

# Between Fairness and a Mistrial: Consensus Under a Deadline

**Marina Bannikova**

Universitat Rovira i Virgili  
marina.bannikova@urv.cat

**Lihl Naamani-Dery**

Ariel University  
lihld@ariel.ac.il

**Svetlana Obratzsova**

Hebrew University of Jerusalem  
svetlana.obraztsova@gmail.com

**Zinovi Rabinovich**

Mobileye Vision Technologies Ltd.  
zr@zinovi.net

**Jeffrey S. Rosenschein**

Hebrew University of Jerusalem  
jeff@cs.huji.ac.il

## Abstract

Jury trial is, perhaps, the most prominent example of seeking a consensus. The process is particularly difficult if the judge places a deadline by which the jury must reach a unanimous decision, otherwise declaring a mistrial. A mistrial is commonly perceived to be worse than any decision the jury might render. As a result, while each juror has her own idea about the fairness of each possible trial outcome, she may eventually choose to vote for a less fair outcome, rather than cause a mistrial by breaking unanimity.

In this paper we propose a model for the above scenario — *Consensus Under a Deadline (CUD)* — based on a time-bounded iterative voting process. We provide some theoretical features of CUDs, particularly focusing on convergence guarantees and the quality of the final decision. An extensive experimental study demonstrates the more subtle features of CUDs, e.g., the difference between two simple types of juror behaviour, lazy vs. proactive voters.

## 1 Introduction

Consider a group of individuals that need to agree on one alternative out of several options. Add a deadline, so that the decision must be reached before a certain time limit, otherwise some default alternative, least preferred by everyone, is chosen. Require unanimity, so that the chosen alternative should be supported by every individual. Under this model, some questions come to mind. How often will the group reach a consensus? What effort will be required from the voters? What attributes will the decision have?

Consider, for instance, a jury trial. There are a few alternatives (e.g., “guilty”, “not guilty”, “not proven”). A civil trial jury is asked to reach a decision by majority agreement, while a criminal case decision requires unanimity, i.e. all the jurors must agree on the verdict. If no decision is reached until a certain deadline, a mistrial is declared. A mistrial is possibly the worst alternative for every juror, since it means that the case is not decided. During the actual trial the jurors do not communicate with one another, and during deliberations they

discuss the case only inside the jury room. The intent may be that (at first) they should independently decide which verdict they consider to be most fair, and potentially have an order of preferences over possible verdicts. Jury deliberation consists of a discussion about the trial case, and searching for agreement about a verdict. In the discussion, jurors reveal some information about which verdict they consider to be fair. Using a simple voting process (“raise your hands”), jurors vote to find the most preferred outcome. The process is iterative, where at every stage some discussion is held, followed by another voting process. It ends either when the jury reaches a consensus, or when the deadline is reached, the sooner of the two.

A jury trial is not the only scenario where consensus under a deadline may be required. Other examples include a group of friends discussing their next vacation destination; members of a university department choosing a day for a weekly seminar; a scientific committee deciding where to hold next year’s conference; even a casual problem such as a family dinner can cause some discussions.

All situations above have several common features. First, there is a strict deadline for reaching an agreement: a judge’s requirement, national holidays, the start of an academic year, the budget approval deadline, or dinnertime. Second, assuming that individuals at least somewhat differ in their preferences, it is unlikely that a unanimous consensus will be reached immediately by a simultaneous vote. A consensus is usually reached, if at all, only after several rounds of a sequential voting process. Commonly used methods include emails, calls, messages, or on-line platforms such as Doodle.

Inspired by these scenarios, we define a strict, formal, time-bounded iterative voting process. The process begins when each voter reveals her most preferred alternative. The preferences are aggregated using majority voting with a threshold (unanimity being the most extreme threshold). If a consensus is not instantly reached, a voting process begins. Voters may choose to change their vote. For instance, a voter that realises that her most preferred alternative has no chance of being elected, might decide to change her mind and vote for another alternative. At each stage, all voters that wish to vote apply for a voting slot. One voter is chosen randomly; the chosen voter states her currently most-preferred alternative. Each stage is defined as one clock tick. The process ends when a consensus is reached, or when the deadline is reached,

the sooner of the two. The voters’ ultimate goal is to reach a consensual decision, and do so before the deadline.

Assuming rational voter behaviour, we define two types of voters: a *proactive voter*, who likes to participate actively in the procedure and will not miss any chance to state her preferences, and a *lazy voter*, who prefers not to act until it is really necessary. We assume a sequential, iterative voting process, and that each voter has a strict order of preferences over the alternatives that she keeps strictly to herself. Bargaining is not permitted. The voter is allowed to apply for a “voting slot”, i.e., an opportunity to vote. When the voter is given permission to vote, she reveals the alternative for which she wants to vote. The voter is allowed to strategically change her vote as often as she likes.

