6-1-2005

Communication Complexity of Common Voting Rules

Vincent Conitzer
Carnegie Mellon University, c@cmu.edu

Tuomas W. Sandholm
Carnegie Mellon University, sandholm@cs.cmu.edu

Communication Complexity of Common Voting Rules[*]

Vincent Conitzer
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
conitzer@cs.cmu.edu

Tuomas Sandholm
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
sandholm@cs.cmu.edu

ABSTRACT

We determine the communication complexity of the common voting rules. The rules (sorted by their communication complexity from low to high) are plurality, plurality with runoff, single transferable vote (STV), Condorcet, approval, Bucklin, cup, maximin, Borda, Copeland, and ranked pairs. For each rule, we first give a deterministic communication protocol and an upper bound on the number of bits communicated in it; then, we give a lower bound on (even the nondeterministic) communication requirements of the voting rule. The bounds match for all voting rules except STV and maximin.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Economics, Theory

Keywords

Communication Complexity, Voting

1. INTRODUCTION

One key factor in the practicality of any preference aggregation rule is its *communication burden.* To successfully aggregate the agents' preferences, it is usually not necessary for all the agents to report all of their preference information. Clever protocols that *elicit* the agents' preferences partially and sequentially have the potential to dramatically reduce the required communication. This has at least the following advantages:

- It can make preference aggregation feasible in settings where the total amount of preference information is too large to communicate.
- Even when communicating all the preference information is feasible, reducing the communication requirements lessens the burden placed on the agents. This is especially true when the agents, rather than knowing all their preferences in advance, need to invest effort (such as computation or information gathering) to determine their preferences [16].
- It preserves (some of) the agents' privacy.

Most of the work on reducing the communication burden in preference aggregation has focused on resource allocation settings such as *combinatorial auctions*, in which an auctioneer auctions off a number of (possibly distinct) items in a single event. Because in a combinatorial auction, bidders can have separate valuations for each of an exponential number of possible bundles of items, this is a setting in which reducing the communication burden is especially crucial. This can be accomplished by supplementing the auctioneer with an *elicitor* that incrementally elicits parts of the bidders' preferences on an as-needed basis, based on what the bidders have revealed about their preferences so far, as suggested by Conen and Sandholm [5]. For example, the elicitor can ask for a bidder's value for a specific bundle (*value queries*), which of two bundles the bidder prefers (*order queries*), which bundle he ranks kth or what the rank of a given bundle is (*rank queries*), which bundle he would purchase given a particular vector of prices (*demand queries*), *etc.*—until (at least) the final allocation can be determined. Experimentally, this yields drastic savings in preference revelation [11]. Furthermore, if the agents' valuation functions are drawn from certain natural subclasses, the elicitation problem can be solved using only polynomially many queries even in the worst case [23, 4, 13, 18, 14]. For a review of preference elicitation in combinatorial auctions, see [17]. Ascending combinatorial auctions are a well-known special form of preference elicitation, where the elicitor asks demand queries with increasing prices [15, 21, 1, 9]. Finally, resource

[*]This material is based upon work supported by the National Science Foundation under ITR grants IIS-0121678 and IIS-0427858, and a Sloan Fellowship. The authors thank Ilya Segal for helpful comments.

EC'05, June 5–8, 2005, Vancouver, British Columbia, Canada.
Copyright 2005 ACM 1-59593-049-3/05/0006 ...$5.00.

allocation problems have also been studied from a *communication complexity* viewpoint, thereby deriving lower bounds on the required communication. For example, Nisan and Segal show that exponential communication is required even to obtain a surplus greater than that obtained by auctioning off all objects as a single bundle [14]. Segal also studies social choice rules in general, and shows that for a large class of social choice rules, supporting *budget sets* must be revealed such that if every agent prefers the same outcome in her budget set, this proves the optimality of that outcome. Segal then uses this characterization to prove bounds on the communication required in resource allocation as well as matching settings [20].

In this paper, we will focus on the communication requirements of a generally applicable subclass of social choice rules, commonly known as *voting* rules. In a voting setting, there is a set of *candidate* outcomes over which the voters express their preferences by submitting a vote (typically, a ranking of the candidates), and the winner (that is, the chosen outcome) is determined based on these votes. The communication required by voting rules can be large either because the number of voters is large (such as, for example, in national elections), or because the number of candidates is large (for example, the agents can vote over allocations of a number of resources), or both. Prior work [8] has studied elicitation in voting, studying how computationally hard it is to decide whether a winner can be determined with the information elicited so far, as well as how hard it is to find the optimal sequence of queries given perfect suspicions about the voters' preferences. In addition, that paper discusses strategic (game-theoretic) issues introduced by elicitation.

In contrast, in this paper, we are concerned with the worst-case number of bits that must be communicated to execute a given voting rule, when nothing is known in advance about the voters' preferences. We determine the communication complexity of the common voting rules. For each rule, we first give an upper bound on the (deterministic) communication complexity by providing a communication protocol for it and analyzing how many bits need to be transmitted in this protocol. (Segal's results [20] do not apply to most voting rules because most voting rules are not intersection-monotonic (or even monotonic).[1]) For many of the voting rules under study, it turns out that one cannot do better than simply letting each voter immediately communicate all her (potentially relevant) information. However, for some rules (such as plurality with runoff, STV and cup) there is a straightforward multistage communication protocol that, with some analysis, can be shown to significantly outperform the immediate communication of all (potentially relevant) information. Finally, for some rules (such as the Condorcet and Bucklin rules), we need to introduce a more complex communication protocol to achieve the best possible upper bound. After obtaining the upper bounds, we show that they are tight by giving matching lower bounds on (even the nondeterministic) communication complexity of each voting rule. There are two exceptions: STV, for which our upper and lower bounds are apart by a factor $\log m$; and maximin, for which our best deterministic upper bound is also a factor $\log m$ above the (nondeterministic) lower bound, although we give a nondeterministic upper bound that matches the lower bound.

2. REVIEW OF VOTING RULES

In this section, we review the common voting rules that we study in this paper. A *voting rule*[2] is a function mapping a vector of the n voters' votes (*i.e.* preferences over candidates) to one of the m candidates (the winner) in the candidate set C. In some cases (such as the Condorcet rule), the rule may also declare that no winner exists. We do not concern ourselves with what happens in case of a tie between candidates (our lower bounds hold regardless of how ties are broken, and the communication protocols used for our upper bounds do not attempt to break the ties). All of the rules that we study are *rank-based* rules, which means that a vote is defined as an ordering of the candidates (with the exception of the plurality rule, for which a vote is a single candidate, and the approval rule, for which a vote is a subset of the candidates).

We will consider the following voting rules. (For rules that define a score, the candidate with the highest score wins.)

- *scoring rules.* Let $\vec{\alpha} = \langle \alpha_1, \dots, \alpha_m \rangle$ be a vector of integers such that $\alpha_1 \geq \alpha_2 \dots \geq \alpha_m$. For each voter, a candidate receives α_1 points if it is ranked first by the voter, α_2 if it is ranked second *etc.* The *score* $s_{\vec{\alpha}}$ of a candidate is the total number of points the candidate receives. The *Borda* rule is the scoring rule with $\vec{\alpha} = \langle m-1, m-2, \dots, 0 \rangle$. The *plurality* rule is the scoring rule with $\vec{\alpha} = \langle 1, 0, \dots, 0 \rangle$.

