Communication Complexity (Spring'26) April 1, 2026

Lecture: Applications in Social Choice Theory

Instructor: *Toniann Pitassi* Presenter: *Philip Kaelbling*

1 Introduction to Social Choice Theory

Social choice theory is the mathematical study of how individual preferences are aggregated into a collective decision. The central object is a *voting rule*: a function that takes as input the ranked preferences of n voters and returns a winner (or a ranking) among m candidates.

The theory dates to the 18th-century work of Condorcet and Borda, received its modern axiomatic treatment from Arrow [3], and has since become a core topic in economics, political science, and (as we will soon see) theoretical computer science.

Our focus in these notes is the *communication complexity* of voting rules: how many bits of information must voters reveal in order for the rule to be computed? This question is both practically important (bandwidth is finite in real systems) and technically rich, connecting voting theory to classic results in communication complexity such as fooling sets and disjointness lower bounds.

2 The Formal Setup

Definition 1 (Preference profile)**.** *Let* $[n] = \{1, \ldots, n\}$ *be the set of* voters *and* $[m] = \{1, \ldots, m\}$ *be the set of* candidates *(also called* alternatives*). A* preference order σ_i *for voter* i *is a complete, transitive, antisymmetric binary relation over* $[m]$*i.e., a strict total order. A* preference profile *is the* n*-tuple* $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)$*.*

There are $(m!)^n$ possible preference profiles, so naively communicating a profile requires $n \log_2(m!) = \Theta(nm \log m)$ bits.

Definition 2 (Social choice function)**.** *A* social choice function *(or* voting rule*) is a function*

$$f : \mathcal{L}([m])^n \to [m],$$

where $\mathcal{L}([m])$ *is the set of all strict total orders over* $[m]$*. We primarily study* winner selection *(also called* 1-selection*): finding the unique top element of* $f(\boldsymbol{\sigma})$*.*

Remark 2.1. In some formulations f returns a full ranking; in others it is a partial function (see the Condorcet rule, Section 5.3). We will be explicit about which case applies.

3 Arrow's Impossibility Theorem

Before cataloguing specific rules, it is worth understanding a fundamental negative result that explains *why* no single rule is universally satisfactory.

Definition 3 (Three classical axioms)**.** *Let f be a social choice function for $m \geq 3$ candidates.*

1. ***Non-dictatorship:*** *There is no voter i^* such that $f(\boldsymbol{\sigma})$ always equals the top candidate in σ_{i^*}, regardless of other voters.*

2. ***Pareto efficiency:*** *If every voter prefers candidate A to candidate B (i.e., $A \succ_i B$ for all i), then B is never the winner.*

3. ***Independence of Irrelevant Alternatives (IIA):*** *The social ranking between any two candidates A and B depends only on the voters' relative rankings of A vs. B, not on how other candidates are ranked.*

Theorem 4 (Arrow's Impossibility Theorem [3])**.** *For $m \geq 3$ candidates, no social choice function satisfies all three of non-dictatorship, Pareto efficiency, and IIA simultaneously.*

Arrow's theorem tells us that any reasonable voting rule must sacrifice at least one of these properties. This motivates studying a broader menu of desiderata and rules, each making different tradeoffs.

4 Desiderata for Voting Rules

Beyond Arrow's three axioms, practitioners and theorists care about many additional properties.

Definition 5 (Further desiderata)**.** *Let f be a social choice function.*

- ***Monotonicity:*** *Ranking a candidate strictly higher (while keeping all other relative rankings fixed) cannot decrease that candidate's chance of winning.*

- ***Consistency:*** *If candidate a wins for two disjoint, non-empty electorates V_1 and V_2, then a wins for the combined electorate $V_1 \cup V_2$.*

- ***Majority-winning:*** *If a candidate is ranked first by a strict majority of voters ($> n/2$), that candidate wins.*

- ***Condorcet-winning:*** *If a candidate a beats every other candidate in every pairwise election (i.e., $|\{i : a \succ_i b\}| > n/2$ for all $b \neq a$), then a wins. Such a candidate is called the* Condorcet winner*.*

- ***Anonymity:*** *Permuting the voters' ballots does not change the outcome.*

- ***Neutrality:*** *Permuting candidate labels on all ballots correspondingly permutes the outcome; no candidate is treated differently by the rule itself.*

Remark 4.1. Note that majority-winning and Condorcet-winning are logically related but distinct: the Condorcet property speaks to pairwise comparisons, while majority-winning speaks to first-place counts. Any Condorcet winner with n odd is also the majority winner when $m = 2$, but not necessarily for $m \geq 3$.