**Contributions:** We provide what is, to the best of our knowledge, the first model for iterative voting processes with a deadline (Section 3). The model can easily be generalised to other voting rules, but for the ease of exposition, we initiate this line of research with a specific model, namely, *Consensus Under a Deadline*, based on Plurality with a threshold (also known as Majority). Then, in Section 3.1, we prove the theoretical properties of CUDs, such as stopping, guarantees of no-mistrial runs, and the (additive) Price of Anarchy bounds. Since the progress of CUDs has a rigorous algorithmic description, it is possible to effectively simulate such games. We resort to such a simulation to investigate statistical properties and tradeoffs of CUDs. In more detail, Section 4 concentrates on Unanimous CUDs, and provides an encompassing experimental analysis of CUD trade-offs. In particular, we measure the effects of voter behavioural types (lazy vs. proactive) on the number of voting steps and the Price of Anarchy. We find an indication of a trade-off between the fairness of the voting outcome and the effort required from the voters during the process.

## 2 Related Work

Deadline scenarios are abundant in models of bargaining; “[f]inal deadlines are fixed time limits that end a negotiation” [Ma and Manove, 1993; Moore, 2004]. One of the first experimental studies on deadlines in bargaining demonstrated the deadline effect: a majority of agreements are obtained in the final seconds before the deadline; even complete information does not speed agreement much, contrary to what might be expected [Roth *et al.*, 1988]. The concession rate increases as the deadline approaches [Cramton and Tracy, 1992; Lim and Murnighan, 1994]. However, there is a probability that no agreement will be reached before the deadline [Ma and Manove, 1993]; more than half of negotiations may end without agreement [Cramton and Tracy, 1992]. Perhaps surprisingly, moderate deadlines were found to have a positive effect on the outcome of negotiation, though the participants expected that the deadline would hurt their negotiation [Moore, 2004].

Deadlines in sequential voting can be represented as a moment in time when the voters become indifferent between alternatives and are willing to finish the process with any alternative winning. “Indifferent time” is the moment in time when the utility of the least-preferred alternative becomes

equal to the utility of the most-preferred alternative a moment later. The agreement is reached immediately at the first round of voting, and it favours the pivotal voter with higher indifference time [Kwiek, 2014].

While in a bargaining procedure the outcome may be a compromise among the most preferred outcomes by the rivals, in CUDs a strict alternative must be chosen. For example, in a jury trial there is no compromise between “guilty” – “not-guilty”. We focus on iterative voting processes under a deadline and do not allow negotiation or bargaining. Informally, a deadline is a fixed time limit that ends the voting process.

In general, CUDs closely follow the way other iterative voting games progress, e.g., [Meir *et al.*, 2010; Reijngoud and Endriss, 2012; Naamani-Dery *et al.*, 2015; Obraztsova *et al.*, 2015]. However, CUDs have several unique features. First, although CUDs do build up from a known voting rule, they work directly with the set of possible winners (i.e., alternatives that might be chosen by the majority), and behave much like non-myopic games based on local-dominance (see, e.g., [Meir *et al.*, 2014; Lev *et al.*, 2015; Meir, 2015]). On the other hand, the distinction between lazy and proactive voter behaviour links CUDs with biased voting (see, e.g., [Elkind *et al.*, 2015]). The concept of a *default alternative* (e.g., a mistrial) is also similar to the way lazy-biased voters act: an alternative outside the candidate set is introduced, namely an abstention. However, unlike the abstention of lazy-biased voters, a default alternative is enforced on all voters in circumstances that encompass the entire voter set. Another feature that distinguishes CUDs from other iterative voting processes is a *deadline timeout*. The standard assumption in the iterative voting literature is that voting processes do not stop, unless no voter has a way to further manipulate the outcome. Convergence is the subject of an extensive research effort, both to determine when these processes stop and with what ballot profile [Meir *et al.*, 2010; Reijngoud and Endriss, 2012; Reyhani and Wilson, 2012; Obraztsova *et al.*, 2015]. CUDs, on the other hand, always stop, as we will show in Theorem 1.

To the best of our knowledge, an iterative voting procedure, where voters must vote for certain well-defined alternatives, and where a deadline exists as a restriction of the number of voting stages, has not been examined. This paper is a first attempt at analysing theoretical features of CUDs in iterative voting and to demonstrate them experimentally, with a focus on the convergence rate and the quality of the final decision.

## 3 Model

In this section we formally model an attempt to reach a decision under a deadline based on a bounded iterative voting process. If the iterative process converges to an alternative, that alternative is called the *winner*. We concentrate on an extension of the Majority rule, i.e., on the plurality voting rule with a threshold. However, the model can be further extended to more general voting rules. The threshold can be tight, requiring the decision to be unanimous, or relaxed, requiring a certain threshold of the voters to agree on the decision. The formal details of our model are as follows.