- *single transferable vote* (STV). The rule proceeds through a series of $m-1$ rounds. In each round, the candidate with the lowest plurality score (that is, the least number of voters ranking it first among the remaining candidates) is eliminated (and each of the votes for that candidate "transfer" to the next remaining candidate in the order given in that vote). The winner is the last remaining candidate.

- *plurality with run-off.* In this rule, a first round eliminates all candidates except the two with the highest plurality scores. Votes are transferred to these as in the STV rule, and a second round determines the winner from these two.

- *approval.* Each voter labels each candidate as either approved or disapproved. The candidate approved by the greatest number of voters wins.

- *Condorcet.* For any two candidates i and j, let $N(i, j)$ be the number of voters who prefer i to j. If there is a candidate i that is preferred to any other candidate by a majority of the voters (that is, $N(i, j) > N(j, i)$ for all $j \neq i$—that is, i wins every *pairwise election*), then candidate i wins.

[1] For two of the rules that we study that are intersection-monotonic, namely the approval and Condorcet rules, Segal's results can in fact be used to give alternative proofs of our lower bounds. We only give direct proofs for these rules here because 1) these direct proofs are among the easier ones in this paper, 2) the alternative proofs are nontrivial even given Segal's results, and 3) a space constraint applies. However, we hope to also include the alternative proofs in a later version.

[2] The term *voting protocol* is often used to describe the same concept, but we seek to draw a sharp distinction between the rule mapping preferences to outcomes, and the communication/elicitation protocol used to implement this rule.

- *maximin* (aka. *Simpson*). The *maximin score* of i is $s(i) = \min_{j \neq i} N(i,j)$—that is, i's worst performance in a pairwise election. The candidate with the highest maximin score wins.

- *Copeland*. For any two distinct candidates i and j, let $C(i,j) = 1$ if $N(i,j) > N(j,i)$, $C(i,j) = 1/2$ if $N(i,j) = N(j,i)$ and $C(i,j) = 0$ if $N(i,j) < N(j,i)$. The *Copeland score* of candidate i is $s(i) = \sum_{j \neq i} C(i,j)$.

- *cup* (sequential binary comparisons). The cup rule is defined by a balanced[3] binary tree T with one leaf per candidate, and an assignment of candidates to leaves (each leaf gets one candidate). Each non-leaf node is assigned the winner of the pairwise election of the node's children; the candidate assigned to the root wins.

- *Bucklin*. For any candidate i and integer l, let $B(i,l)$ be the number of voters that rank candidate i among the top l candidates. The winner is $\arg\min_i(\min\{l : B(i,l) > n/2\})$. That is, if we say that a voter "approves" her top l candidates, then we repeatedly increase l by 1 until some candidate is approved by more than half the voters, and this candidate is the winner.

- *ranked pairs*. This rule determines an order $\succ$ on all the candidates, and the winner is the candidate at the top of this order. Sort all ordered pairs of candidates (i,j) by $N(i,j)$, the number of voters who prefer i to j. Starting with the pair (i,j) with the highest $N(i,j)$, we "lock in" the result of their pairwise election ($i \succ j$). Then, we move to the next pair, and we lock the result of their pairwise election. We continue to lock every pairwise result that does not contradict the ordering $\succ$ established so far.

We emphasize that these definitions of voting rules do *not* concern themselves with how the votes are elicited from the voters; all the voting rules, including those that are suggestively defined in terms of "rounds", are in actuality just functions mapping the vector of all the voters' votes to a winner. Nevertheless, there are always many different ways of eliciting the votes (or the relevant parts thereof) from the voters. For example, in the plurality with runoff rule, one way of eliciting the votes is to ask every voter to declare her entire ordering of the candidates up front. Alternatively, we can first ask every voter to declare only her most preferred candidate; then, we will know the two candidates in the runoff, and we can ask every voter which of these two candidates she prefers. Thus, we distinguish between the *voting rule* (the mapping from vectors of votes to outcomes) and the *communication protocol* (which determines how the relevant parts of the votes are actually elicited from the voters). The goal of this paper is to give efficient communication protocols for the voting rules just defined, and to prove that there do not exist any more efficient communication protocols.

It is interesting to note that the choice of the communication protocol may affect the strategic behavior of the voters. Multistage communication protocols may reveal to the voters some information about how the other voters are voting (for example, in the two-stage communication protocol just given for plurality with runoff, in the second stage voters will know which two candidates have the highest plurality scores). In general, when the voters receive such information, it may give them incentives to vote differently than they would have in a single-stage communication protocol in which all voters declare their entire votes simultaneously. Of course, even the single-stage communication protocol is not strategy-proof[4] for any reasonable voting rule, by the Gibbard-Satterthwaite theorem [10, 19]. However, this does not mean that we should not be concerned about adding even more opportunities for strategic voting. In fact, many of the communication protocols introduced in this paper do introduce additional opportunities for strategic voting, but we do not have the space to discuss this here. (In prior work [8], we do give an example where an elicitation protocol for the approval voting rule introduces strategic voting, and give principles for designing elicitation protocols that do not introduce strategic problems.)

Now that we have reviewed voting rules, we move on to a brief review of communication complexity theory.

3. REVIEW OF SOME COMMUNICATION COMPLEXITY THEORY

In this section, we review the basic model of a communication problem and the lower-bounding technique of constructing a *fooling set*. (The basic model of a communication problem is due to Yao [22]; for an overview see Kushilevitz and Nisan [12].)

Each player $1 \leq i \leq n$ knows (only) input x_i. Together, they seek to compute $f(x_1, x_2, \ldots, x_n)$. In a *deterministic* protocol for computing f, in each stage, one of the players announces (to all other players) a bit of information based on her own input and the bits announced so far. Eventually, the communication terminates and all players know $f(x_1, x_2, \ldots, x_n)$. The goal is to minimize the worst-case (over all input vectors) number of bits sent. The deterministic communication complexity of a problem is the worst-case number of bits sent in the best (correct) deterministic protocol for it. In a *nondeterministic* protocol, the next bit to be sent can be chosen nondeterministically. For the purposes of this paper, we will consider a nondeterministic protocol correct if for every input vector, there is some sequence of nondeterministic choices the players can make so that the players know the value of f when the protocol terminates. The nondeterministic communication complexity of a problem is the worst-case number of bits sent in the best (correct) nondeterministic protocol for it.

We are now ready to give the definition of a fooling set.

Definition 1. *A* fooling set *is a set of input vectors* $\{(x_1^1, x_2^1, \ldots, x_n^1), (x_1^2, x_2^2, \ldots, x_n^2), \ldots, (x_1^k, x_2^k, \ldots, x_n^k)\}$ *such that for any* i, $f(x_1^i, x_2^i, \ldots, x_n^i) = f_0$ *for some constant* f_0, *but for any* $i \neq j$, *there exists some vector* $(r_1, r_2, \ldots, r_n) \in \{i,j\}^n$ *such that* $f(x_1^{r_1}, x_2^{r_2}, \ldots, x_n^{r_n}) \neq f_0$. *(That is, we can mix the inputs from the two input vectors to obtain a vector with a different function value.)*

It is known that if a fooling set of size k exists, then $\log k$ is a lower bound on the communication complexity (even the nondeterministic communication complexity) [12].

[3] Balanced means that the difference in depth between two leaves can be at most one.

