Ariel University

Master Thesis Proposal

Consensus Under a Deadline

Author: David Ben Yosef
Supervisor: Dr. Lihi Dery

A thesis proposal submitted in partial fulfillment of the requirements for the degree of Master in the Department of Computer Science

January 17, 2018
Declaration of Authorship

I, David Ben Yosef, hereby declare that this thesis proposal entitled, “Consensus Under a Deadline” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 

________________________________________
Abstract

Voting is a common way to reach a group decision. When possible, voters will attempt to vote strategically, in order to optimize their satisfaction from the outcome. Previous research has modeled how rational voter agents (bots) vote to maximize their personal utility in an iterative voting process that has a deadline (a timeout). However, it remains an open question whether human beings behave rationally when faced with the same settings. The focus of this research is therefore to examine how the deadline factor affects manipulative behavior in real-world scenarios where humans are required to reach a decision before a deadline. An On-line platform was built to enable voting games by all types of users: agents (bots), humans, and mixed games with both humans and agents. We compared the results of human behavior and bot behavior and concluded that it might be wise to allow bots to make (certain) decisions on our behalf.
Acknowledgements

I would like to thank the group I worked with on this research: Svetlana Obraztsova and Zinovi Rabinovich (Nanyang Technological University, Singapore) Marina Ban-nikova (Universitat Autònoma de Barcelona, Spain) And last but not least, my su-pervisor Lihi Naamani-Dery.
Contents

Declaration of Authorship                      ii

Abstract                                          iii

Acknowledgements                                  iv

1 Introduction                                    1
  1.1 Contributions                                  2
  1.2 Work Structure                                  2

2 Related Work                                    3

3 Model                                           5

4 The CUD Framework                               7
  4.1 CUD-Game flow                                  7
  4.2 CUD-Game admin                                 11
  4.3 Reward method                                  11

5 Experimental Evaluation                       13
  5.1 Irrational voters                              13
  5.2 Evaluation Metrics                            14
  5.3 Data collection                                15
  5.4 Results Analysis                              16

6 Discussion                                      19

A Poster presentation for the IAAI16@Technion     23

B Paper presentation for the WIC17@Leipzig        25
List of Figures

4.1 CUD-Game System Diagram ............................................. 8
4.2 CUD-Game Flow Diagram ................................................ 8
4.3 Instructions screen ........................................................ 8
4.4 Login screen ............................................................... 9
4.5 Waiting for the game to start ............................................. 9
4.6 Game running ............................................................... 9
4.7 Game finished ............................................................. 10
4.8 Help chat ................................................................. 10
4.9 Administration screen ..................................................... 11
5.1 IRR1 behavior scenario ................................................... 14
5.2 IRR2 behavior scenario ................................................... 14
5.3 Both IRR1 and IRR2 behavior scenario ............................... 14
5.4 Convergence percentage comparison ................................. 16
5.5 Average reward points comparison ................................... 17
5.6 POA comparison .......................................................... 17
5.7 Convergence time comparison ......................................... 18
Chapter 1

Introduction

Reaching a consensus within a group is a hard task. Decisions may range from choosing the next location for a family’s winter vacation, to a jury’s decision on the outcome of a trial case. In both cases, a deadline for reaching the decision exists. Family members must reach a decision an adequate amount of time before the vacation date. A jury is required to reach a unanimous decision within a reasonable time limit, or else the defendant is acquitted or a new trial is held.

Usually, a decision is not reached immediately, but rather a sequential voting process takes place, where voters change some of their preferences as the deadline approaches. For example, consider Bob, a family member whose preferences for a family vacation are: Hawaii, then Greece and lastly Spain. The rest of Bob’s family members prefer either Spain or Greece. Bob might first try and convince the family of the merits of a vacation in Hawaii. But as the deadline approaches, Bob might change his preferences and report Greece as his most preferred option, in order to ensure that Spain will not be debated for consensus.

Voters may attempt to vote strategically[8], in order to optimize their satisfaction from the outcome (in the example above, Bob’s change of preferences, from Hawaii to Greece, is a strategic move). In a previous study, a model for the behavior of strategic agents in a time-bounded iterative voting process has been proposed, providing convergence guarantees and an analysis of the quality of the final decision [2]. However, people are not always rational in their choices.

In this research, we introduce the CUD game, an on-line voting system game where a group of people (voters) is required to reach a joint decision. The voters’ ultimate goal is to reach a unanimous winner - a decision which is agreed at the end by all voters, and to reach this winner as fast as possible, before the deadline. The voters’ gained utility depends on their performance. At the beginning of the game, each voter receives a set of ranked preferences, and then in an iterative process, voters may change their vote. As in the theoretical model [2], we rely on the following assumptions: the voting iteration is singular (one voter per iteration), and we fixed the number of alternatives and voters. With these assumption we simulate common voting structures where voters need to send email, submit request, or any other “hand raising” option, but not all speak at once. It helps us to examine voter’s discrete actions, since simultaneous vote changes increase the complexity of analyzing the voter’s actions.

We collected data that allows us to analyze human behavior: How often is a consensus reached? When does the process of strategic behavior begin (if at all)? When and if do people change their vote? Lastly, we examined the quality of the final decision reached by a group of people, in terms of the average reward and the price of anarchy. We compared our results with the performance of rational bots, and a mixture of bots and humans.
Chapter 1. Introduction

1.1 Contributions

This is a novel attempt to examine strategic voting for a consensus under a deadline, we dropped the rational voter behavior assumption and examine how real users behave:

- We built an on-line system to support the case study.
- We developed stand-alone voting Bots, with internal support of different voting algorithms.
- We ran the case study and identified weaknesses in the model.
- We improved the model accordingly and ran another case study.
- We collected the data, and analyzed it with a custom dedicated tool
- We reached some key insights as to bot vs. human voting:
  - Bots are better voters - they reach the highest benchmarks in all metrics
  - Humans tend to converge faster than needed - probably because they are afraid of others irrationality
  - Any irrational action is bad for the voter, we didn’t find any good reason for the human’s irrationality

1.2 Work Structure

This work is organized as follows: First, we review the related work of our research background (Chapter 2). Then, we describe the model used for our research (Chapter 3). Next, we present the CUD framework we developed for this research (Chapter 4), the experiments we ran with it and we analyze the results (Chapter 5). Lastly, we finalize with conclusions and discussion (Chapter 6).
Chapter 2

Related Work

Social Choice is not a new field. In fact, there is written evidence of research in this field since ancient Rome around the 1st century A.D. [17].