Let  $V$  be a set of  $n$  voters,  $C$  a set of  $m$  alternatives, and  $\tau$  the number of discrete time slices before deadline. We assume that the process starts at time  $t = \tau$  and finishes at deadline  $t = 0$ , that is, time is decreasing and the sequence of the stages is  $(\tau, \tau - 1, \dots, t + 1, t, t - 1, \dots, 1, 0)$ . Each voter is characterised by a *truthful preference*  $a_i \in \mathcal{L}(C)$ , where  $\mathcal{L}(C)$  is the space of complete and non-reflexive orderings over the set of alternatives  $C$ . We write  $a_i(c, c')$  if voter  $i$  prefers  $c$  to  $c'$ . At the beginning of the process, every voter  $i \in V$  casts a ballot  $b_i^t \in C$  that reveals her most preferred candidate at time  $t = \tau$ . The ballots of all voters are collected, forming a *ballot profile*,  $\mathbf{b}^t = (b_1^t, \dots, b_n^t)$ . The *score* of each candidate within a given ballot profile is the number of the voters that voted for this candidate:  $\text{sc}_c(\mathbf{b}^t) = |\{i | b_i^t = c\}|$ . A *score vector* is a collection of scores of all candidates  $\text{sc}(\mathbf{b}^t) = (\text{sc}_1(\mathbf{b}^t), \dots, \text{sc}_m(\mathbf{b}^t))$ . For convenience, a shorthand  $\mathbf{s}^t = (s_1^t, \dots, s_m^t) = \text{sc}(\mathbf{b}^t)$  is used, and we omit the time superscript to denote an arbitrary score vector. The collection of score vectors is public information, and is visible to all the voters. Using the score vector, the possible winners at time  $t$  can be computed using a *Possible Winner Function (PWF)*. Our notion of PWF extends the classical concept of possible winners (that refers to expansions of partial preference ballots, see, e.g., [Xia and Conitzer, 2011]) to iterative ballot modifications. Specifically, PWFs capture the possibility that some sequence of ballot changes will make a candidate the final winner of the voting process. More formally, a Possible Winner Function, denoted  $\mathcal{F}$ , maps a score vector at time  $t$  to a set of *possible winners*, and has the form  $\mathcal{F} : \Delta \rightarrow 2^C$ , where  $\Delta$  is the space of all possible score vectors.

In this paper we investigate two PWFs: the *Iterative Majority (IMaj)*, and its special sub-case *Iterative Unanimity (IUn)*. The possible winner function of IMaj is defined by:

$$\mathcal{F}^{\text{IMaj}}(\mathbf{s}, t) = \{c \in C | \sigma - s_c < t + 1\}.$$

That is, the difference between the score that the alternative needs in order to win, and the alternative's score in the next step, is bounded by time. Once the deadline is reached,  $\mathcal{F}(\mathbf{s}, 0)$  is either (i) a singleton, containing the candidate with the highest score (the winner), or (ii) empty, signifying that the default candidate must be adopted.<sup>1</sup>

The majority threshold is  $\sigma > \frac{n}{2}$  and the unanimous threshold is  $\sigma = n$ . Hence, the possible winner function for IUn is generated by setting  $\sigma = n$ , i.e.,  $\mathcal{F}^{\text{IUn}} = \mathcal{F}_n^{\text{IMaj}}$ .

Once the set of possible winners are computed, at any given time  $t \in [0 : \tau]$ , the voter decides whether to change her vote and produce a ballot  $b_i^t \in C$ , i.e., state her vote at time  $t$ . Notice that the voter's decision is based on the best possible winner for that voter, and on the voter's utility function.

**The voter's best possible winner:** For any  $W \subset C$  the best alternative in  $W$  w.r.t. the voter's truthful preferences  $a_i$  is denoted  $\text{top}_i(W) \in W$ . That is,  $w$  is the best possible winner for voter  $i$  if voter  $i$  prefers it over any other candidate in the possible winner set:  $w = \text{top}_i(W)$  iff for all  $c \in W \setminus \{w\}$  holds  $a_i(w, c)$ .

<sup>1</sup>With selfish utility optimisation (see below), this will entail time dependent decision making.

**The voter's utility function:** Each voter has a utility function that matches a score vector at time  $t$  to the voter's utility:  $u_i : \Delta \times [0 : \tau] \rightarrow \mathbb{R}$ . We assume that the utility function is consistent with the voter's truthful preferences  $a_i$  for any  $t \in \tau$ . We will particularly emphasise the classes of *lazy consistent* and *proactive consistent* utility functions. Intuitively, a lazy consistent utility function drives the voter to change her ballot only if a better alternative can be injected into the set of possible winners, while a proactive consistent function would also induce ballot change if the score of the best possible winner can be improved. Formally, these utility classes are defined as follows.

**Definition 1.** A utility function  $u_i$  is lazy consistent if for all  $\mathbf{s}, \mathbf{s}' \in \Delta$  and  $t \in [0 : \tau]$  the following condition holds:

$$u_i(\mathbf{s}, t) > u_i(\mathbf{s}', t) \iff a_i(w, w'),$$

where  $w = \text{top}_i(\mathcal{F}(\mathbf{s}, t))$  and  $w' = \text{top}_i(\mathcal{F}(\mathbf{s}', t))$ .