5 Classical Voting Rules

5.1 Scoring Rules

Definition 6 (Scoring rule)**.** *Fix a* score vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{Z}^m$ *with* $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$*. The scoring rule induced by* $\boldsymbol{\alpha}$ *awards* α_k *points to a candidate ranked* k*-th by a voter. The* score *of candidate* a *is the sum of points it receives across all voters, and the winner is the candidate with the highest score.*

Plurality: $\boldsymbol{\alpha} = (1, 0, \ldots, 0)$. Each voter casts one vote for their top-ranked candidate; the candidate with the most votes wins.

- *Pros:* Communication-efficient (each voter sends only $\lceil \log m \rceil$ bits), monotone, consistent, majority-winning.
- *Cons:* Fails the Condorcet criterion and IIA. Classic examples show a candidate preferred by a majority in pairwise contests can lose under plurality ("spoiler effect").

Borda Count: $\boldsymbol{\alpha} = (m-1, m-2, \ldots, 1, 0)$. A candidate ranked k-th by voter i receives $m-k$ points.

- *Pros:* Good social-welfare maximizer, monotone, consistent.
- *Cons:* Fails the Condorcet criterion and the majority criterion. The Borda winner can be the Condorcet loser in contrived examples.

Example 7. *Three voters, three candidates* $\{a, b, c\}$*:*

$$\sigma_1 : a \succ b \succ c, \quad \sigma_2 : a \succ c \succ b, \quad \sigma_3 : b \succ c \succ a.$$

Borda scores: a *gets* $2 + 2 + 0 = 4$*,* b *gets* $1 + 0 + 2 = 3$*,* c *gets* $0 + 1 + 1 = 2$*. Winner:* a*.*

5.2 Single Transferable Vote (STV)

STV proceeds in $m - 1$ elimination rounds. In each round, every voter's ballot counts toward their highest-ranked *remaining* candidate. The candidate with the lowest plurality score is eliminated. The last remaining candidate wins.

- *Pros:* Widely used in practice (Ireland, Australia, many others); satisfies the majority criterion.
- *Cons:* Fails monotonicity, consistency, the Condorcet criterion, and IIA. The failure of monotonicity is particularly counterintuitive: a candidate can be made to *lose* by voters ranking them *higher*.

Example 8. *Three voters, three candidates* $\{a, b, c\}$*:*

$$\sigma_1 : a \succ b \succ c, \quad \sigma_2 : a \succ c \succ b, \quad \sigma_3 : b \succ a \succ c.$$

Round 1 plurality scores: $a : 2,\ b : 1,\ c : 0$*. Eliminate* c*. Round 2:* $a : 2,\ b : 1$*. Winner:* a*.*

5.3 Condorcet Method

Definition 9 (Condorcet winner). *Candidate a^* is a* Condorcet winner *if for every other candidate b,*

$$|\{i \in [n] : a^* \succ_i b\}| > n/2.$$

The Condorcet method elects the Condorcet winner if one exists; otherwise it returns no winner (the rule is *partial*).

- *Pros:* Satisfies the Condorcet criterion (tautologically) and monotonicity.
- *Cons:* Not total. A Condorcet winner may fail to exist ("Condorcet cycle").

Example 10 (Condorcet cycle). *Three voters, three candidates:*

$$\sigma_1 : a \succ b \succ c, \quad \sigma_2 : b \succ c \succ a, \quad \sigma_3 : c \succ a \succ b.$$

Pairwise: a beats b (2-1), b beats c (2-1), c beats a (2-1). No Condorcet winner exists.

This failure of totality is one of the key motivations for the Cup rule.

5.4 Cup (Knockout Tournament)

The Cup rule fixes a balanced binary-tree bracket over the m candidates (assuming m is a power of 2 for simplicity). At each internal node, the two candidates meeting in that match conduct a pairwise election; the winner advances. The candidate reaching the root wins.