[4] A strategy-proof protocol is one in which it is in the players' best interest to report their preferences truthfully.

For the purposes of this paper, f is the voting rule that maps the votes to the winning candidate, and x_i is voter i's vote (the information that the voting rule would require from the voter if there were no possibility of multistage communication—*i.e.* the most preferred candidate (plurality), the approved candidates (approval), or the ranking of all the candidates (all other protocols)). However, when we derive our lower bounds, f will only signify whether a distinguished candidate a wins. (That is, f is 1 if a wins, and 0 otherwise.) This will strengthen our lower bound results (because it implies that even finding out whether one given candidate wins is "hard").[5] Thus, a fooling set in our context is a set of vectors of votes so that a wins (does not win) with each of them; but for any two different vote vectors in the set, there is a way of taking some voters' votes from the first vector and the others' votes from the second vector, so that a does not win (wins).

To simplify the proofs of our lower bounds, we make assumptions such as "the number of voters n is odd" in many of these proofs. Therefore, technically, we do not prove the lower bound for (number of candidates, number of voters) pairs (m, n) that do not satisfy these assumptions (for example, if we make the above assumption, then we technically do not prove the lower bound for any pair (m, n) in which n is even). Nevertheless, we always prove the lower bound for a representative set of (m, n) pairs. For example, for every one of our lower bounds it is the case that for infinitely many values of m, there are infinitely many values of n such that the lower bound is proved for the pair (m, n).

4. RESULTS

We are now ready to present our results. For each voting rule, we first give a deterministic communication protocol for determining the winner to establish an upper bound. Then, we give a lower bound on the nondeterministic communication complexity (even on the complexity of deciding whether a given candidate wins, which is an easier question). The lower bounds match the upper bounds in all but two cases: the STV rule (upper bound $O(n(\log m)^2)$; lower bound $\Omega(n \log m)$) and the maximin rule (upper bound $O(nm \log m)$, although we do give a *nondeterministic* protocol that is $O(nm)$; lower bound $\Omega(nm)$).

When we discuss a voting rule in which the voters rank the candidates, we will represent a ranking in which candidate c_1 is ranked first, c_2 is ranked second, *etc.* as $c_1 \succ c_2 \succ \ldots c_m$. Sometimes for the purposes of a proof the internal ranking of a subset of the candidates does not matter, and in this case we will not specify it. For example, if $S = \{c_2, c_3\}$, then $c_1 \succ S \succ c_4$ indicates that either the ranking $c_1 \succ c_2 \succ c_3 \succ c_4$ or the ranking $c_1 \succ c_3 \succ c_2 \succ c_4$ can be used for the proof.

We first give a universal upper bound.

THEOREM 1. *The deterministic communication complexity of any rank-based voting rule is $O(nm \log m)$.*

PROOF. This bound is achieved by simply having everyone communicate their entire ordering of the candidates (indicating the rank of an individual candidate requires only $O(\log m)$ bits, so each of the n voters can simply indicate the rank of each of the m candidates). □

The next lemma will be useful in a few of our proofs.

LEMMA 1. *If m divides n, then $\log(n!) - m\log((n/m)!) \geq n(\log m - 1)/2$.*

PROOF. If $n/m = 1$ (that is, $n = m$), then this expression simplifies to $\log(n!)$. We have $\log(n!) = \sum_{i=1}^{n} \log i \geq \int_{x=1}^{n} \log(i)dx$, which, using integration by parts, is equal to $n \log n - (n-1) > n(\log n - 1) = n(\log m - 1) > n(\log m - 1)/2$. So, we can assume that $n/m \geq 2$. We observe that $\log(n!) = \sum_{i=1}^{n} \log i = \sum_{i=0}^{n/m-1} \sum_{j=1}^{m} \log(im+j) \geq \sum_{i=1}^{n/m-1} \sum_{j=1}^{m} \log(im)$ $= m \sum_{i=1}^{n/m-1} \log(im)$, and that $m\log((n/m)!) = m \sum_{i=1}^{n/m} \log(i)$. Therefore, $\log(n!) - m\log((n/m)!) \geq m \sum_{i=1}^{n/m-1} \log(im) - m \sum_{i=1}^{n/m} \log(i) = m((\sum_{i=1}^{n/m-1} \log(im/i)) - \log(n/m)) = m((n/m - 1)\log m - \log n + \log m) = n\log m - m\log n$. Now, using the fact that $n/m \geq 2$, we have $m \log n = n(m/n)\log m(n/m) = n(m/n)(\log m + \log(n/m)) \leq n(1/2)(\log m + \log 2)$. Thus, $\log(n!) - m\log((n/m)!) \geq n\log m - m\log n \geq n\log m - n(1/2)(\log m + \log 2) = n(\log m - 1)/2$. □

THEOREM 2. *The deterministic communication complexity of the plurality rule is $O(n \log m)$.*

PROOF. Indicating one of the candidates requires only $O(\log m)$ bits, so each voter can simply indicate her most preferred candidate. □

THEOREM 3. *The nondeterministic communication complexity of the plurality rule is $\Omega(n \log m)$ (even to decide whether a given candidate a wins).*

PROOF. We will exhibit a fooling set of size $\frac{n'!}{((\frac{n'}{m})!)^m}$ where $n' = (n-1)/2$. Taking the logarithm of this gives $\log(n'!) - m\log((n'/m)!)$, so the result follows from Lemma 1. The fooling set will consist of all vectors of votes satisfying the following constraints:

- For any $1 \leq i \leq n'$, voters $2i - 1$ and $2i$ vote the same.

[5] One possible concern is that in the case where ties are possible, it may require much communication to verify whether a specific candidate a is among the winners, but little communication to produce *one* of the winners. However, all the fooling sets we use in the proofs have the property that if a wins, then a is the unique winner. Therefore, in these fooling sets, if one knows any one of the winners, then one knows whether a is a winner. Thus, computing one of the winners requires at least as much communication as verifying whether a is among the winners. In general, when a communication problem allows multiple correct answers for a given vector of inputs, this is known as computing a *relation* rather than a function [12]. However, as per the above, we can restrict our attention to a subset of the domain where the voting rule truly is a (single-valued) function, and hence lower bounding techniques for functions rather than relations will suffice.

- Every candidate receives equally many votes from the first $2n' = n-1$ voters.
- The last voter (voter n) votes for a.