Social choice theory as a mathematical scientific discipline started only during the middle of the previous century with the publications of J. Arrow [1] and other theorems where the basic formal properties had been defined. Computational social choice is the field where we combine the Social choice theory investigation with computer science notations and methodologies [3].

Strategic voting is one of the main focuses of the social choice research. Strategic voting usually happen when voters believe that their most preferred option is not going to win, so they declare different selection than their truthful preference, which according to their preference set is still better than other option. The result of strategic voting is hard to predict, and might change the results dramatically [8, 7, 25]. In some cases, the strategic behavior can lead to a worse outcome for everyone than declaring truthful ballots (votes) [29], and it leads to the model of safe (and unsafe) strategic voting [27, 12] which defines the scenarios where we can avoid ending up with a worse outcome than voting sincerely.

Meir et al. [19] were one of the first attempts to identify when the elections converge to equilibrium while allowing the voters to change their vote. They defined the model of iterative voting, in order to study how a single vote change influences the score vector every iteration, and when this process converges.

Bannikova et al. [2] analyzed theoretical features of 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. We followed their model but focused on human behavior, our goal being to examine whether humans behave rationally (as in the model).

A feature that distinguishes our framework from other iterative voting processes is the deadline. Informally, the deadline is a time limit on the voting process. A deadline factor has been studied from various angles by the classical game theory [16, 15, 20], but as far as we know, it has never been done for the computational social choice.

In the iterative voting process (e.g. [19, 6, 24, 14, 11, 21, 22, 23, 13, 18]) it is a question whether or not the process will converge. In all these papers, many scoring rules and voting methods are researched, some of them might never converge. Our framework will always converge if the voters are rational and there is a sufficient amount of time before the deadline [2].

Reijngoud et al. [24] analyzed the iterated voting model from different angles: how the amount of information received in one single poll influences the manipulated behavior of the voter, and how the repeated polls affect the manipulated behavior of all voters. Brânzei et al. [4] studies the different influence of strategic voting on the main voting methods. They defined the Price Of Anarchy (POA) factor
that measures the differences between truthful voting and worse-case manipulated voting. POA is one of the metrics we used in the experimental evaluation.

On-line platforms such as Doodle (www.doodle.com) allow people to submit their votes on the problem in question, to view the votes of other group members, and to change their votes if they wish. Zou et al. [30] studied the strategic voting behavior on Doodle polls, and how the knowledge of current voters preferences affected the results. Interestingly, people tend to change their vote preference in light of the votes of their peers. In our model the voter preferences are kept private and voters can see only the intermediate aggregated results.

Experimental studies of bargaining deadlines have shown that the majority of agreements are obtained in the final seconds before the deadline; contrary to what might be expected, even complete information does not speed agreement much [26]. As opposed to bargaining models [5, 9, 10], in our suggested framework voters must agree on one alternative and no compromise between alternatives is possible.

Tal et al. [28] performed a study of human behavior on online voting. They classified voters into 3 distinct types, two of which are not strategic and one that will perform straight forward strategic moves. However their settings do not contain a deadline. Moreover, while they provide valuable insights on human voting patterns, they do not compare their framework with a rational strategic model, so the question of whether humans and bots act alike remains unanswered.
Chapter 3

Model

We followed the rational agents’ behavior model suggested by Bannikova et al. [2] the model introduces an attempt to reach a decision under a deadline based on a bounded iterative voting process. We concentrated on the plurality voting rule, where each voter casts a single vote for his most preferred alternative. The alternative that receives the most votes wins.

Let $V$ be a set of $n$ voters, $C$ a set of $m$ alternatives, and $\tau$ the number of discrete time slices before the deadline. We denote two types of voters: rational automated voters (bots) $v$, and human voters $\hat{v}$. Each voter is characterized by a truthful preference $A_i, \hat{A}_i \in L(C)$; a complete and non-reflexive order over the set of alternatives. We write $c \succ_i c'$ if voter $i$ prefers $c$ to $c'$. At the beginning of the process, every voter $i \in V$ casts a ballot $b_t^i \in C$ that reveals their most preferred candidate at time $t = \tau$. The ballots of all voters are collected, forming a ballot profile, $b^t = (b^t_1, \ldots, b^t_n)$.

The score of each candidate within a given ballot profile is the sum of the voters that voted for this candidate:

$$sc_c(b^t) = |\{i | b^t_i = c\}|.$$

A score vector is a collection of scores of all candidates $sc(b^t) = (sc_1(b^t), \ldots, sc_m(b^t))$.

The collection of score vectors is public information, and is visible to all the voters.

Once the collection of score vectors are published, each voter decides whether to change her vote and produce a ballot $b^t_i \in C$, i.e., state her (new) vote at time $t$.

On each round, one voter is selected randomly from all requested changes, and only his change is updated in the score vector.

If the iterative process converges to an alternative, that alternative is called the winner. A Possible winner candidate is a candidate which still has enough time to become a winner if all voters will ask to change to it. More formally:

**Definition 3.0.1.** A candidate is a possible winner PW if the number of required vote changes is less than number of remaining rounds. The Required vote changes for a candidate is the current candidate score subtracted from the number of voters. $PW_c = n - sc_c(b^t)$

For the bots, we followed the design of proactive voters, as defined in [2] as follows: A proactive voter will always seek to change his vote and improve the outcome even if his current selection is a PW (While a lazy voter will change his selection only if by not changing his selection he will cause a mistrial). The behavior of our proactive bots described below in Algorithm 1. Namely, if the bot’s current selection is a possible winner (line 1), do not ask to change the vote (line 2). If the bot’s current selection is not a possible winner (line 3) then check the next candidate in the bot’s ordered set of preference (line 4). If that candidate is a possible winner (line 5), the bot will ask to change its vote to this candidate (line 6). If not, continue to the candidate ranked next.