- *Pros:* Total; satisfies the Condorcet criterion *for the specific bracket chosen* when a Condorcet winner exists (since a Condorcet winner beats any opponent); uses only $m-1$ pairwise elections, which is fewer than the $\binom{m}{2}$ needed to fully determine the Condorcet winner.
- *Cons:* Fails monotonicity, Pareto efficiency, and neutrality (the outcome depends on the bracket, so candidates are not treated symmetrically).

Remark 5.1. The Cup rule is a clean illustration of how *procedural* choices (the bracket) can substantially affect outcomes even when the underlying preferences are fixed.

6 Communication Complexity of Voting Rules

6.1 Setting and Model

We follow the framework of Conitzer and Sandholm [1].

Definition 11 (Communication model). *We use the* number-in-hand, blackboard *model: each voter i holds a private preference order $\sigma_i \in \mathcal{L}([m])$. Voters communicate by writing bits on a shared* blackboard *visible to all. The communication complexity of a protocol is the total number of bits written on the blackboard in the worst case. The* deterministic communication complexity *$D(f)$ is the minimum over all deterministic protocols computing f. The* nondeterministic communication complexity *$N(f)$ is the minimum over all nondeterministic protocols.*

Remark 6.1. We are interested in computing whether a *given* candidate a is the winner, not in communicating the entire preference profile. Since winner selection is the primary goal, the lower bounds we prove are for this decision version of the problem. The upper bounds hold for computing the full winner.

Trivial upper bound Since any preference order can be encoded in $\lceil \log(m!) \rceil = O(m \log m)$ bits, communicating all n profiles costs $O(nm \log m)$ bits. Hence every voting rule has communication complexity $O(nm \log m)$.

6.2 Summary of Results

The following table (from [1]) gives tight or near-tight bounds for the five rules discussed above. The lower bounds are *nondeterministic* and hold even for the winner-decision problem. The upper bounds are deterministic. Note the $\log m$ gap in the STV bounds. As far as I know, this gap is open to this day.

Rule	**Lower Bound**	**Upper Bound**
Plurality	$\Omega(n \log m)$	$O(n \log m)$
STV	$\Omega(n \log m)$	$O(n \log^2 m)$
Condorcet	$\Omega(nm)$	$O(nm)$
Cup	$\Omega(nm)$	$O(nm)$
Borda	$\Omega(nm \log m)$	$O(nm \log m)$

6.3 The Fooling Set Method

The primary lower-bound technique is the *fooling set lemma*, a standard tool from communication complexity [2]. We've seen the 2-player version, where the object of interest is a 2-dimensional matrix. For the multiparty setting, the matrix of interest is higher-dimensional.

Definition 12 (Fooling set (multiparty))**.** *Let $f : X_1 \times \cdots \times X_n \to \{0,1\}$. A* fooling set *for the value $f_0 \in \{0,1\}$ is a collection of input tuples $\{(x_1^{(1)}, \ldots, x_n^{(1)}), \ldots, (x_1^{(k)}, \ldots, x_n^{(k)})\}$ such that:*

1. *For every ℓ, $f(x_1^{(\ell)}, \ldots, x_n^{(\ell)}) = f_0$.*
2. *For every $\ell \neq \ell'$, there exists a "mixing" tuple $(r_1, \ldots, r_n)$ with each $r_i \in \{\ell, \ell'\}$ such that $f(x_1^{(r_1)}, \ldots, x_n^{(r_n)}) \neq f_0$.*

Theorem 13 (Fooling set lower bound [2])**.** *If there exists a fooling set of size k for f, then the nondeterministic communication complexity of f satisfies $N(f) = \Omega(\log k)$.*

Note that this lemma applies to nondeterministic communication complexity as well, in addition to the deterministic communication complexity discussed in class. The intuition behind the deterministic version is similar to that discussed in class, but the induced matrix of a multiparty protocol is k-dimensional. Formally, every possible transcript in a correct protocol corresponds to a combinatorial subrectangle of the protocol's induced matrix. Messages sent by a player depend only on that player's input and the existing transcript. So, if you take any input tuple consistent with a transcript and independently swap out each input for any other in the same subrectangle, all messages in that transcript are still the same. Therefore, if two inputs share a transcript, so must every mix of those two inputs.