**Definition 2.** A utility function  $u_i$  is proactive consistent if for all  $\mathbf{s}, \mathbf{s}' \in \Delta$  and  $t \in [0 : \tau]$  the following condition holds:

$$u_i(\mathbf{s}, t) > u_i(\mathbf{s}', t) \iff a_i(w, w') \vee ((w = w') \wedge (s_w > s'_w))$$

where  $w = \text{top}_i(\mathcal{F}(\mathbf{s}, t))$  and  $w' = \text{top}_i(\mathcal{F}(\mathbf{s}', t))$ .

We assume that all voters are selfish and myopic, and seek to maximise their utility function  $u_i(\mathbf{s}^t, t)$ . We further assume that all utility functions are homogeneously either lazy consistent or proactive consistent. Notice that, although here we only use  $\mathcal{F}^{\text{IMaj}}$  and  $\mathcal{F}^{\text{IUn}}$ , CUDs can be naturally extended to more general forms of PWFs.

The algorithm for a CUD iterative voting game is given in Algorithm 1. Since score vectors simply accumulate ballots from  $\mathbf{b}^t$ , we use algebraic operations between a score vector and a ballot. That is, if  $\mathbf{s}' = \mathbf{s} - c$  (respectively,  $\mathbf{s} = \mathbf{s}' + c$ ) then  $s'_k = s_k$  for all  $k \in C \setminus \{c\}$  and  $s'_c = s_c - 1$  (respectively,  $s'_c = s_c + 1$ ).

The algorithm receives as input a Possible Winner Function  $\mathcal{F}$ , the initial time until the deadline  $\tau$ . Iteratively, as long as the deadline has not been reached: all voters calculate the current score vector (line 3). If there is only one possible winner,  $w$ , a decision has been reached and the game ends (lines 4-5). The game also ends if there are no possible winners (lines 7-8). If the game continues, every voter calculates what is her best possible winner, given the current score vector (line 11). If there are ties (i.e., if a few alternatives receive the same score), the voter selects the alternative that is ranked highest in her truthful preferences  $a_i$ . Each voter decides if she wants to vote, the decision being based on her current possible winners and on her utility function. Voters who want to vote "raise their hands", i.e., are collected into a set  $I$  (line 13). A random voter is chosen from set  $I$  (line 14). The chosen voter casts her ballot (lines 15-18) and we are now one step closer to the deadline (line 19).

In order to analyse the quality of the result of a CUD game, features of voting processes can be adapted. One such feature is the *Additive Price of Anarchy (PoA)* [Branzei et al., 2013]. We adapt the additive PoA to CUDs as follows.

**Definition 3.** Let  $\mathbf{a}$  be the truthful profile of voters participating in a CUD,  $\mathbf{b}$  a ballot profile consistent with  $\mathbf{a}$  (i.e.,

---

**Algorithm 1** Consensus Under Deadline: Game Progress

---

**Input:** PWF  $\mathcal{F}$ , deadline timeout  $\tau$   
**Input:** Set of voters  $V$ , set of alternatives  $C$ ,  
**Input:** Truthful profile  $\mathbf{a}$ , and utilities  $u_i$ .  
**Initialise:** Set  $t \leftarrow \tau$ , and  $b_i^\tau \leftarrow \mathbf{top}_i(C)$  for all  $i \in V$

- 1: **while**  $t \geq 0$  **do**
- 2:     Ballots  $b_i^t$  are declared
- 3:     All voters calculate  $\mathbf{s}^t = \mathbf{sc}(\mathbf{b}^t)$
- 4:     **if**  $\mathcal{F}(\mathbf{s}^t, t) = \{w\}$  for some  $w \in C$  **then**
- 5:         **return**  $w$  as the winner ▷ Game stops
- 6:     **end if**
- 7:     **if**  $\mathcal{F}(\mathbf{s}^t, t) = \emptyset$  **then**
- 8:         **return** *mistrial* ▷ Game stops
- 9:     **end if**
- 10:    **for**  $i \in V$  **do**
- 11:        $w_i \leftarrow \arg \max_{c \in C} u_i(\mathbf{s}^t - b_i^t + c, t - 1)$  ▷ ties are  
      broken by  $a_i$
- 12:     **end for**
- 13:      $I \leftarrow \{i \in V \mid w_i \neq b_i^t\}$
- 14:      $j \leftarrow \text{Random}(I)$  ▷ random voter choice
- 15:     **for**  $i \in V$  **do**
- 16:        $b_i^{t-1} \leftarrow b_i^t$
- 17:     **end for**
- 18:      $b_j^{t-1} = w_j$  ▷ Only  $j$  is allowed to revote
- 19:      $t \leftarrow t - 1$
- 20: **end while**

---

$b_i = \mathbf{top}_i(C)$ , and  $\mathbf{s} = \mathbf{sc}(\mathbf{b})$ . Denote all candidates that the CUD may converge to by  $\hat{C}$ . Then the CUD's additive Price of Anarchy is

$$PoA^+(\mathbf{a}) = \max_{c \in \hat{C}} s_c - \min_{c \in \hat{C}} s_c$$

Namely, Additive PoA is the score of the least preferred alternative that could become the winner of a CUD, subtracted from the score of the truthful winner.