Candidate a wins with each one of these vote vectors because of the extra vote for a from the last voter. Given that m divides n', let us see how many vote vectors there are in the fooling set. We need to distribute n' voter pairs evenly over m candidates, for a total of n'/m voter pairs per candidate; and there are precisely $\frac{n'!}{((\frac{n'}{m})!)^m}$ ways of doing this.[6] All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let i be a number such that the two vote vectors disagree on the candidate for which voters $2i-1$ and $2i$ vote. Without loss of generality, suppose that in the first vote vector, these voters do not vote for a (but for some other candidate, b, instead). Now, construct a new vote vector by taking votes $2i-1$ and $2i$ from the first vote vector, and the remaining votes from the second vote vector. Then, b receives $2n'/m+2$ votes in this newly constructed vote vector, whereas a receives at most $2n'/m+1$ votes. So, a is not the winner in the newly constructed vote vector, and hence we have a correct fooling set. □

Theorem 4. *The deterministic communication complexity of the plurality with runoff rule is $O(n \log m)$.*

Proof. First, let every voter indicate her most preferred candidate using $\log m$ bits. After this, the two candidates in the runoff are known, and each voter can indicate which one she prefers using a single additional bit. □

Theorem 5. *The nondeterministic communication complexity of the plurality with runoff rule is $\Omega(n \log m)$ (even to decide whether a given candidate a wins).*

Proof. We will exhibit a fooling set of size $\frac{n'!}{((\frac{n'}{m'})!)^{m'}}$ where $m' = m/2$ and $n' = (n-2)/4$. Taking the logarithm of this gives $\log(n'!) - m' \log((n'/m')!)$, so the result follows from Lemma 1. Divide the candidates into m' pairs: $(c_1, d_1), (c_2, d_2), \dots, (c_{m'}, d_{m'})$ where $c_1 = a$ and $d_1 = b$. The fooling set will consist of all vectors of votes satisfying the following constraints:

- For any $1 \leq i \leq n'$, voters $4i-3$ and $4i-2$ rank the candidates $c_{k(i)} \succ a \succ C - \{a, c_{k(i)}\}$, for some candidate $c_{k(i)}$. (If $c_{k(i)} = a$ then the vote is simply $a \succ C - \{a\}$.)
- For any $1 \leq i \leq n'$, voters $4i-1$ and $4i$ rank the candidates $d_{k(i)} \succ a \succ C - \{a, d_{k(i)}\}$ (that is, their most preferred candidate is the candidate that is paired with the candidate that the previous two voters vote for).
- Every candidate is ranked at the top of equally many of the first $4n' = n-2$ votes.
- Voter $4n'+1 = n-1$ ranks the candidates $a \succ C - \{a\}$.
- Voter $4n'+2 = n$ ranks the candidates $b \succ C - \{b\}$.

Candidate a wins with each one of these vote vectors: because of the last two votes, candidates a and b are one vote ahead of all the other candidates and continue to the runoff, and at this point all the votes that had another candidate ranked at the top transfer to a, so that a wins the runoff. Given that m' divides n', let us see how many vote vectors there are in the fooling set. We need to distribute n' groups of four voters evenly over the m' pairs of candidates, and (as in the proof of Theorem 3) there are $\frac{n'!}{((\frac{n'}{m'})!)^{m'}}$ ways of doing this. All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let i be a number such that $c_{k(i)}$ is not the same in both of these two vote vectors, that is, $c^1_{k(i)}$ ($c_{k(i)}$ in the first vote vector) is not equal to $c^2_{k(i)}$ ($c_{k(i)}$ in the second vote vector). Without loss of generality, suppose $c^1_{k(i)} \neq a$. Now, construct a new vote vector by taking votes $4i-3, 4i-2, 4i-1, 4i$ from the first vote vector, and the remaining votes from the second vote vector. In this newly constructed vote vector, $c^1_{k(i)}$ and $d^1_{k(i)}$ each receive $4n'/m+2$ votes in the first round, whereas a receives at most $4n'/m+1$ votes. So, a does not continue to the runoff in the newly constructed vote vector, and hence we have a correct fooling set. □

Theorem 6. *The nondeterministic communication complexity of the Borda rule is $\Omega(nm \log m)$ (even to decide whether a given candidate a wins).*

Proof. We will exhibit a fooling set of size $(m'!)^{n'}$ where $m' = m-2$ and $n' = (n-2)/4$. This will prove the theorem because $m'!$ is $\Omega(m \log m)$, so that $\log((m'!)^{n'}) = n' \log(m'!)$ is $\Omega(nm \log m)$. For every vector $(\pi_1, \pi_2, \dots, \pi_{n'})$ consisting of n' orderings of all candidates other than a and another fixed candidate b (technically, the orderings take the form of a one-to-one function $\pi_i : \{1, 2, \dots, m'\} \to C - \{a, b\}$ with $\pi_i(j) = c$ indicating that candidate c is the jth in the order represented by π_i), let the following vector of votes be an element of the fooling set:

- For $1 \leq i \leq n'$, let voters $4i-3$ and $4i-2$ rank the candidates $a \succ b \succ \pi_i(1) \succ \pi_i(2) \succ \dots \succ \pi_i(m')$.
- For $1 \leq i \leq n'$, let voters $4i-1$ and $4i$ rank the candidates $\pi_i(m') \succ \pi_i(m'-1) \succ \dots \succ \pi_i(1) \succ b \succ a$.
- Let voter $4n'+1 = n-1$ rank the candidates $a \succ b \succ \pi_0(1) \succ \pi_0(2) \succ \dots \succ \pi_0(m')$ (where π_0 is an arbitrary order of the candidates other than a and b which is the same for every element of the fooling set).
- Let voter $4n'+2 = n$ rank the candidates $\pi_0(m') \succ \pi_0(m'-1) \succ \dots \succ \pi_0(1) \succ a \succ b$.

We observe that this fooling set has size $(m'!)^{n'}$, and that candidate a wins in each vector of votes in the fooling set (to

[6] An intuitive proof of this is the following. We can count the number of permutations of n' elements as follows. First, divide the elements into m buckets of size n'/m, so that if x is placed in a lower-indexed bucket than y, then x will be indexed lower in the eventual permutation. Then, decide on the permutation within each bucket (for which there are $(n'/m)!$ choices per bucket). It follows that $n'!$ equals the number of ways to divide n' elements into m buckets of size n'/m, times $((n'/m)!)^m$.

see why, we observe that for any $1 \leq i \leq n'$, votes $4i-3$ and $4i-2$ rank the candidates in the exact opposite way from votes $4i-1$ and $4i$, which under the Borda rule means they cancel out; and the last two votes give one more point to a than to any other candidate—besides b who gets two fewer points than a). All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let the first vote vector correspond to the vector $(\pi_1^1, \pi_2^1, \ldots, \pi_{n'}^1)$, and let the second vote vector correspond to the vector $(\pi_1^2, \pi_2^2, \ldots, \pi_{n'}^2)$. For some i, we must have $\pi_i^1 \neq \pi_i^2$, so that for some candidate $c \notin \{a, b\}$, $(\pi_i^1)^{-1}(c) < (\pi_i^2)^{-1}(c)$ (that is, c is ranked higher in π_i^1 than in π_i^2). Now, construct a new vote vector by taking votes $4i-3$ and $4i-2$ from the first vote vector, and the remaining votes from the second vote vector. a's Borda score remains unchanged. However, because c is ranked higher in π_i^1 than in π_i^2, c receives at least 2 more points from votes $4i-3$ and $4i-2$ in the newly constructed vote vector than it did in the second vote vector. It follows that c has a higher Borda score than a in the newly constructed vote vector. So, a is not the winner in the newly constructed vote vector, and hence we have a correct fooling set. □

THEOREM 7. *The nondeterministic communication complexity of the Copeland rule is $\Omega(nm \log m)$ (even to decide whether a given candidate a wins).*

PROOF. We will exhibit a fooling set of size $(m'!)^{n'}$ where $m' = (m-2)/2$ and $n' = (n-2)/2$. This will prove the theorem because $m'!$ is $\Omega(m \log m)$, so that $\log((m'!)^{n'}) = n' \log(m'!)$ is $\Omega(nm \log m)$. We write the set of candidates as the following disjoint union: $C = \{a, b\} \cup L \cup R$ where $L = \{l_1, l_2, \ldots, l_{m'}\}$ and $R = \{r_1, r_2, \ldots, r_{m'}\}$. For every vector $(\pi_1, \pi_2, \ldots, \pi_{n'})$ consisting of n' permutations of the integers 1 through m' ($\pi_i : \{1, 2, \ldots, m'\} \rightarrow \{1, 2, \ldots, m'\}$), let the following vector of votes be an element of the fooling set:

- For $1 \leq i \leq n'$, let voter $2i-1$ rank the candidates $a \succ b \succ l_{\pi_i(1)} \succ r_{\pi_i(1)} \succ l_{\pi_i(2)} \succ r_{\pi_i(2)} \succ \ldots \succ l_{\pi_i(m')} \succ r_{\pi_i(m')}$.
- For $1 \leq i \leq n'$, let voter $2i$ rank the candidates $r_{\pi_i(m')} \succ l_{\pi_i(m')} \succ r_{\pi_i(m'-1)} \succ l_{\pi_i(m'-1)} \succ \ldots \succ r_{\pi_i(1)} \succ l_{\pi_i(1)} \succ b \succ a$.
- Let voter $n-1 = 2n'+1$ rank the candidates $a \succ b \succ l_1 \succ r_1 \succ l_2 \succ r_2 \succ \ldots \succ l_{m'} \succ r_{m'}$.
- Let voter $n = 2n'+2$ rank the candidates $r_{m'} \succ l_{m'} \succ r_{m'-1} \succ l_{m'-1} \succ \ldots \succ r_1 \succ l_1 \succ a \succ b$.

We observe that this fooling set has size $(m'!)^{n'}$, and that candidate a wins in each vector of votes in the fooling set (every pair of candidates is tied in their pairwise election, with the exception that a defeats b, so that a wins the election by half a point). All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let the first vote vector correspond to the vector $(\pi_1^1, \pi_2^1, \ldots, \pi_{n'}^1)$, and let the second vote vector correspond to the vector $(\pi_1^2, \pi_2^2, \ldots, \pi_{n'}^2)$. For some i, we must have $\pi_i^1 \neq \pi_i^2$, so that for some $j \in \{1, 2, \ldots, m'\}$, we have $(\pi_i^1)^{-1}(j) < (\pi_i^2)^{-1}(j)$. Now, construct a new vote vector by taking vote $2i-1$ from the first vote vector, and the remaining votes from the second vote vector. a's Copeland score remains unchanged. Let us consider the score of l_j. We first observe that the rank of l_j in vote $2i-1$ in the newly constructed vote vector is at least 2 higher than it was in the second vote vector, because $(\pi_i^1)^{-1}(j) < (\pi_i^2)^{-1}(j)$. Let $D^1(l_j)$ be the set of candidates in $L \cup R$ that voter $2i-1$ ranked lower than l_j in the first vote vector ($D^1(l_j) = \{c \in L \cup R : l_j \succ_{2i-1}^1 c\}$), and let $D^2(l_j)$ be the set of candidates in $L \cup R$ that voter $2i-1$ ranked lower than l_j in the second vote vector ($D^2(l_j) = \{c \in L \cup R : l_j \succ_{2i-1}^2 c\}$). Then, it follows that in the newly constructed vote vector, l_j defeats all the candidates in $D^1(l_j) - D^2(l_j)$ in their pairwise elections (because l_j receives an "extra" vote in each one of these pairwise elections relative to the second vote vector), and loses to all the candidates in $D^2(l_j) - D^1(l_j)$ (because l_j loses a vote in each one of these pairwise elections relative to the second vote vector), and ties with everyone else. But $|D^1(l_j)| - |D^2(l_j)| \geq 2$, and hence $|D^1(l_j) - D^2(l_j)| - |D^2(l_j) - D^1(l_j)| \geq 2$. Hence, in the newly constructed vote vector, l_j has at least two more pairwise wins than pairwise losses, and therefore has at least 1 more point than if l_j had tied all its pairwise elections. Thus, l_j has a higher Copeland score than a in the newly constructed vote vector. So, a is not the winner in the newly constructed vote vector, and hence we have a correct fooling set. □

THEOREM 8. *The* nondeterministic *communication complexity of the maximin rule is $O(nm)$.*

PROOF. The nondeterministic protocol will guess which candidate w is the winner, and, for each other candidate c, which candidate $o(c)$ is the candidate against whom c receives its lowest score in a pairwise election. Then, let every voter communicate the following:

- for each candidate $c \neq w$, whether she prefers c to w;
- for each candidate $c \neq w$, whether she prefers c to $o(c)$.

We observe that this requires the communication of $2n(m-1)$ bits. If the guesses were correct, then, letting $N(d, e)$ be the number of voters preferring candidate d to candidate e, we should have $N(c, o(c)) < N(w, c')$ for any $c \neq w, c' \neq w$, which will prove that w wins the election. □

THEOREM 9. *The nondeterministic communication complexity of the maximin rule is $\Omega(nm)$ (even to decide whether a given candidate a wins).*

PROOF. We will exhibit a fooling set of size $2^{n'm'}$ where $m' = m-2$ and $n' = (n-1)/4$. Let b be a candidate other than a. For every vector $(S_1, S_2, \ldots, S_{n'})$ consisting of n' subsets $S_i \subseteq C - \{a, b\}$, let the following vector of votes be an element of the fooling set:

- For $1 \leq i \leq n'$, let voters $4i-3$ and $4i-2$ rank the candidates $S_i \succ a \succ C - (S_i \cup \{a, b\}) \succ b$.
- For $1 \leq i \leq n'$, let voters $4i-1$ and $4i$ rank the candidates $b \succ C - (S_i \cup \{a, b\}) \succ a \succ S_i$.

- Let voter $4n' + 1 = n$ rank the candidates $a \succ b \succ C - \{a, b\}$.

We observe that this fooling set has size $(2^{m'})^{n'} = 2^{n'm'}$, and that candidate a wins in each vector of votes in the fooling set (in every one of a's pairwise elections, a is ranked higher than its opponent by $2n'+1 = (n+1)/2 > n/2$ votes). All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let the first vote vector correspond to the vector $(S^1_1, S^1_2, \dots, S^1_{n'})$, and let the second vote vector correspond to the vector $(S^2_1, S^2_2, \dots, S^2_{n'})$. For some i, we must have $S^1_i \neq S^2_i$, so that either $S^1_i \not\subseteq S^2_i$ or $S^2_i \not\subseteq S^1_i$. Without loss of generality, suppose $S^1_i \not\subseteq S^2_i$, and let c be some candidate in $S^1_i - S^2_i$. Now, construct a new vote vector by taking votes $4i - 3$ and $4i - 2$ from the first vote vector, and the remaining votes from the second vote vector. In this newly constructed vote vector, a is ranked higher than c by only $2n'-1$ voters, for the following reason. Whereas voters $4i-3$ and $4i - 2$ do not rank c higher than a in the second vote vector (because $c \notin S^2_i$), voters $4i - 3$ and $4i - 2$ do rank c higher than a in the first vote vector (because $c \in S^1_i$). Moreover, in every one of b's pairwise elections, b is ranked higher than its opponent by at least $2n'$ voters. So, a has a lower maximin score than b, therefore a is not the winner in the newly constructed vote vector, and hence we have a correct fooling set. □

THEOREM 10. *The deterministic communication complexity of the STV rule is $O(n(\log m)^2)$.*