Then, suppose we have some fooling set F, some protocol π with deterministic communication complexity C that is correct for f, and some $s, t \in F$ share a transcript. This transcript corresponds to some shared subrectangle (that, by definition of rectangle, must contain every mix of the s and t). As $f(s) = 1$ and the protocol is assumed to be correct, it follows that $f(s \bowtie t) = 1$. But, $f(s \bowtie t) = 0$ by construction of the fooling set. Therefore, it must be the case that every $a \in F$ has a distinct transcript. There are at most 2^C transcripts for a protocol using C bits, so it must be the case that $C \geq \log |F|$.

6.4 Borda Lower Bound via Fooling Sets

6.4.1 A Warmup

Let's start with a warmup. We'll have 6 voters, 4 candidates A, B, C, D, and a fooling set of size 2. Our fooling set elements are as follows:

Voter Index	Fooling Set Element 1	Fooling Set Element 2
1	$A \succ B \succ C \succ D$	$A \succ B \succ D \succ C$
2	$A \succ B \succ C \succ D$	$A \succ B \succ D \succ C$
3	$D \succ C \succ B \succ A$	$C \succ D \succ B \succ A$
4	$D \succ C \succ B \succ A$	$C \succ D \succ B \succ A$
5	$A \succ B \succ C \succ D$	$A \succ B \succ C \succ D$
6	$D \succ C \succ A \succ B$	$D \succ C \succ A \succ B$

Note that in both of the above elements, A wins with 10 points, beating both C and D by exactly one point. We then mix by taking voters 1 and 2 from element 1, and 3-6 from element 2, giving us a preference profile:

Voter Index	Mixed Preference Profile
1_1	$A \succ B \succ C \succ D$
2_1	$A \succ B \succ C \succ D$
3_2	$C \succ D \succ B \succ A$
4_2	$C \succ D \succ B \succ A$
5_2	$A \succ B \succ C \succ D$
6_2	$D \succ C \succ A \succ B$

Now, C wins by 1! So, our fooling set of two is indeed a fooling set. Obviously, two elements is not a helpful fooling set, so how big can we make this? Remember that we have n voters and m candidates. We need two reserved candidates (previously A and B) for things to work out, so we'll have $a = m - 2$ left to work over. We need this property where every 4 voters cancel out (as above), and then two extra voters for A to win by 1 point with, so we can have $b = \frac{n-2}{4}$ different preferences in each element of the fooling set. Finally, we can construct an element of the fooling set for every vector $(\sigma_1, \ldots \sigma_b)$ of b orderings over all candidates other than A and B. (in the warmup, $b = 1$, so this vector is of length 1. Specifically, fooling set 1 was constructed using $(\sigma_1 = C \succ D)$, and 2 was constructed using $(\sigma_1 = D \succ C)$). Finally, the last object we need is some fixed order over all candidates other than A and B that is shared by all elements of the fooling set. Above, this was $C \succ D$. As there are $(a!)^b$ different such vectors, we can construct a fooling set of size $(a!)^b$.

6.4.2 In Generality

Here, I present the $\Omega(nm \log m)$ bound for Borda.

Proof. We exhibit a fooling set of size $(a!)^b$, for $a = m - 2, b = \frac{n-2}{4}$. Let $\sigma_i : [a] \to A - \{\alpha, \beta\}$ such that $\sigma_i(j) = c$ meaning candidate c is ranked j-th in σ_i. We make use of some fixed ordering $\sigma_0 : [a] \to A - \{\alpha, \beta\}$ that is shared between all elements of the fooling set. For every vector of $(\sigma_1, \ldots, \sigma_b)$ of b orderings of all candidates other than some fixed α, β, we construct an element of the fooling set. (Note: there are $a!$ such orderings, and therefore $(a!)^b$ such vectors). So, for a given $(\sigma_1, \ldots, \sigma_b)$, we define a fooling set element of n preferences over all m candidates as follows:

1. For $1 \leq i \leq b$, let voters $4i - 3$ and $4i - 2$ rank:

$$\alpha \succ \beta \succ \sigma_i(1) \succ \cdots \succ \sigma_i(a)$$

2. For $1 \leq i \leq b$, let voters $4i - 1$ and $4i$ rank:

$$\sigma_i(a) \succ \sigma_i(a-1) \succ \cdots \succ \beta \succ \alpha$$

3. Let voter $4b + 1 = n - 1$ rank:

$$\alpha \succ \beta \succ \sigma_0(1) \succ \cdots \succ \sigma_0(a)$$

4. Finally, let voter $4b + 2 = n$ rank:

$$\sigma_0(a) \succ \cdots \sigma_0(1) \succ \alpha \succ \beta$$

Note that, in every element of the fooling set, α wins over $A - \{\alpha, \beta\}$ by exactly one point, and wins over β by two points.