Using this algorithm, a bot will always maximize its outcome since it always selects the most highly ranked possible winner. When all voters play with this logic,
Chapter 3. Model

Algorithm 1 Rational agent logic

<table>
<thead>
<tr>
<th>Input:</th>
<th>Truthful profile $A$, selected candidate $c_i$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input:</td>
<td>Number of voters $v$, deadline timeout $\tau$</td>
</tr>
<tr>
<td>Input:</td>
<td>Current score vector $R$</td>
</tr>
</tbody>
</table>

1: if $c_i$ is a PW in $R$ with $v$ voters on time $\tau$ then
2: \hspace{2em} return $c_i$ as the selected \hspace{2em} $\triangleright$ No Change
3: else
4: \hspace{2em} for $i \in A$ do \hspace{2em} $\triangleright$ From most preferred to least preferred
5: \hspace{4em} if $c_i$ is a PW in $R$ with $v$ voters at time $\tau$ then
6: \hspace{6em} return $c_i$ as the selected
7: \hspace{5em} end if
8: \hspace{4em} end for
9: end if

the game will always converge if the number of rounds is greater than the number of voters, as proofed by Theorem 3 of Bannikova et al. [2].

The following example demonstrates the actions of bot voters.

Example 3.0.2. Let there be a game with 6 voters and 5 candidates. Let bot voter $v_i$ have the following preferences: $a \succ b \succ c \succ d \succ e$ The current selection of $v_i$ is $a$, the current score vector is: $\{a=1, b=0, c=3, d=2, e=0\}$.

- $t = 6$ - For $a$ to become the winner, the five other voters need to change their vote. There are 6 rounds left, so the required number of vote changes is smaller than the number of rounds. ($6 - 1 < 6$). Therefore $a$ is a possible winner. The bot keeps his current selection.

- $t = 5$ - Candidate $a$ is no longer a possible winner ($6 - 1 \geq 5$). The bot checks the other candidates from highest to lowest - $b$ is not a possible winner ($6 - 0 \geq 5$), but $c$ is possible winner, so the bot opts to change to $c$: $b^5_i = c$. 


Chapter 4

The CUD Framework

We built the Consensus Under a Deadline (CUD) framework in order to examine the behavior of human voters and enable a comparison with rational bots. The framework contains 2 modules:

1. The CUD — Game1: a web-based interactive multi-player decision game designed to facilitate an iterative group decision process. The game is implemented as a Java server-client system with an online multilingual HTML with Javascript interface. All voter’s actions are collected and saved to a MySQL database. The game uses a full-duplex asynchronous communication (Web-socket protocol), to enable a fully interactive game between all voters. The full system diagram is demonstrated in Figure 4.1.

2. The CUD — Runner2: a stand alone Java application which receives the data collected by CUD — Game runs, and allows to simulate and analyze the games.

In the following sections we describe the CUD — Game flow in details with step-by-step screenshots (section 4.1), then we describe the Admin screen with all configuration settings (section 4.2), and lastly we explain the reward method defined for this game (section 4.3).

4.1 CUD-Game flow

The running flow of CUD — Game is demonstrated in Figure 4.2. The game can be switched on and off by the game admin, so all voters will play at the same time. While the game is off, the user will not be able to login, and will get an appropriate message. Once the game is on, after reading and confirming to participate in the research (Figure 4.3), each voter logs-in (Figure 4.4) with a name and an identification number, and waits for the other voters to join the game (Figure 4.5). The game begins once the predefined number of voters is reached (Figure 4.6). When the number of logged voters exceeds the number of voters for a game, other games begin simultaneously and independently. At the beginning of the game, each voter receives a predefined preference profile chosen uniformly at random. The highest preference for each voter is selected automatically and the current result is shown.

On each round, each voter may ask to change his selection, or keep his current selection. Once all of the voters reply, the system randomly selects one of the voters who applied for a vote change and submits his new vote. The round ends. The system checks whether the deadline has been reached and whether a consensus has been reached. When the answer to both of these questions is negative, a new round

---

1https://github.com/DavidBenYosef/CUDGame
2https://github.com/DavidBenYosef/CUDRunner
Chapter 4. The CUD Framework

**Figure 4.1: CUD-Game System Diagram**

**Figure 4.2: CUD-Game Flow Diagram**

**Figure 4.3: Instructions screen**
4.1. CUD-Game flow

**Figure 4.4:** Login screen

**Figure 4.5:** Waiting for the game to start

**Figure 4.6:** Game running
FIGURE 4.7: Game finished

FIGURE 4.8: Help chat
begins with the display of the current votes. If no consensus is reached by the deadline, all voters receive 0 points. When a consensus is reached, each voter receives a score corresponding to the chosen preference (Figure 4.7). For example, if candidate $c_1$ is chosen, and voter $v_1$ has this candidate ranked as his second preference, then $v_1$ will receive 80 points for this game.

While the game is on, there was also an active help chat, to provide live support for questions and issues, as seen in Figure 4.8.

### 4.2 CUD-Game admin

The CUD - Game is controlled and configured using the administration page (Figure 4.9). Once the admin logged in to the password-secured interface, he can switch the game on and off, and also modify the following parameters:

- **Preferences set** - the collection of ordered candidate sets. Each voter will be assigned one random set from this collection.
- **Voters** - the amount of voters per game.
- **Rounds** - the total amount of rounds per game.
- **Time per Round** - the time limit for the voter to response per round (measured in seconds).
- **Games Limit** - the total games limit per voter (identified by voter ID)
- **Score Limit** - the total score limit per voter (identified by voter ID)

By modifying these settings, the system can be used for interactively, supporting an effort to reach a group decision by iterative real-time voting.

### 4.3 Reward method

One of the fundamental requirements from CUD - Game is that no consensus is the worse option for all voters, and will be no reward for it. This is the reason why we decided to avoid using the Amazon Turk platform as done by [28], since this
platform requires paying the players for logging in, regardless of their performance. In their research, they faced "ghost" players that played only for the "show-up payment" and thus corrupt the data by inserting noise. In our research, we defined the term of gamepoints to represent abstract reward layer. On each game, the voter had the option to gain between 0 to 100 game points, e.g. when there are 5 candidates, the score if the first preference is selected is 100 game points, the score for the second preference is 80 game points, and so on so that the score of the least preferred candidate is 20 game points. If there was no consensus by the deadline, the voter get 0 game points. The score for each voter is collected and saved, and can be converted and paid in any relevant method later on.
Chapter 5

Experimental Evaluation

We set out to examine human behavior in an iterative voting scenario, where the goal is to reach a consensus before the deadline. We were intrigued by whether human voters would follow the theoretical model suggested by [2], namely, to see whether human voters play in a rational manner.