### 3.1 Theoretical Features of CUD

From the first look at Algorithm 1 it may appear that it is incomplete, since there is no default decision after the main loop ends. The following theorem shows that this eventuality never occurs. That is, the loop never completes, but is always broken by one of the game's stopping conditions (Line 5 and Line 8). Notice that Theorem 1 applies for both types of voters, lazy and proactive. In fact, all of our theoretical results are applicable to both of these voter types.

**Theorem 1.** For  $\mathcal{F}_\sigma^{IMaj}$ , for any  $\sigma \in (\frac{n}{2}, n]$ , and for all consistent utility functions, CUD either stops at  $t = \tau$  with a mistrial, or at a time  $t \in [0 : \tau]$  with a valid alternative declared the winner.

In light of Theorem 1, we will slightly overload the concept of *convergence*. Convergence in general iterative voting schemes means that the game stops at some stable point. For CUDs, on the other hand, we will say that it converges if the game stops by declaring some  $w \in C$  the winner. If the game stopped with a mistrial, we will say that the CUD did not

converge.<sup>2</sup>

Now, having established that the algorithm always stops, we can place a condition on game features that ensure that it stops with a valid alternative, rather than a mistrial.

**Theorem 2.** Let  $\mathbf{a} = (a_1, \dots, a_n)$  be the truthful profile, let  $\tau$  be the deadline time, and let  $\mathbf{b}$  be the ballot profile induced by  $\mathbf{a}$ , i.e.  $b_i = \mathbf{top}_i(C)$ . CUD stops with some  $w \in C$  if and only if there is an alternative  $c \in C$  so that  $\mathbf{sc}_c(\mathbf{b}) \geq \sigma - \tau$ .

Theorem 2 essentially provides a finer bound on what the initial scores must look like, so that CUD converges. Intuitively, the theorem says that for an alternative to become the declared winner, there must be enough time for it to gather additional support to achieve the majority threshold. However, Theorem 2 does not guarantee that a particular alternative will be declared a winner. For such a guarantee, a much more stringent condition must be required of  $\tau$  and  $n$ , as the following theorem states.

**Theorem 3.** Let  $\mathbf{a} = (a_1, \dots, a_n)$  be the truthful profile, let  $\tau$  be the deadline time, and let  $\mathbf{b}$  be the ballot profile induced by  $\mathbf{a}$ , i.e.,  $b_i = \mathbf{top}_i(C)$ . If there is an alternative  $c \in C$  so that  $\mathbf{sc}_c(\mathbf{b}) \geq \max\{\lfloor \frac{n}{2} \rfloor + 1, \sigma - \tau\}$  then CUD stops with  $c$  as the winner.

The following example demonstrates that the bound of Theorem 3 is tight. That is, if the score was any lower than Theorem 3 suggests, then, for at least some truthful profiles, a CUD would have more than one alternative that can be declared the winner.

**Example 1.** Let the number of voters  $n$  be even, and the number of alternatives  $m \geq 2$ . Furthermore, assume that  $\tau > \frac{n}{2}$ . Let  $\mathbf{a}$  be such that for  $i \in [1 : \frac{n}{2}]$ ,  $\mathbf{top}_i(C) = c_1$  holds, and for  $i \in [\frac{n}{2} + 1 : n]$ ,  $\mathbf{top}_i(C) = c_2$  holds. All other candidates may appear in  $a_i$  in any order. Then, both  $c_1$  and  $c_2$  can possibly be declared as the winner in a CUD.

Having dealt with the characterisation of alternatives that can be declared a winner, and those that are guaranteed to become a winner, we can exploit this knowledge to place some bounds on the additive Price of Anarchy for CUDs.

**Theorem 4.** Let  $\mathbf{a}$  be the truthful profile of voters participating in a CUD. The following bounds can be placed on the additive Price of Anarchy,  $PoA^+$ , depending on the ratio of the deadline timeout  $\tau$  and the number of voters  $n$ :

- If  $\tau \leq \sigma + \lfloor \frac{n}{2} \rfloor$ , then

$$PoA^+(\mathbf{a}) = 0. \tag{1}$$

- If  $\sigma - \lfloor \frac{n}{2} \rfloor < \tau < \sigma$ , then

$$PoA^+(\mathbf{a}) \leq \lfloor \frac{n}{2} \rfloor + \tau - \sigma. \tag{2}$$

- If  $\tau \geq \sigma$ , then

$$PoA^+(\mathbf{a}) \leq \lfloor \frac{n}{2} \rfloor - 1. \tag{3}$$

---

<sup>2</sup>Intuitively, a mistrial represents a deadlock or a livelock, similar to the infinite running time of non-converging iterative voting models in the standard sense.