Now, we need to show that any two distinct preference profiles can mix such that α loses. Note that an element of the fooling set is uniquely associated with a partial preference profile (as mentioned above). Therefore, let the first fooling set element be induced by $(\sigma_1^1, \ldots, \sigma_b^1)$, and the second by $(\sigma_1^2, \ldots, \sigma_b^2)$. As these are assumed to be distinct, there must be some i such that $\sigma_i^1 \neq \sigma_i^2$. As these are preferences, it must be that there exists some γ such that $(\sigma_i^1)^{-1}(\gamma) < (\sigma_i^2)^{-1}(\gamma)$ (1 ranks γ higher than 2 does).

We then construct our mixed preference profile as follows: Take voters' $4i - 3, 4i - 2$ preferences from preference profile 1, and the rest from preference profile 2. Note that the scores of α and β are unchanged (the same as they are for any element of the fooling set). But, γ's score has increased by at least two. As α was only winning by one point, it is now the case that γ wins, so our constructed set is a fooling set as desired. To conclude, we note that $\log(a!)^b = \Omega(ab \log b) = \Omega(nm \log m)$, as desired. □

7 Implicit Utilitarian Voting and Distortion

7.1 Motivation

The framework of Section 5.3 models voter preferences as *ordinal*: voters report rankings, and the voting rule operates on rankings. But voters also have *cardinal* preferencesthey care not just about which candidate they prefer, but by *how much*.

The implicit utilitarian model, introduced by Procaccia and Rosenschein [4], formalizes this idea. Voters have hidden numerical utility functions; their ordinal ranking is merely a proxy. The designer cannot access these utilities directly (that would require infinite bits), but must make good decisions using limited information.

7.2 Formal Model

Definition 14 (Valuation function)**.** *Let A be the set of m candidates. A* normalized valuation function *for voter i is a function $v_i : A \to \mathbb{R}_{\geq 0}$ such that $\sum_{a \in A} v_i(a) = 1$. The space of all such functions is the $(m-1)$-dimensional simplex Δ^m.*

Definition 15 (Social welfare)**.** *For a valuation vector $\vec{v} = (v_1, \ldots, v_n) \in (\Delta^m)^n$ and a candidate $a \in A$, the* social welfare *is*

$$\mathrm{sw}(\vec{v}, a) = \sum_{i \in [n]} v_i(a).$$

The optimal candidate is $a^* = \arg\max_{a \in A} \mathrm{sw}(\vec{v}, a)$.

Definition 16 (Voting rule (new model))**.** *A* voting rule *f in this model consists of:*

- *An* elicitation rule *Π_f: a distribution over a set of possible queries $\mathcal{Q}$. If Π_f is a point mass, the elicitation is* deterministic*.*
- *An* aggregation rule *Γ_f: a function mapping a tuple of voter responses $(\rho_1, \ldots, \rho_n)$ to a distribution over A. If Γ_f always outputs a point mass, the aggregation is* deterministic*.*

The protocol is: (1) sample query $q \sim \Pi_f$; (2) send q to every voter; (3) voter i responds with ρ_i according to v_i; (4) output $\hat{a} \sim \Gamma_f(\rho_1, \ldots, \rho_n)$.

Remark 7.1. The model is *simultaneous*: all voters receive the same query and respond in parallel. This is a special case of the blackboard model. Note also that the "query" can be as rich as "report your full ordinal ranking"the classical rules are special cases of this model.