We identified that our definition of rational agent behavior is too strict for human behavior - we have many "gray" areas which can be defined as rational or irrational actions, for example, a good action but not the best. Moreover, sometimes no action can be irrational as well, and since we can’t find and filter out "sleeping" voters, defining no action as an irrational action might lead to big "noise" in the data.

Therefore, we relaxed the model by defining a rational voter as a voter who does not perform any active irrational actions (section 5.1). We then discuss our evaluation criteria (section 5.2), the collected data (section 5.3) and finally our findings (section 5.4).

5.1 Irrational voters

Irrational voters are votes that actively make actions that are irrational, i.e., actions that do not make any sense. We define 2 types of irrational human voters:

Definition 5.1.1. IRR1 - a voter who opts to change his vote to a candidate $c_j$ that he ranks lower than $c_{maj}$, the candidate with the current highest score. Namely, the voter prefers $c_{maj}$ over $c_j$, but still would like to vote for $c_j$.

If the voter realized that his current selection will not win, and decided to change his selection, he should at least change to $c_{maj}$ and not to a less preferred candidate.

Definition 5.1.2. IRR2 - a voter who opts to change his vote to a candidate $c_j$, when there are other candidates that the voter prefers over $c_j$, and furthermore have a score that is higher than the score of $c_j$. Namely, the voter opts to change his vote for a less preferred candidate with fewer votes.

This behavior is irrational, since there is no reason to change to a less preferred candidate, if this candidate also has a lower score.

The following example demonstrates the differences between the irrational voters of type IRR1 and IRR2.

Example 5.1.3. Let there be a game with 6 voters and 5 candidates. Let voter $v_i$ has the following preferences: $a \succ b \succ c \succ d \succ e$ The game begins with voter $v_i$ voting for his highest preference, $a$. The current score vector is: \{a=1, b=0, c=3, d=2, e=0\}. In all example figures, the current selected candidate is surrounded by a black square, and the new selection is surrounded by an orange square.
5. Experimental Evaluation

Figure 5.1: IRR1 behavior scenario

Figure 5.2: IRR2 behavior scenario

- IRR1 - Figure 5.1 - the voter decides to vote for d, which has a lower rank in \( v_i \) profile than the popular candidate c, but a higher score than the current selection.

- IRR2 - Figure 5.2 - the voter decides to vote for b, which has a lower score and rank than a, but still a higher rank than the popular candidate c.

- Both IRR1 and IRR2 - Figure 5.3 - the voter decides to vote for e. Both IRR1 and IRR2 definitions hold.

We emphasize that we studied only voters that performed irrational actions. For example, consider the following scenario: At time \( t = 6 \) before the deadline, 6 changes are still required for the game to convergence. Voter \( v_1 \) does not opt to change his vote. He might be relying on others to change their vote if no one else does, so he is not defined as irrational. However, consider a more extreme case: At time \( t = 1 \) before the deadline, all voters but \( v_1 \) have agreed on one candidate. If \( v_1 \) does not change his vote, the game will not converge. But we do not define \( v_1 \) as irrational, since he has not performed any irrational action. Since we cannot define \( v_1 \) as rational in the first scenario and irrational in the second scenario, we actually define only very irrational voters as irrational.

5.2 Evaluation Metrics

In order to analyze 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) [4]. We adapt the additive PoA to CUDs as follows.

Definition 5.2.1. Let \( a \) be the truthful profile of voters participating in a CUD, \( b \) denote a ballot profile consistent with \( a \) (i.e., \( b_i = \text{top}_i(C) \)), and \( s = \text{sc}(b) \). Let us denote the

Figure 5.3: Both IRR1 and IRR2 behavior scenario
candidate that CUD converged to by $\hat{C}$, and $s^* = \max_{c \in C} s_c$. Then the Price of Anarchy (PoA) of the CUD is

$$\text{PoA}(a) = \max_{c \in C} s_c - \hat{C}$$

Namely, Price of Anarchy equals to the score of the plurality winner in truthful profile minus the score of the final winner in the truthful profile.

Another evaluation criteria is the convergence time. When there is enough time before the deadline, games should converge. We measure at what time they converge, i.e., do voters wait until the “last minute” to reach a consensus, or do they strive to reach an agreement as early as they can.

Lastly, to measure the satisfaction of the voters, we measure the average reward points of all voters in a game, which are simply the total reward points received by all voters in a game, divided by $n$ (the number of voters).

5.3 Data collection

Phase I: We ran the first experiment with a group of 160 students, which played to get 5 bonus points in a course they took, every 200 gamepoints were equal to one bonus point. The game configuration included: 7 voters per game, 5 candidates, 10 rounds. Every voter was able to participate in an unlimited number of games.

A total of 156 voters played a total of 397 games. In 366 games a consensus was reached. The average convergence time was 5.10 meaning that most games converged well before the deadline. We think that the convergence time was very fast because the voters didn’t really try to maximize their outcome. As they had an unlimited number of games, and their incentive being winning 5 bonus points, they preferred any quick consensus and playing another game to waiting for the last moment and possibly loosing the points.

Phase II: We ran another experiment with a few changes in order to address the findings in the first phase. Again, the voters played to get 5 bonus points in a course they took, every 200 gamepoints were equal to one bonus point, but now we set a limit for the number of games per voter to 15 games, so that the voters will remain motivated to play and “fight” for every point. In addition, we identified that by defining an odd number of voters (7) in the first phase, we made the game easier since ties between candidates were not possible. This meant the voters were required to make less decisions in order to reach a consensus. It is easy for a voter to understand he should change his vote if his preferred candidate is not the one with the most votes. But what should a voter do when there are two candidates tied in the first place, with an equal number of votes? In order to capture these situations, we changed the number of voters per game to be an even number of 8 voters.

A total of 72 students played a total of 264 games: 144 mixed games with bots, 137 of them converged, and 120 Real games, 105 of them converged. Note that humans had no idea whether they were playing against humans or against bots.