**Lemma 1.** *The last two bounds in Theorem 4 are tight. For all  $\tau$  and  $n$  that satisfy the conditions of Equations 2 and 3, there exists a truthful profile  $\mathbf{a}$  such that the corresponding bound holds as an equality.*

## 4 Experimental Features of CUD

We evaluated the behaviour of lazy and proactive voters on four real world data sets: the Sushi data set (5000 voters, 10 candidates) [Kamishima *et al.*, 2005], the T-shirt data set (30 voters, 11 candidates), the courses 2003 data set (146 voters, 8 candidates) and the courses 2004 data set (153 voter, 7 candidates). The three latter data sets are taken from the Preflib library [Mattei and Walsh, 2013]. For a fixed number of  $n$  ( $n = 10, 20$  or  $30$ ) voters, we varied the time left until the deadline:  $t \in [1, \tau]$ . For each experimental setting, we created 20 random sets of voter profiles by sampling with return from each data set. For each set of voter profiles, the experiment was conducted 30,000 times.

We examined: (1) The convergence rate, i.e., the ratio of games that converge to the total number of games within a given sub-class (e.g. games with 10 voters, or 7 candidates); (2) How many changes in votes are required for the process to converge; and (3) The Additive Price of Anarchy for a process that has converged (Definition 3).

Our theoretical results are relevant for all majority thresholds  $\sigma \in (\frac{n}{2}, n]$ . However, to investigate the finer features of CUDs, we fix this parameter as  $n$ , the number of voters. Even though it means that we experimentally study an extreme CUD case, i.e., unanimity, fixing  $\sigma$  allows us to exclude it as a free parameter and better concentrate on studying complex game features such as the Additive Price of Anarchy.

In order to conclude which voter type performs best over multiple data sets, we followed a robust non-parametric procedure proposed by [García *et al.*, 2010]. This procedure allows us to drop the assumption that the differences between the voter types are normally distributed, and is thus more adequate than a t-test. We first used the Friedman Aligned Ranks test in order to reject the null hypothesis that all heuristics perform the same. This was followed by the Bonferroni-Dunn test to find whether one of the heuristics performs significantly better than other heuristics.

### 4.1 Results

**Convergence:** As Theorem 2 indicates, convergence always occurs when there is enough time until the deadline to allow voters to change their vote (i.e., when the number of iterations is larger than the number of voters). The experiments reveal that there is no difference in the convergence rate between lazy and proactive voters. Interestingly, we find that in all experimental settings the process converges even when the time until deadline is somewhat smaller than the number of voters, i.e., the convergence rate (estimated from 30K processes for each experimental setting) reaches 1 with *less* time than is theoretically necessary. Table 1 shows the time left until the deadline when the convergence rate equals 1 (i.e., when all experiments converge). Each row represents a different number of voters, and each column a different data set. The number of candidates in the data set is indicated in brackets.

Table 1: All processes’ convergence time

Number of voters	Courses 2004 (7)	Courses 2003 (8)	Sushi (10)	T-shirts (11)
10	6	7	8	8
20	13	14	14	14
30	19	23	24	23

Table 2: Number of vote changes

Datasets	10 voters		20 voters		30 voters	
	lazy	proactive	lazy	proactive	lazy	proactive
Courses 2003	3.98	5.17	6.79	7.86	9.05	10.24
Courses 2004	3.55	5.73	6.38	9.51	8.89	12.71
Sushi	4	4.73	6.66	7.45	8.41	9.35
T-shirts	3.96	4.84	6.19	6.88	7.87	8.57

For example, for the courses 2004 data set (column 2), for 10 voters (row 2), all of the experiments converge when the initial time before the deadline is  $\tau \geq 6$ . The process seems to begin to converge faster when there are fewer candidates. The exact impact of the candidate number on convergence should be examined on a wider variety of data sets; we leave this for future research. The more voters, the longer it takes the process to converge. This is illustrated in Figure 1, on the courses 2004 data set (similar results were obtained for the other data sets).

**Required number of vote changes:** Table 2 shows, for different data sets and varying number of voters, the normalised average of vote changes required to reach consensus. According to the Friedman test, there is a significant difference between the number of vote changes performed when either all voters are lazy or all of them are proactive. According to the Bonferroni-Dunn test, proactive voters require a significantly higher number of vote changes in order to reach consensus. Notice that, regardless of the difference in the number of required vote changes, the converge rate for both voter types is the same. That is, proactive voters do not require more rounds to converge, but simply begin changing their votes sooner, further away from the deadline.