Definition 17 (Communication complexity, revisited)**.** *The* communication complexity *$C^m(f)$ of rule f for m alternatives is the* expected *number of bits elicited from each voter by f.*

Definition 18 (Distortion)**.** *The* distortion *of f for m alternatives is*

$$\mathrm{dist}^m(f) = \sup_{\vec{v} \in (\Delta^m)^n} \frac{\max_{a \in A} \mathrm{sw}(a, \vec{v})}{\mathbb{E}_{\hat{a} \sim f(\vec{v})}[\mathrm{sw}(\hat{a}, \vec{v})]}.$$

This is the worst-case ratio between the optimal social welfare and the expected social welfare of the candidate selected by f. Smaller distortion is better; $\mathrm{dist}(f) = 1$ *means f is always optimal.*

Remark 7.2. Distortion is inherently a worst-case measure over the space of all possible valuation profiles consistent with the information revealed to the rule. For example, under plurality (where voters report only their top choice), the distortion is $\Theta(m)$: the rule might always elect a candidate whose true social welfare is only $1/m$ of optimal. With full ordinal information, distortion improves substantially but remains $\Omega(\sqrt{m})$ for any deterministic rule.

7.3 Distortion–Communication Tradeoff

The central theorem connecting the two frameworks is due to Mandal, Procaccia, Shah, and Wajc [5].

Theorem 19 (Distortion–communication tradeoff [5])**.** *For any $d \geq 1$, if a voting rule f uses a* deterministic *elicitation rule and satisfies* $\mathrm{dist}(f) \leq d$*, then*

$$C(f) = \Omega\Big(\frac{m}{d}\Big).$$

Theorem 19 reveals a fundamental tradeoff: to reduce distortion by a factor of d (i.e., get d times closer to optimal social welfare), the rule must ask each voter $\Omega(m/d)$ bits. In particular:

- $d = O(1)$: constant distortion requires $\Omega(m)$ bits per voter (i.e., the full cardinal information).
- $d = O(\sqrt{m})$: achievable with $O(\sqrt{m})$ bits per voter.
- $d = O(m)$: trivially achievable with $O(1)$ bits (e.g., a random candidate has distortion $O(m)$).

7.4 Reduction to Disjointness

The lower bound in Theorem 19 is proved by a reduction from a promise version of the multiparty disjointness problem, which we now describe.

Definition 20 (Fixed-size disjointness, $\mathrm{FDISJ}_{m,s,t}$)**.** *There are t players and a universe $[m]$. Player i holds a subset $S_i \subseteq [m]$ with $|S_i| = s$. The problem is:*

- YES*: there exists an element $x \in [m]$ such that $x \in S_i$ for all $i \in [t]$ (all sets share a common element).*
- NO*: all sets $S_1, \ldots, S_t$ are pairwise disjoint.*

The standard disjointness problem is hard under various promise conditions. For the voting application, the relevant variant uses a *substantial intersection* promise:

Definition 21 (Substantial intersection promise)**.** *Fix $\gamma > 0$. In every* YES *instance, there exists at least one element $x \in [m]$ and a subset of players $P \subseteq [t]$ with $|P| \geq \gamma t$ such that $x \in S_i$ for every $i \in P$.*

Theorem 22 ([5])**.** *Under the substantial intersection promise with $\gamma \leq 1/76$ and $t \leq (m/2)(1 - 1/e)$, we have*

$$D\big(\mathrm{FDISJ}_{m,\,m/t,\,t}\big) \geq m.$$

Proof idea: The lower bound is established by monochromaticity of combinatorial rectangles. A valid protocol induces a partition of the input space into monochromatic rectangles. One shows that any NO instance and any YES instance with substantial intersection cannot be covered by the same rectangle, so an exponential number of rectanglesand hence $\Omega(m)$ bitsare needed.

The reduction: To reduce FDISJ to voting, interpret each element of the universe as a candidate and each set S_i held by player i as the set of candidates that player i values highly. A YES instance (substantial intersection) corresponds to a valuation profile in which there exists a candidate with high social welfare (substantially supported across voters). A NO instance (disjoint sets) corresponds to a profile where no candidate is widely supported.

Formally, a low-distortion voting rule must be able to find the high-welfare candidate in YES instances, which is exactly like finding a substantially intersecting element in the disjointness problem. The communication lower bound for disjointness therefore transfers to a communication lower bound for the voting rule.

8 Other Applications of Communication Complexity in Game Theory

The techniques we have developedespecially reductions from disjointness and related problemsapply broadly throughout algorithmic game theory. We survey several further results.