Bot phase: We ran a set of bot-only games, 8 bots per game. A total of 10000 bots played 1250 games. As expected, all of the games converged.
Chapter 5. Experimental Evaluation

5.4 Results Analysis

So how do humans behave when asked to reach a consensus? Are they rational? We want to identify how the irrational behavior of humans affects the games from different aspects. First, we identified that even when having "Introduction" games, humans still play irrationally, so we concluded that the irrational actions are not due to misunderstanding. Following this conclusion, we decided to include the introduction games in our test data, and use it to compare the results. We compared between games with irrational actions (IRR1 or IRR2) to games without irrational actions (rational games). We examined all the data collected in the case studies - the first phase with only humans, and the second phase with only humans, mixed games of humans and bots and bot-only games. In order to determine whether our results obtain a statistically significant difference with respect to the four different phases and with respect to the game state (rational or irrational), we performed a two-way ANOVA analysis. All results were found to be significantly different with pvalue<0.05.

Figure 5.4 shows the influence of irrational actions on the convergence of the games. Axis y displays the percentage of converged games out of all games. As expected, we see that irrational actions lower the chances to reach an agreement by 10% – 15%. We also see that bot-only games always converge, and that mixed games converge in higher percentages than human-only games, since less voters can play irrational actions. An interesting comparison is between the human games of Phase1 and Phase2. We see that the convergence of rational games of Phase1 is higher than rational games of Phase2, however the convergence of irrational games of Phase1 is lower than irrational games of Phase2. We explain this phenomenon by the different configuration between the phases - in Phase1, the voters had an unlimited number of games so they were not so motivated to "fight". When the results were good enough for them they played rationally, and when not they "gave up" and started to play irrationally. In the second phase, every voter had a limited number of games, so they were motivated to play even when the game was not in their favor, so more games with irrational actions converged. On the other hand, when playing rationally they tried to strategize, but they miscalculated the needed strategic moves, and less games converged.

Next, we examine whether a high convergence percentage necessarily results in a higher satisfaction, as theoretically a few converged game with high average reward might produce higher satisfaction than many converged game with low average reward.

Figure 5.5 shows the global satisfaction factor, i.e. the average reward points per game. We notice a full correlation with the convergence percentage (figure 5.4); we see a consistent decrease in the satisfaction factor in games with irrational votes.
5.4. Results Analysis

This is to be expected since when a voter plays irrationally, he plays against his interests, and more games do not converge, resulting in a decrease in the satisfaction. The global satisfaction of the rational games of Phase1 is higher than the games of Phase2, and the global satisfaction of irrational games of Phase1 is lower than the games of Phase2. This again correlates with the finding on the convergence percentage; humans are not a good strategists. When they play rationally and are motivated to play and fight, the global satisfaction decreases.

Our most important finding is found here: we see that the global satisfaction increases when bots play with humans, and reaches it’s highest value in bot-only games. In other words, bots are better than human players, even if they are rational human players. Bots in a game (a human-bots game or a bot-only game) increase convergence percentage and increase the satisfaction of all voters, bots and human alike.

Next, we examined the PoA and the convergence time factors between converged games with rational actions and converged games with irrational actions. We avoided comparing games that ended with a mistrial, as the factors for these games are not defined.

Figure 5.6 presents the comparison of the PoA factor for rational and irrational games that converged. The PoA factor represents the distance between the truthful majority winner and the actual unanimity winner. We see that irrational actions cause a higher PoA, since irrational voters play against their interests and may sometimes become the deciding factor, resulting in a consensus which is not the majority winner. We see differences between the PoA of rational games of Phase1 and Phase2. We explain it by the difference in the number of voters per game (7 voters in the first phase and 8 in the second phase). An even number of voters may cause ties between candidates, this leads to a higher PoA, as ties commonly cause a "leader switch" so the final winner is different from the Plurality winner. We also identify a big influence of irrational actions in the mixed games. After further investigation, we identified that when playing with irrational human, Bots reached an higher average

![Figure 5.5: Average reward points comparison](image1)

![Figure 5.6: POA comparison](image2)
Chapter 5. Experimental Evaluation

Figure 5.7: Convergence time comparison

score. We conclude that bots manage to "recover" from irrational actions better than humans, and to reach the best possible option considering the new situation. Interestingly, we see that the PoA of bot-only games is higher than the PoA of games with rational humans. In fact, a high PoA joint with a global satisfaction improvement is an indication of good strategic moves. Bots strategies change the majority winner to some other candidate, leading to an improvement in the global satisfaction.

Figure 5.7 demonstrates the comparison of the convergence time factor for rational and irrational games. The convergence time is the number of rounds the game took to converge. Convergence time 10 is the slowest, and it means the game converged only on the last round. We see that irrational actions cause the games to converge slower, as all other voters need to adapt their logic to the unexpected results. As described above, we identified that Phase2 human games converged slower than Phase1 games, as the voters were motivated to play due to the limitation in the number of games inflicted on the Phase2 voters. Phase2-mixed games converge even slower, and bot-only games converge the slowest, as bot logic makes them converge only when necessary. Humans converge faster since they are afraid of loosing all the point, so they prefer not to wait until the last round, while bots are not afraid since they assume the other voters will work with the same logic.
Chapter 6

Discussion

We set out to examine how humans reach an agreement, i.e. a consensus, within a tight deadline. Using the CUD framework, a game developed in order to collect and analyze human behavior, we compared human behavior to rational bot behavior. The CUD game initiates an iterative voting process, where voters can see the current score that each candidate received, and must decide whether or not to change their vote. The goal is to reach a unanimous agreement. We implemented voter bots, which follow the theoretical model suggested by [2], and compared it with data collected in a user study.

Not surprisingly, there are differences between humans and bots. Perhaps the bots main weakness is that they operate under the assumption that all other voters (who are also bots) have the same logic. However, humans do not hold the same assumption about other humans. Games played by humans converge faster since humans are concerned that the game will not converge and end in a mistrial (the worst option for all involved), so they prefer not to wait until the last round. For the same reason, a single voter playing against bots can manipulate them and change the outcome to his favor, meaning that the result is not necessarily the option preferred by the majority. However, bots increase the convergence percentage and increase the satisfaction of all voters, bots and human alike. Our inevitable conclusion is therefore that when we want confidence that a decision will converge, and that all voters will be as satisfied as possible with the result, then it is best for humans to leave bots to reach a decision on their behalf. A human can notify his personal bot of his preferences, and let the bot represent him in the decision making process.