PROOF. Consider the following communication protocol. Let each voter first announce her most preferred candidate ($O(n \log m)$ communication). In the remaining rounds, we will keep track of each voter's most preferred candidate among the remaining candidates, which will be enough to implement the rule. When candidate c is eliminated, let each of the voters whose most preferred candidate among the remaining candidates was c announce their most preferred candidate among the candidates remaining after c's elimination. If candidate c was the ith candidate to be eliminated (that is, there were $m - i + 1$ candidates remaining before c's elimination), it follows that at most $n/(m - i + 1)$ voters had candidate c as their most preferred candidate among the remaining candidates, and thus the number of bits to be communicated after the elimination of the ith candidate is $O((n/(m-i+1)) \log m)$.[7] Thus, the total communication in this communication protocol is $O(n \log m + \sum_{i=1}^{m-1} (n/(m-i+1)) \log m)$. Of course, $\sum_{i=1}^{m-1} 1/(m - i + 1) = \sum_{i=2}^{m} 1/i$, which is $O(\log m)$. Substituting into the previous expression, we find that the communication complexity is $O(n(\log m)^2)$. □

THEOREM 11. *The nondeterministic communication complexity of the STV rule is $\Omega(n \log m)$ (even to decide whether a given candidate a wins).*

PROOF. We omit this proof because of space constraint. □

THEOREM 12. *The deterministic communication complexity of the approval rule is $O(nm)$.*

PROOF. Approving or disapproving of a candidate requires only one bit of information, so every voter can simply approve or disapprove of every candidate for a total communication of nm bits. □

THEOREM 13. *The nondeterministic communication complexity of the approval rule is $\Omega(nm)$ (even to decide whether a given candidate a wins).*

PROOF. We will exhibit a fooling set of size $2^{n'm'}$ where $m' = m - 1$ and $n' = (n - 1)/4$. For every vector $(S_1, S_2, \dots, S_{n'})$ consisting of n' subsets $S_i \subseteq C - \{a\}$, let the following vector of votes be an element of the fooling set:

- For $1 \leq i \leq n'$, let voters $4i - 3$ and $4i - 2$ approve $S_i \cup \{a\}$.
- For $1 \leq i \leq n'$, let voters $4i - 1$ and $4i$ approve $C - (S_i \cup \{a\})$.
- Let voter $4n' + 1 = n$ approve $\{a\}$.

We observe that this fooling set has size $(2^{m'})^{n'} = 2^{n'm'}$, and that candidate a wins in each vector of votes in the fooling set (a is approved by $2n' + 1$ voters, whereas each other candidate is approved by only $2n'$ voters). All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let the first vote vector correspond to the vector $(S^1_1, S^1_2, \dots, S^1_{n'})$, and let the second vote vector correspond to the vector $(S^2_1, S^2_2, \dots, S^2_{n'})$. For some i, we must have $S^1_i \neq S^2_i$, so that either $S^1_i \not\subseteq S^2_i$ or $S^2_i \not\subseteq S^1_i$. Without loss of generality, suppose $S^1_i \not\subseteq S^2_i$, and let b be some candidate in $S^1_i - S^2_i$. Now, construct a new vote vector by taking votes $4i - 3$ and $4i - 2$ from the first vote vector, and the remaining votes from the second vote vector. In this newly constructed vote vector, a is still approved by $2n' + 1$ votes. However, b is approved by $2n' + 2$ votes, for the following reason. Whereas voters $4i-3$ and $4i-2$ do not approve b in the second vote vector (because $b \notin S^2_i$), voters $4i - 3$ and $4i-2$ do approve b in the first vote vector (because $b \in S^1_i$). It follows that b's score in the newly constructed vote vector is b's score in the second vote vector ($2n'$), plus two. So, a is not the winner in the newly constructed vote vector, and hence we have a correct fooling set. □

Interestingly, an $\Omega(m)$ lower bound can be obtained even for the problem of finding a candidate that is approved by *more than one voter* [20].

THEOREM 14. *The deterministic communication complexity of the Condorcet rule is $O(nm)$.*

PROOF. We maintain a set of active candidates S which is initialized to C. At each stage, we choose two of the active candidates (say, the two candidates with the lowest indices), and we let each voter communicate which of the two candidates she prefers. (Such a stage requires the communication of n bits, one per voter.) The candidate preferred by fewer

[7] Actually, $O((n/(m - i + 1)) \log(m - i + 1))$ is also correct, but it will not improve the bound.

voters (the loser of the pairwise election) is removed from S. (If the pairwise election is tied, both candidates are removed.) After at most $m-1$ iterations, only one candidate is left (or zero candidates are left, in which case there is no Condorcet winner). Let a be the remaining candidate. To find out whether candidate a is the Condorcet winner, let each voter communicate, for every candidate $c \neq a$, whether she prefers a to c. (This requires the communication of at most $n(m-1)$ bits.) This is enough to establish whether a won each of its pairwise elections (and thus, whether a is the Condorcet winner). □

Theorem 15. *The nondeterministic communication complexity of the Condorcet rule is $\Omega(nm)$ (even to decide whether a given candidate a wins).*

Proof. We will exhibit a fooling set of size $2^{n'm'}$ where $m' = m-1$ and $n' = (n-1)/2$. For every vector $(S_1, S_2, \ldots, S_{n'})$ consisting of n' subsets $S_i \subseteq C - \{a\}$, let the following vector of votes be an element of the fooling set:

- For $1 \leq i \leq n'$, let voter $2i-1$ rank the candidates $S_i \succ a \succ C - S_i$.
- For $1 \leq i \leq n'$, let voter $2i$ rank the candidates $C - S_i \succ a \succ S_i$.
- Let voter $2n'+1 = n$ rank the candidates $a \succ C - \{a\}$.

We observe that this fooling set has size $(2^{m'})^{n'} = 2^{n'm'}$, and that candidate a wins in each vector of votes in the fooling set (a wins each of its pairwise elections by a single vote). All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let the first vote vector correspond to the vector $(S_1^1, S_2^1, \ldots, S_{n'}^1)$, and let the second vote vector correspond to the vector $(S_1^2, S_2^2, \ldots, S_{n'}^2)$. For some i, we must have $S_i^1 \neq S_i^2$, so that either $S_i^1 \not\subseteq S_i^2$ or $S_i^2 \not\subseteq S_i^1$. Without loss of generality, suppose $S_i^1 \not\subseteq S_i^2$, and let b be some candidate in $S_i^1 - S_i^2$. Now, construct a new vote vector by taking vote $2i-1$ from the first vote vector, and the remaining votes from the second vote vector. In this newly constructed vote vector, b wins its pairwise election against a by one vote (vote $2i-1$ ranks b above a in the newly constructed vote vector because $b \in S_i^1$, whereas in the second vote vector vote $2i-1$ ranked a above b because $b \notin S_i^2$). So, a is not the Condorcet winner in the newly constructed vote vector, and hence we have a correct fooling set. □

Theorem 16. *The deterministic communication complexity of the cup rule is $O(nm)$.*

Proof. Consider the following simple communication protocol. First, let all the voters communicate, for every one of the matchups in the first round, which of its two candidates they prefer. After this, the matchups for the second round are known, so let all the voters communicate which candidate they prefer in each matchup in the second round–*etc.* Because communicating which of two candidates is preferred requires only one bit per voter, and because there are only $m-1$ matchups in total, this communication protocol requires $O(nm)$ communication. □

Theorem 17. *The nondeterministic communication complexity of the cup rule is $\Omega(nm)$ (even to decide whether a given candidate a wins).*

Proof. We will exhibit a fooling set of size $2^{n'm'}$ where $m' = (m-1)/2$ and $n' = (n-7)/2$. Given that $m+1$ is a power of 2, so that one candidate gets a bye (that is, does not face an opponent) in the first round, let a be the candidate with the bye. Of the m' first-round matchups, let l_j denote the one ("left") candidate in the jth matchup, and let r_j be the other ("right") candidate. Let $L = \{l_j : 1 \leq j \leq m'\}$ and $R = \{r_j : 1 \leq j \leq m'\}$, so that $C = L \cup R \cup \{a\}$.