**Additive price of anarchy:** The Additive Price of Anarchy was computed as the plurality score of the least preferred candidate that was elected to be a unanimous winner in one of the 30,000 experiments, subtracted from the plurality score of the truthful winner. For example, consider 12 voters and 4 candidates and the following scores at the beginning of the process:  $c_1 = 2, c_2 = 6, c_3 = 1, c_4 = 3$ . The truthful plurality winner is  $c_2$  with a score of 6. If in one of the experiments  $c_3$  is the unanimous winner, the additive price of anarchy is 5. If  $c_3$  does not win in any of the experiments, but in some of them  $c_1$  wins, the price is 4. Table 3 shows the normalised average of the additive price of anarchy. Although one can notice a trend in the data, and see that for 20 and 30 voters the additive price of anarchy is somewhat higher for lazy voters, we could not confirm a significant difference using the Friedman test. Less accurate measures such as a simple t-test reveal that the additive price of anarchy is significantly higher for lazy voters, in the case of 30 voters. Unfortunately, overall, statistical analysis was inconclusive, and no significance test could decide the issue completely.

Table 3: Additive Price of Anarchy

Datasets	10 voters		20 voters		30 voters	
	lazy	proactive	lazy	proactive	lazy	proactive
Courses 2003	0.21	0.23	0.52	0.38	0.6	0.33
Courses 2004	0.03	0.03	0.02	0.02	0.18	0.18
Sushi	0.15	0.15	0.29	0.27	0.12	0.12
T-shirts	0.69	0.27	0.38	0.35	0.51	0.3

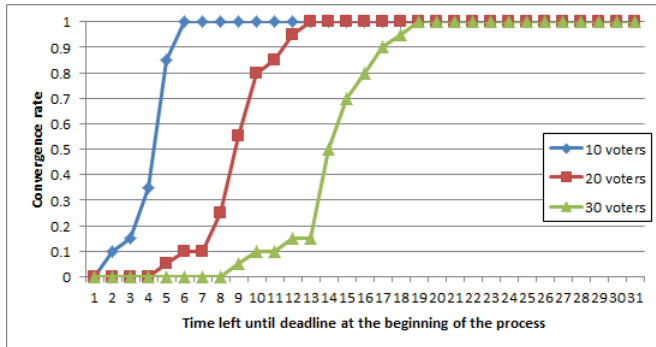


Figure 1: Voting process convergence rate.

To conclude, it is interesting to note that although the convergence rate is equal for both voter types, the required number of vote changes is higher for the proactive voters than for the lazy voters, whereas the additive price of anarchy seems to be lower for the proactive voters than for the lazy voters. This may indicate that there is a tradeoff between the accuracy of the results and the required voter effort, at least when the number of voters is relatively large.

## 5 Conclusions

When a group of individuals with different preferences is asked to agree on a certain alternative, it would be natural to expect that in some cases the consensus will be not reached, even if the deadline is moderate or very far. Surprisingly, our model of an iterative voting process with time restriction (CUD) predicts that if there is a possibility to converge, then a consensus will be reached.

If there is at least one candidate for whom there is enough time to gain the missing votes (in other words, a possible winner), then the process converges with such a candidate chosen. Furthermore, if there is a candidate who, a priori, is the top choice of the majority of voters, the process converges to that very candidate, still subject to a sufficiently long deadline timeout. These results remain valid even for the strictest special case of our model, Unanimity, and are confirmed (for sanity) by our experiments.

It is obvious that not all individuals behave identically. We define two types of voters: proactive and lazy, according to their behaviour. Proactive voters are, in a sense, trigger-happy to change their vote, even if just to ensure that their preferred possible winner gets one more point. Lazy voters change their votes only when it is necessary to do so, i.e., when their vote is pivotal to keep a particular alternative as a possible winner.

It would be natural to interpret proactive voters as those

that actively seek consensus. This, however, is incorrect. Our experiments show that the convergence rates of both proactive and lazy voter CUDs are the same. On the other hand, the number of vote changes until convergence is higher for proactive voters. In a way, they are inefficient in their behaviour. However, there is a benefit to the proactive voters' activism.

Our experiments looked deeper into the Additive Price of Anarchy (PoA) as a measure of winner quality. Theoretical results, while showing PoA principal bounds, do not provide specific trade-offs. On the other hand, while re-confirming our bounds experimentally, our experiments indicate that there might be an interesting trade-off with regard to PoA. Namely, the final winner is closer to the truthful plurality winner (has lower PoA) for proactive voters, than for lazy voters.

### 5.1 Future Work

Currently, our model assumes that all voters have the same type; either all are lazy, or all are proactive. However, given the possible tradeoff with PoA found in experiments, it would be interesting to analyse the behaviour of mixed voter populations, e.g., how the ratio of proactive and lazy voters affects the balance between the number of re-votes and PoA. This could also refine our data set sufficiently for statistical tests to unequivocally determine the significance of this tradeoff. Furthermore, the two voter types are an initial foray. It's more feasible now to construct other varieties of voters.

We plan to expand our experimental base, and seek additional tradeoffs between various model parameters. For example, we conjecture that the convergence rate is affected by the number of alternatives, and we plan to investigate whether the convergence rate is subject to a tradeoff between the number of alternatives, the number of voters, and the deadline timeout.