Bots take an increasing part of our every day lives. We rely on them for navigation directions (e.g. Google Maps or Waze), we rely on them for information retrieval (e.g. any search engine) and we might one day rely on them when striving for a group decision. This research is one step forward in that direction. We see that the results of decisions made by bots only have a higher quality then when we leave it to us humans to reach decisions.

In the future we will be able to extend our framework to accommodate real-life decisions instead of games. Hopefully, voters will be able to set their preferences, and then use the system to find a consensus. The model can be also extended to explore majority scenarios, where a majority decision is needed instead of a unanimous consensus. Lastly, it will be interesting to investigate and develop bots who change their rationalization pending on whether the other voters are humans or bots.
Bibliography


Appendix A

Poster presentation for the IAAI16@Technion
The Goal:
Reaching a consensus under a deadline

For example:
- Location for a family vacation (deadline: a national holiday)
- Jury decision (deadline: court requirements)
- Hiring committee (deadline: beginning of job)

Assumption:
Failing to reach a consensus is the worst outcome for everyone

We wish to examine different voting methods:
- Plurality: \( \{1,0,0, \ldots\} \)
- Borda: \( \{m-1, m-2, \ldots, 0\} \)

Consensus can be defined as a unanimous decision or by a large majority. Vote changes are accommodated, one vote change per iteration.

Strategical Voting

According to the selected voting method, voters may vote strategically in order to optimize their satisfaction from the outcome.

For example: Bob's preferences for a family vacation:
- Hawaii
- Greece
- Spain

The rest of the family prefers either Spain or Greece. Bob will change his preferences and report Greece as his most preferred option, in order to ensure that Spain will not be chosen.

Related Work

- Strategic voting: J.J. Bartholdi III et al. 1989, Zou et al. 2015

What if we combine them all?
How does the deadline factor affect strategic voting?

The Research

We define two rational voter types:
- **Proactive Voter**: Will change his selection whenever it will maximize his outcome.
- **Lazy Voter**: Will change his selection only when the unanimous decision depends on his selection.

The behavior of a rational voter is predictable using a deterministic algorithm according to voter types, and we have prior results for pure computed agent-based game. However, human voting behavior is not predictable.

First stage - Human voters only to determine the human behavior pattern.
Second stage - Mixed type of human and computed agents.

Game Flow:
Every voter receives a preference set.
On every round every voter that wishes to change his vote applies for a vote change, one voter is selected randomly and his new vote is updated.
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' goal is to reach a consensus before the deadline.
Appendix B

Paper presentation for the WIC17@Leipzig
Haste Makes Waste: a Case to Favour Voting Bots

David Ben Yosef1, Lihi Naamani-Dery1, Svetlana Obraztsova2, Zinovi Rabinovich3, Marina Bannikova3

1Ariel University, Israel. davidbenyos@gmail.com, lihid@ariel.ac.il
2Nanyang Technological University, Singapore. [lana,zinovi]@ntu.edu.sg
3Universitat Autònoma de Barcelona, Spain. marina.bannikova@uab.cat

ABSTRACT

Voting is a common way to reach a group decision. When possible, voters will attempt to vote strategically, in order to optimize their satisfaction from the outcome. Previous research has modelled how rational voter agents (bots) vote to maximize their personal utility in an iterative voting process that has a deadline (a timeout). However, it remains an open question whether human beings behave rationally when faced with the same settings. The focus of this paper is therefore to examine how the deadline factor affects manipulative behavior in real-world scenarios. Humans are required to reach a decision before a deadline. An Online platform was built to enable voting games by all types of users: agents (bots), humans, and mixed games with both humans and agents. We compare the results of human behavior and bot behavior and conclude that it might be wise to allow bots to make (certain) decisions on our behalf.

KEYWORDS

Social choice, Voting, Consensus, Deadline, Collective Decision Making

1 INTRODUCTION

Reaching a consensus within a group is a hard task. Decisions may range from choosing the next location for a family’s winter vacation, to a jury’s decision on the outcome of a trial case. In both cases, a deadline for reaching the decision exists. Family members must reach a decision an adequate amount of time before the vacation date. A jury is required to reach a unanimous decision within a reasonable time limit, or else the defendant is acquitted or a new trial is held.

Usually, a decision is not reached immediately, but rather a sequential voting process takes place, where voters change some of their preferences as the deadline approaches. For example, consider Bob, a family member whose preferences for a family vacation are: Hawaii, then Greece and lastly Spain. The rest of Bob’s family members prefer either Spain or Greece. Bob might first try and convince the family of the merits of a vacation in Hawaii. But as the deadline approaches, Bob might change his preferences and report Greece as his most preferred option, in order to ensure that Spain will not be debated for consensus.

Voters may attempt to vote strategically[3], in order to optimize their satisfaction from the outcome (in our toy example, Bob’s change of preferences, from Hawaii to Greece, is a strategic move).

In a previous study, a model for the behavior of strategic agents in a time-bounded iterative voting process has been proposed, providing convergence guarantees and an analysis of the quality of the final decision [1]. However, people are not always rational in their choices.

In this paper, we introduce the CUD game, an online voting system game where a group of people (voters) is required to reach a joint decision. The voters’ ultimate goal is to reach a unanimous winner, and to reach this winner as fast as possible, before the deadline. The voters’ gained utility depends on their performance. At the beginning of the game, each voter receives a set of ranked preferences, and then in an iterative process, voters may change their preferences. As in the theoretical model [1], we rely on the following assumptions: the voting iteration is singular (one voter per iteration) and the number of alternatives and voters is fixed.

We collected data that allows us to analyze human behavior: How often is a consensus reached? When does the process of strategic behavior begin (if at all)? When and if do people change their vote? Lastly, we examined the quality of the final decision reached by a group of people, in terms of the average reward and the price of anarchy. We compared our results with the performance of rational bots, and a mixture of bots and humans.