8.1 Stable Matching

Definition 23 (Stable matching)**.** *Given n men and n women, each with a complete strict preference order over members of the opposite side, a* matching μ *is* stable *if there is no* blocking pair*: a man m and woman w such that both prefer each other to their assigned partners under μ.*

The Gale–Shapley algorithm computes a stable matching in $O(n^2)$ queries (pairwise comparisons). Is this optimal?

Theorem 24 (Gonczarowski et al. [6])**.** *Any algorithm that computes a stable matching using Boolean preference queries requires $\Omega(n^2)$ queries in the worst case. This holds even for randomized algorithms.*

The proof reduces to a variant of disjointness: each man's preference list can be encoded as a set, and determining stability requires resolving a sufficient number of intersection queries.

8.2 Approximate Nash Equilibria

Theorem 25 (Babichenko and Rubinstein [7])**.** *Finding an ε-Nash equilibrium in an $N \times N$ bimatrix game requires $\Omega(N^2)$ communication (for constant ε).*

This result was obtained via two independent approaches:

1. A query complexity lower bound for the *end-of-line* problem combined with a simulation theorem.

2. A lifting theorem of Gs, Pitassi, and Watson [8] relating query complexity to communication complexity, applied to the Brouwer Fixed Point problem (of which Nash equilibria are a special case via the Brouwer–Nash equivalence).

8.3 Combinatorial Auctions

Definition 26 (Welfare maximization)**.** *Given k bidders and m items, each bidder i has a valuation $v_i(S)$ for each subset $S \subseteq [m]$ of items. The goal is to find a partition $(S_1, \ldots, S_k)$ of $[m]$ that maximizes total welfare $\sum_i v_i(S_i)$.*

Theorem 27 (Nisan [9])**.** *The communication complexity of exact welfare maximization for k bidders and m items is* $\exp\big(\Omega(m/k^2)\big)$.

Dobzinski, Nisan, and Schapira [10] extended this to show that even for the special case of *subadditive* valuations (where $v(S \cup T) \leq v(S) + v(T)$), exact welfare maximization remains exponentially hard. Both proofs reduce from multiparty disjointness.

The unifying theme across all three applications is that finding a "socially good" outcome, whether a stable match, an equilibrium, or a high-welfare allocation, requires sufficient communication to resolve ambiguity about which inputs are "adjacent" in the combinatorial structure. Disjointness captures exactly this type of ambiguity.

References

[1] V. Conitzer and T. Sandholm, "Communication Complexity of Common Voting Rules," *Proc. 6th ACM Conference on Electronic Commerce (EC'05)*, pp. 78–87, 2005.

[2] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press, 1996.

[3] K. J. Arrow, "A Difficulty in the Concept of Social Welfare," *Journal of Political Economy*, vol. 58, no. 4, pp. 328–346, 1950.

[4] A. D. Procaccia and J. S. Rosenschein, "The Distortion of Cardinal Preferences in Voting," *Proc. 10th International Conference on Cooperative Information Agents (CIA'06)*, pp. 317–331, 2006.

[5] D. Mandal, A. D. Procaccia, N. Shah, and D. Wajc, "Optimal Communication-Distortion Tradeoff in Voting," *Proc. 21st ACM Conference on Economics and Computation (EC'20)*, pp. 795–813, 2020.

[6] Y. A. Gonczarowski, N. Nisan, R. Ostrovsky, and W. Rosenbaum, "A Stable Marriage Requires Communication," *Games and Economic Behavior*, vol. 118, pp. 626–647, 2019.

[7] Y. Babichenko and A. Rubinstein, "Communication Complexity of Approximate Nash Equilibria," *Proc. 49th Annual ACM STOC*, pp. 878–889, 2017.

[8] M. Göös, T. Pitassi, and T. Watson, "Query-to-Communication Lifting for BPP," *Proc. 58th IEEE FOCS*, pp. 132–143, 2017.

[9] N. Nisan, "The Communication Complexity of Approximate Set Packing," *Proc. 29th ICALP*, LNCS vol. 2380, pp. 868–875, 2002.

[10] S. Dobzinski, N. Nisan, and M. Schapira, "Approximation Algorithms for Combinatorial Auctions with Complement-Free Bidders," *Mathematics of Operations Research*, vol. 35, no. 1, pp. 1–13, 2010.