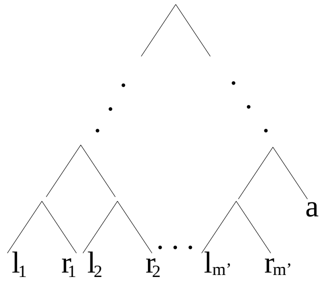

Figure 1: The schedule for the cup rule used in the proof of Theorem 17.

For every vector $(S_1, S_2, \ldots, S_{n'})$ consisting of n' subsets $S_i \subseteq R$, let the following vector of votes be an element of the fooling set:

- For $1 \leq i \leq n'$, let voter $2i-1$ rank the candidates $S_i \succ L \succ a \succ R - S_i$.
- For $1 \leq i \leq n'$, let voter $2i$ rank the candidates $R - S_i \succ L \succ a \succ S_i$.
- Let voters $2n'+1 = n-6, 2n'+2 = n-5, 2n'+3 = n-4$ rank the candidates $L \succ a \succ R$.
- Let voters $2n'+4 = n-3, 2n'+5 = n-2$ rank the candidates $a \succ r_1 \succ l_1 \succ r_2 \succ l_2 \succ \ldots \succ r_{m'} \succ l_{m'}$.
- Let voters $2n'+6 = n-1, 2n'+7 = n$ rank the candidates $r_{m'} \succ l_{m'} \succ r_{m'-1} \succ l_{m'-1} \succ \ldots \succ r_1 \succ l_1 \succ a$.

We observe that this fooling set has size $(2^{m'})^{n'} = 2^{n'm'}$. Also, candidate a wins in each vector of votes in the fooling set, for the following reasons. Each candidate r_j defeats its opponent l_j in the first round. (For any $1 \leq i \leq n'$, the net effect of votes $2i-1$ and $2i$ on the pairwise election between r_j and l_j is zero; votes $n-6, n-5, n-4$ prefer l_j to r_j, but votes $n-3, n-2, n-1, n$ all prefer r_j to l_j.) Moreover, a defeats every r_j in their pairwise election. (For any $1 \leq i \leq n'$, the net effect of votes $2i-1$ and $2i$ on the pairwise election between a and r_j is zero; votes $n-1, n$ prefer r_j to a, but votes $n-6, n-5, n-4, n-3, n-2$ all prefer a to r_j.) It follows that a will defeat all the candidates that it faces.

All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let the first vote vector correspond to the vector

$(S_1^1, S_2^1, \ldots, S_{n'}^1)$, and let the second vote vector correspond to the vector $(S_1^2, S_2^2, \ldots, S_{n'}^2)$. For some i, we must have $S_i^1 \neq S_i^2$, so that either $S_i^1 \nsubseteq S_i^2$ or $S_i^2 \nsubseteq S_i^1$. Without loss of generality, suppose $S_i^1 \nsubseteq S_i^2$, and let r_j be some candidate in $S_i^1 - S_i^2$. Now, construct a new vote vector by taking vote $2i$ from the first vote vector, and the remaining votes from the second vote vector. We note that, whereas in the second vote vector vote $2i$ preferred r_j to l_j (because $r_j \in R - S_i^2$), in the newly constructed vote vector this is no longer the case (because $r_j \in S_i^1$). It follows that, whereas in the second vote vector, r_j defeated l_j in the first round by one vote, in the newly constructed vote vector, l_j defeats r_j in the first round. Thus, at least one l_j advances to the second round after defeating its opponent r_j. Now, we observe that in the newly constructed vote vector, any l_k wins its pairwise election against any r_q with $q \neq k$. This is because among the first $2n'$ votes, at least $n' - 1$ prefer l_k to r_q; votes $n-6, n-5, n-4$ prefer l_k to r_q; and, because $q \neq k$, either votes $n-3, n-2$ prefer l_k to r_q (if $k < q$), or votes $n-1, n$ prefer l_k to r_q (if $k > q$). Thus, at least $n' + 4 = (n+1)/2 > n/2$ votes prefer l_k to r_q. Moreover, any l_k wins its pairwise election against a. This is because only votes $n-3$ and $n-2$ prefer a to l_k. It follows that, after the first round, any surviving candidate l_k can only lose a matchup against another surviving $l_{k'}$, so that one of the l_k must win the election. So, a is not the winner in the newly constructed vote vector, and hence we have a correct fooling set. □

Theorem 18. *The deterministic communication complexity of the Bucklin rule is $O(nm)$.*

Proof. Let l be the minimum integer for which there is a candidate who is ranked among the top l candidates by more than half the votes. We will do a binary search for l. At each point, we will have a lower bound l_L which is smaller than l (initialized to 0), and an upper bound l_H which is at least l (initialized to m). While $l_H - l_L > 1$, we continue by finding out whether $\lfloor (l_H - l)/2 \rfloor$ is smaller than l, after which we can update the bounds.

To find out whether a number k is smaller than l, we determine every voter's k most preferred candidates. Every voter can communicate which candidates are among her k most preferred candidates using m bits (for each candidate, indicate whether the candidate is among the top k or not), but because the binary search requires $\log m$ iterations, this gives us an upper bound of $O((\log m)nm)$, which is not strong enough. However, if $l_L < k < l_H$, and we already know a voter's l_L most preferred candidates, as well as her l_H most preferred candidates, then the voter no longer needs to communicate whether the l_L most preferred candidates are among her k most preferred candidates (because they must be), and she no longer needs to communicate whether the $m - l_H$ least preferred candidates are among her k most preferred candidates (because they cannot be). Thus the voter needs to communicate only $m - l_L - (m - l_H) = l_H - l_L$ bits in any given stage. Because each stage, $l_H - l_L$ is (roughly) halved, each voter in total communicates only (roughly) $m + m/2 + m/4 + \ldots \leq 2m$ bits. □

Theorem 19. *The nondeterministic communication complexity of the Bucklin rule is $\Omega(nm)$ (even to decide whether a given candidate a wins).*

Proof. We will exhibit a fooling set of size $2^{n'm'}$ where $m' = (m-1)/2$ and $n' = n/2$. We write the set of candidates as the following disjoint union: $C = \{a\} \cup L \cup R$ where $L = \{l_1, l_2, \ldots, l_{m'}\}$ and $R = \{r_1, r_2, \ldots, r_{m'}\}$. For any subset $S \subseteq \{1, 2, \ldots, m'\}$, let $L(S) = \{l_i : i \in S\}$ and let $R(S) = \{r_i : i \in S\}$. For every vector $(S_1, S_2, \ldots, S_{n'})$ consisting of n' sets $S_i \subseteq \{1, 2, \ldots, m'\}$, let the following vector of votes be an element of the fooling set:

- For $1 \leq i \leq n'$, let voter $2i-1$ rank the candidates $L(S_i) \succ R - R(S_i) \succ a \succ L - L(S_i) \succ R(S_i)$.
- For $1 \leq i \leq n'$, let voter $2i$ rank the candidates $L - L(S_i) \succ R(S_i) \succ a \succ L(S_i) \succ R - R(S_i)$.