Contributions: This is a novel attempt to examine strategic voting for a consensus under a deadline, we drop the rational voter behavior assumption and examine how real users behave. Consequently, we reach some key insights as to bot vs. human voting.

2 RELATED WORK

Bannikova et al. [1] analyzed theoretical features of 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. We follow their model but focus on human behavior, our goal being to examine whether humans behave rationally (as in the model).

A feature that distinguishes our framework from other iterative voting processes is the deadline. Informally, the deadline is a time limit on the voting process. In iterative voting processes (e.g. [4–8]) it is a question whether or not the process will converge. Meir et al. [5] were one of the first to attempt to identify when the elections converge to equilibria when allowing the voters to change their vote. They defined the model of iterative voting, in order to study how a single vote change influences the election outcome every time, and when this process converges. Our framework will always converge if the voters are rational and there is a sufficient amount of time before the deadline [1].
Reijngoud et al. [8] analyzed the iterated voting model from different angles: how the amount of information received in one single poll influences the manipulated behavior of the voter, and how the repeated polls affect the manipulated behavior of all voters. Brânzei et al. [2] studies the different influence of strategic voting on the main voting methods. They defined the Price Of Anarchy (POA) factor that measures the differences between truthful voting and worse-case manipulated voting. POA is one of the measures we use in the experimental evaluation.

On-line platforms such as Doodle (www.doodle.com) allow people to submit their votes on the problem in question, to view the votes of other group members, and to change their votes if they wish. Zou et al. [11] studied the strategic voting behaviour on Doodle polls, and how the knowledge of current voters preferences affected the results. Interestingly, people tend to change their vote preferences in light of the votes of their peers. In our model the voter preference are kept private and voters can see only the intermediate aggregated results.

Experimental studies of bargaining deadlines have shown that the majority of agreements are obtained in the final seconds before the deadline; contrary to what might be expected, even complete information does not speed agreement much [9]. As opposed to bargaining models, in our suggested framework voters must agree on one alternative and no compromise between alternatives is possible.

Tal et al. [10] performed a study of human behavior in online voting. They classify voters into 3 distinct types, two of which are not strategic and one that will perform straight forward strategic moves. However their setting does not contain a deadline. Moreover, while they provide valuable insights on human behavior, they do not compare their framework with a rational strategic model, so the question of whether humans and bots act alike remains unanswered.

3 RATIONAL AGENT BEHAVIOUR

We follow the rational agents’ behavior model suggested by Ban- níková et al. [1] the model introduces an attempt to reach a decision under a deadline based on a bounded iterative voting process. We concentrate on the plurality voting rule, where each voter casts a single vote for his most preferred alternative. The alternative that receives the most votes wins.

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 denote two types of voters: rational automated voters (bots) $v$, and human voters $\hat{v}$. Each voter is characterised by a truthful preference $A_i, \hat{A}_i \in L(C)$; a complete and non-reflexive order over the set of alternatives. We write $c >_i c'$ if voter $i$ prefers $c$ to $c'$. At the beginning of the process, every voter $i \in V$ casts a ballot $b^i$ of $C$ that reveals her most preferred candidate at time $t = \tau$. The ballots of all voters are collected, forming a ballot profile, $b = (b^1, \ldots, b^n)$. The score of each candidate within a given ballot profile is the sum of the voters that voted for this candidate:

$$\text{sc}_c(b) = |\{i | b^i_c = c\}|.$$

A score vector is a collection of scores of all candidates $\text{sc}(b) = (\text{sc}(b^1), \ldots, \text{sc}_m(b^i))$. The collection of score vectors is public information, and is visible to all the voters.

Once the collection of score vectors are published, each voter decides whether to change her vote and produce a ballot $b^i_t \in C$, i.e., state her (new) vote at time $t$.

If the iterative process converges to an alternative, that alternative is called the winner. A Possible winner candidate is a candidate which still has enough time to become a winner if all voters will ask to change to it. More formally:

Definition 3.1. A candidate is a possible winner $PW$ if the number of required vote changes is less than number of remaining rounds. The Required vote changes for a candidate is the current candidate score subtracted from the number of voters. $PW_c = \text{sc}_c(b^i) - n$

Algorithm 1 Rational agent logic

```
Input: Truthful profile $A$, selected candidate $c_i$
Input: Number of voters $v$, deadline timeout $\tau$
Input: Current score vector $R$
1: if $c_i$ is a PW in $R$ with $v$ voters on time $\tau$ then
2: return $c_i$ as the selected $> \text{No Change}$
3: else
4: for $i \in A$ do $> \text{From most preferred to least preferred}$
5: if $c_i$ is a PW in $R$ with $v$ voters at time $\tau$ then
6: return $c_i$ as the selected
7: end if
8: end for
9: end if
```

For the bots, we follow the design of proactive voters, as defined in [1] and described in Algorithm 1. Namely, if the bot’s current selection is a possible winner, do not ask to change the vote. If the bot’s current selection is not a possible winner then check the next candidate in the bot’s ordered set of preference. If that candidate is a possible winner, the bot will ask to change its vote to this candidate. If not, continue to the candidate ranked next.

Using this algorithm, a bot will always maximize its outcome since it always selects the most highly ranked possible winner. When all voters play with this logic, the game will always converge if the number of rounds is greater than the number of voters.

The following example demonstrates the actions of bot voters.

Example 3.2. Let there be a game with 6 voters and 5 candidates. Let bot voter $v_i$ have the following preferences: $a > b > c > d > e$ The current selection of $v_i$ is $a$, the current score vector is: $\{a=1, b=0, c=3, d=2, e=0\}$.

