternatives that are selected by  $S$  in some completion of  $G$ , and necessary winners are alternatives that are selected in all completions. A completion means a (complete) tournament extending  $G$ . The research is aimed to explore solution concepts—including the uncovered set, Borda, ranked pairs, and maximin and calculate the computational complexity of identifying the possible and necessary winners for the above mentioned cases whose winner determination problem for tournaments is tractable.

While analysing the computational complexity of tournament solutions, it is observed that while some tournament solutions can be computed very efficiently (e.g., in linear time), others are shown to be NP-hard and thus computationally infeasible. Therefore, another objective of the research is to find a polynomial-time algorithm for computing the tournament solution if possible and to check if it can be computed in linear time and whether is it first-order definable. For tournaments for which a polynomial-time algorithm is not possible, I will try to show NP-hardness of the membership decision problem. The solution of tournament problems will, then, be applied to collective decision-making (social choice theory), adversarial decision-making (theory of zero-sum games), and coalitional decision-making (cooperative game theory).

## 5 References

1. Tournament Solutions: Extensions of Maximality and their Applications to Decision-Making by Felix Brandt
2. A note on the query complexity of the Condorcet winner problem by Ariel D. Procaccia
3. Possible and Necessary Winners of Partial Tournaments by Haris Aziz, Markus Brill, Felix Fischer, Paul Harrenstein, Jeerome Lang, and Hans Georg

Seedig

4. Stanford Encyclopedia of Philosophy : Social Choice Theory
5. Computational Social Choice by F. Brandt, V. Conitzer and U. Endris
6. Fixed-Parameter Tractability and Parameterized Complexity, Applied to Problems From Computational Social Choice by Claudia Lindner and Jorg Rothe
7. A quick summary of the field of Parameterized Complexity, 2012