We observe that this fooling set has size $(2^{m'})^{n'} = 2^{n'm'}$, and that candidate a wins in each vector of votes in the fooling set, for the following reason. Each candidate in $C - \{a\}$ is ranked among the top m' candidates by exactly half the voters (which is not enough to win). Thus, we need to look at the voters' top $m'+1$ candidates, and a is ranked $m'+1$th by all voters. All that remains to show is that for any two distinct vectors of votes in the fooling set, we can let each of the voters vote according to one of these two vectors in such a way that a loses. Let the first vote vector correspond to the vector $(S_1^1, S_2^1, \ldots, S_{n'}^1)$, and let the second vote vector correspond to the vector $(S_1^2, S_2^2, \ldots, S_{n'}^2)$. For some i, we must have $S_i^1 \neq S_i^2$, so that either $S_i^1 \nsubseteq S_i^2$ or $S_i^2 \nsubseteq S_i^1$. Without loss of generality, suppose $S_i^1 \nsubseteq S_i^2$, and let j be some integer in $S_i^1 - S_i^2$. Now, construct a new vote vector by taking vote $2i-1$ from the first vote vector, and the remaining votes from the second vote vector. In this newly constructed vote vector, a is still ranked $m'+1$th by all votes. However, l_j is ranked among the top m' candidates by $n'+1 = n/2+1$ votes. This is because whereas vote $2i-1$ does not rank l_j among the top m' candidates in the second vote vector (because $j \notin S_i^2$, we have $l_j \notin L(S_i^2)$), vote $2i-1$ does rank l_j among the top m' candidates in the first vote vector (because $j \in S_i^1$, we have $l_j \in L(S_i^1)$). So, a is not the winner in the newly constructed vote vector, and hence we have a correct fooling set. □

Theorem 20. *The nondeterministic communication complexity of the ranked pairs rule is $\Omega(nm \log m)$ (even to decide whether a given candidate a wins).*

Proof. We omit this proof because of space constraint. □

5. DISCUSSION

One key obstacle to using voting for preference aggregation is the communication burden that an election places on the voters. By lowering this burden, it may become feasible to conduct more elections over more issues. In the limit, this could lead to a shift from representational government to a system in which most issues are decided by referenda—a veritable *e-democracy*. In this paper, we analyzed the communication complexity of the common voting rules. Knowing which voting rules require little communication is especially important when the issue to be voted on is of low enough importance that the following is true: the parties involved are willing to accept a rule that tends to produce outcomes that are slightly less representative of the voters' preferences, if this rule reduces the communication burden on the voters significantly. The following table summarizes the results we obtained.

Rule	Lower bound	Upper bound
plurality	$\Omega(n \log m)$	$O(n \log m)$
plurality w/ runoff	$\Omega(n \log m)$	$O(n \log m)$
STV	$\Omega(n \log m)$	$O(n(\log m)^2)$
Condorcet	$\Omega(nm)$	$O(nm)$
approval	$\Omega(nm)$	$O(nm)$
Bucklin	$\Omega(nm)$	$O(nm)$
cup	$\Omega(nm)$	$O(nm)$
maximin	$\Omega(nm)$	$O(nm)$
Borda	$\Omega(nm \log m)$	$O(nm \log m)$
Copeland	$\Omega(nm \log m)$	$O(nm \log m)$
ranked pairs	$\Omega(nm \log m)$	$O(nm \log m)$

Communication complexity of voting rules, sorted from low to high. All of the upper bounds are deterministic (with the exception of maximin, for which the best deterministic upper bound we proved is $O(nm \log m)$). All of the lower bounds hold even for nondeterministic communication and even just for determining whether a given candidate a is the winner.

One area of future research is to study what happens when we restrict our attention to communication protocols that do not reveal any strategically useful information. This restriction may invalidate some of the upper bounds that we derived using multistage communication protocols. Also, all of our bounds are worst-case bounds. It may be possible to outperform these bounds when the distribution of votes has additional structure.

When deciding which voting rule to use for an election, there are many considerations to take into account. The voting rules that we studied in this paper are the most common ones that have survived the test of time. One way to select among these rules is to consider recent results on complexity. The table above shows that from a communication complexity perspective, plurality, plurality with runoff, and STV are preferable. However, plurality has the undesirable property that it is computationally easy to manipulate by voting strategically [3, 7]. Plurality with runoff is NP-hard to manipulate by a coalition of weighted voters, or by an individual that faces correlated uncertainty about the others' votes [7, 6]. STV is NP-hard to manipulate in those settings as well [7], but also by an individual with perfect knowledge of the others' votes (when the number of candidates is unbounded) [2]. Therefore, STV is more robust, although it may require slightly more worst-case communication as per the table above. Yet other selection criteria are the computational complexity of determining whether enough information has been elicited to declare a winner, and that of determining the optimal sequence of queries [8].

6. REFERENCES

[1] Lawrence Ausubel and Paul Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1, 2002. No. 1, Article 1.

[2] John Bartholdi, III and James Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[3] John Bartholdi, III, Craig Tovey, and Michael Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[4] Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research*, 5:649–667, 2004.

[5] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions: Extended abstract. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 256–259, 2001.

[6] Vincent Conitzer, Jerome Lang, and Tuomas Sandholm. How many candidates are needed to make elections hard to manipulate? In *Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 201–214, 2003.

[7] Vincent Conitzer and Tuomas Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 314–319, 2002.

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

[9] Sven de Vries, James Schummer, and Rakesh V. Vohra. On ascending auctions for heterogeneous objects, 2003. Draft.

[10] Allan Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[11] Benoit Hudson and Tuomas Sandholm. Effectiveness of query types and policies for preference elicitation in combinatorial auctions. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 386–393, 2004.

[12] E Kushilevitz and N Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[13] Sebastién Lahaie and David Parkes. Applying learning algorithms to preference elicitation. In *Proceedings of the ACM Conference on Electronic Commerce*, 2004.

[14] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 2005. Forthcoming.

[15] David Parkes. iBundle: An efficient ascending price bundle auction. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 148–157, 1999.

[16] Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 256–262, 1993.

[17] Tuomas Sandholm and Craig Boutilier. Preference elicitation in combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 10. MIT Press, 2005.

[18] Paolo Santi, Vincent Conitzer, and Tuomas Sandholm. Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. In *Conference on Learning Theory (COLT)*, pages 1–16, 2004.

[19] Mark Satterthwaite. Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[20] Ilya Segal. The communication requirements of social choice rules and supporting budget sets, 2004. Draft. Presented at the DIMACS Workshop on Computational Issues in Auction Design, Rutgers University, New Jersey, USA.

[21] Peter Wurman and Michael Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 21–29, 2000.

[22] A. C. Yao. Some complexity questions related to distributed computing. In *Proceedings of the 11th ACM symposium on theory of computing (STOC)*, pages 209–213, 1979.

[23] Martin Zinkevich, Avrim Blum, and Tuomas Sandholm. On polynomial-time preference elicitation with value queries. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 176–185, 2003.