A more challenging extension, however, would be to adapt our model to use a general Positional Scoring Rules (PSRs), rather than a simple Majority. E.g. veto, approval, and Borda. These rules would allow us to express complex semantic structures over the set of alternatives, still motivated by the jury trial example. For example, in some cases the jurors are asked not only to state "guilty" – "not-guilty", but also to define the amount of damages or penalties in civil trials, or the recommendations for sentencing in criminal trials. There are many disagreements and discussions about the death penalty. Thus, some jurors may wish to choose guilty, but to veto the death penalty, while others would approve both life imprisonment and the death penalty. In these situations, veto and approval voting are far more appropriate than Majority.

Stepping even further away from our basic model, we need to investigate what information about the current vote affects a CUD's outcome, and in what way. For example, rather than using utilities that depend on an anonymous score vector, we can use weights to express the fact that the opinion of certain voters is more influential. The same technique would allow us to impose a price on the number of times a voter changes her mind, e.g. her vote loses influence, as being unstable. Simultaneous re-voting will be another research direction, being a very non-trivial modification.

## References

- [Branzei *et al.*, 2013] S. Branzei, I. Caragiannis, J. Morgenstern, and A. D. Procaccia. How bad is selfish voting? In *AAAI*, pages 138–144, 2013.
- [Cramton and Tracy, 1992] P. C. Cramton and J. S. Tracy. Strikes and Holdouts in Wage Bargaining: Theory and Data. *American Economic Review*, 82(1):100–121, March 1992.
- [Elkind *et al.*, 2015] E. Elkind, E. Markakis, S. Obraztsova, and P. Skowron. Equilibria of plurality voting: Lazy and truth-biased voters. In *SAGT*, pages 110–122, 2015.
- [García *et al.*, 2010] S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.*, 180(10):2044–2064, May 2010.
- [Kamishima *et al.*, 2005] T. Kamishima, H. Kazawa, and S. Akaho. Supervised ordering - an empirical survey. In *Proceedings of the 5th IEEE International Conference on Data Mining*, 2005.
- [Kwiek, 2014] M. Kwiek. Conclave. *European Economic Review*, (70):258–275, 2014.
- [Lev *et al.*, 2015] O. Lev, S. Obraztsova, M. Polukarov, Z. Rabinovich, and J. S. Rosenschein. Farsighted voting dynamics. In *AGT WS @ IJCAI*, 2015.
- [Lim and Murnighan, 1994] S. G.-S. Lim and J. K. Murnighan. Phases, deadlines, and the bargaining process. *Organizational Behavior and Human Decision Processes*, 58(2):153 – 171, 1994.
- [Ma and Manove, 1993] C.-T. A. Ma and M. Manove. Bargaining with Deadlines and Imperfect Player Control. *Econometrica*, 61(6):1313–39, November 1993.
- [Mattei and Walsh, 2013] N. Mattei and T. Walsh. Preflib: A library of preference data [HTTP://PREFLIB.ORG](http://PREFLIB.ORG). In *Proc 3rd Conf. on Algorithmic Decision Theory*, 2013.
- [Meir *et al.*, 2010] R. Meir, M. Polukarov, J. S. Rosenschein, and N. R. Jennings. Convergence to equilibria of plurality voting. In *AAAI*, pages 823–828, 2010.
- [Meir *et al.*, 2014] R. Meir, O. Lev, and J. S. Rosenschein. A local-dominance theory of voting equilibria. In *ACM EC*, pages 313–330, 2014.
- [Meir, 2015] R. Meir. Plurality voting under uncertainty. In *AAAI*, pages 2103–2109, 2015.
- [Moore, 2004] D. A. Moore. The unexpected benefits of final deadlines in negotiation. *Journal of Experimental Social Psychology*, 40(1):121 – 127, 2004.
- [Naamani-Dery *et al.*, 2015] L. Naamani-Dery, S. Obraztsova, Z. Rabinovich, and M. Kalech. Lie on the fly: Manipulative voters under iterative voting centre. In *IJCAI*, pages 2033–2039, 2015.
- [Obraztsova *et al.*, 2015] S. Obraztsova, E. Markakis, M. Polukarov, Z. Rabinovich, and N. R. Jennings. On the convergence of iterative voting: How restrictive should restricted dynamics be? In *AAAI*, pages 993–999, 2015.
- [Reijngoud and Endriss, 2012] A. Reijngoud and U. Endriss. Voter response to iterated poll information. In *AAMAS*, 2012.
- [Reyhani and Wilson, 2012] R. Reyhani and M. Wilson. Best reply dynamics for scoring rules. In *ECAI*, volume 2, pages 672–677, 2012.
- [Roth *et al.*, 1988] A. E. Roth, J. K. Murnighan, and F. Schoumaker. The deadline effect in bargaining : Some experimental evidence. *American Economic Review*, 78(4):806–823, 1988.
- [Xia and Conitzer, 2011] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *JAIR*, 41:25–67, 2011.