- $t = 6$ - For $a$ to become the winner, the five other voters need to change their vote. There are 6 rounds lest, so the required number of vote changes is smaller than the number of rounds. $(6 - 1 < 6)$. Therefore $a$ is a possible winner. The bot keeps his current selection.
- $t = 5$ - Candidate $a$ is no longer a possible winner $(6-1 \not\leq 5)$. The bot checks the other candidates from highest to lowest - $b$ is not a possible winner $(6 - 0 \not\leq 5)$, but $c$ is possible winner, so the bot opts to change to $c$: $b^i_c = c$. 

```
4 THE CUD FRAMEWORK

We built the Consensus Under a Deadline (CUD) framework in order to examine the behaviour of human voters and enable a comparison with rational bots. The framework contains 2 modules:

1. The CUD – Game: a web-based interactive multi-player decision game designed to facilitate an iterative group decision process. The game is implemented as a server-client system with an online interface. All voter’s actions are collected and saved to a database. The game uses a full-duplex asynchronous communication protocol, to enable a fully interactive game between all voters.

2. The CUD – Runner: a stand alone application which receives the data collected by CUD – Game runs, and allows to simulate and analyze the games.

Once the game is on, each voter logs-in with a name and an identification number, and waits for the other voters to join the game. The game begins once the predefined number of voters is reached. When the number of logged voters exceeds the number of voters for a game, other games begin simultaneously and independently. At the beginning of the game, each voter receives a predefined preference profile chosen uniformly at random. The highest preference for each voter is selected automatically and the current result is shown.

Each preference represents the corresponding value for the voter, e.g., when there are 5 candidates, the score of the first preference is 100, the second preference is 80, and so on so that the score of the least preferred candidate is 20.

On each round, each voter may ask to change his selection, or keep his current selection. Once all of the voters reply, the system randomly selects one of the voters who applied for a vote change and submits his new vote. The round ends. The system checks whether the deadline has been reached and whether a consensus has been reached. When the answer to both of these questions is negative, a new round begins with the display of the current votes. If no consensus is reached by the deadline, all voters receive 0 points. When a consensus is reached, each voter receives a score corresponding to the chosen preference. For example, if candidate c1 is chosen, and voter v1 has this candidate ranked as his second preference, then v1 will receive 80 points for this game.

The CUD – Game can be configured using the following parameters:

- Preferences set - the collection of ordered candidate sets. Each voter will be assigned one random set from this collection.
- Voters - the amount of voters per game.
- Rounds - the total amount of rounds per game.
- Time per Round - the time limit for the voter to respond per round (measured in seconds).
- Games Limit - the total games limit per voter (identified by voter ID)
- Score Limit - the total score limit per voter (identified by voter ID)

By modifying these settings, the system can be used for interactively, supporting an effort to reach a group decision by iterative real-time voting.

5 EXPERIMENTAL EVALUATION

We set out to examine human behavior in an iterative voting scenario, where the goal is to reach a consensus before the deadline. We were intrigued by whether human voters would follow the theoretical model suggested by [1], namely, to see whether human voters play in a rational manner. A rational voter is a voter who does not perform any irrational actions. We therefore first defined irrational voters (section 5.1). We then discussed our evaluation criteria (section 5.2), the collected data (section 5.3) and finally our findings (section 5.4).

5.1 Irrational voters

Irrational voters are voters that actively make actions that are irrational, i.e., actions that do not make any sense. We defined 2 types of irrational human voters:

Definition 5.1. IRR1 - a voter who opts to change his vote to a candidate c_j that he ranks lower than c_{maj}, the candidate with the current highest score. Namely, the voter prefers c_{maj} over c_j, but still would like to vote for c_j.

If the voter realised that his current selection will not win, and decided to change his selection, he should at least change to c_{maj} and not to a less preferred candidate.

Definition 5.2. IRR2 - a voter who opts to change his vote to a candidate c_j, when there are other candidates that the voter prefers over c_j, and furthermore have a score that is higher than the score of c_j. Namely, the voter opts to change his vote for a less preferred candidate with fewer votes.

This behavior is irrational, since there is no reason to change to a less preferred candidate, if this candidate also has a lower score.
The current score vector is: \{a, b, c, d, e\}. In all example figures, the current selected candidate is surrounded by a black square, and the new selection is surrounded by an orange square.

- **IRR1** - Figure 2 - the voter decides to vote for c, which has a lower score than the rank of the current selection.
- **IRR2** - Figure 3 - the voter decides to vote for b, which has a lower rank than c, but a higher score than the current candidate c.
- **Both** IRR1 and IRR2 - Figure 4 - the voter decides to vote for c. Both IRR1 and IRR2 definitions hold.

We emphasize that we studied only voters that performed irrational actions. For example, consider the following scenario: At time \( t = 6 \) before the deadline, 6 changes are still required for the game to converge. Voter \( v_2 \) does not opt to change his vote. He might be relying on others to change their vote if no one else does, so he is not defined as irrational. However, consider a more extreme case: At time \( t = 1 \) before the deadline, all voters but \( v_1 \) have agreed on one candidate. If \( v_1 \) does not change his vote, the game will not converge. But we do not define \( v_1 \) as irrational, since he has not performed any irrational action. Since we cannot define \( v_1 \) as rational in the first scenario and irrational in the second scenario, we actually define only very irrational voters as irrational.

### 5.2 Evaluation Metrics

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)* \([2]\). We adapt the additive PoA to CUDs as follows.

**Definition 5.4.** Let \( a \) be the truthful profile of voters participating in a CUD, \( b \) denote a ballot profile consistent with \( a \) (i.e., \( b_i = \text{top}_i(C) \)), and \( s = sc(b) \). Let us denote all candidates that CUD may converge to by \( \hat{C} \), and \( s^* = \max_{c \in \hat{C}} s_c \). Then the *additive Price of Anarchy (PoA)* of the CUD is

\[
\text{PoA}^a(a) = \max_{c \in \hat{C}} s_c - \min_{c \in \hat{C}} s_c
\]

Namely, Additive PoA is the score of the least preferred alternative that was selected as a winner in one out of numerous CUD games, subtracted from the score of the truthful winner.

Another evaluation criteria is the convergence time. When there is enough time before the deadline, games should converge. We measure at what time they converge, i.e., do voters wait until the ‘last minute’ to reach a consensus, or do they strive to reach an agreement as early as they can.

Lastly, to measure the satisfaction of the voters, we measure the average reward points of all voters in a game, which are simply the total reward points received by all voters in a game, divided by \( n \) (the number of voters).

### 5.3 Data collection

**Phase I:** We ran the first experiment with a group of 160 students, which played to get 5 bonus points in a course they took. The game configuration included: 7 voters per game, 5 candidates, 10 rounds. Every voter was able to participate in an unlimited number of games.

A total of 156 Voters played a total of 397 games. In 366 games a consensus was reached. The average convergence time was 5.10 meaning that most games converged well before the deadline. We think that the convergence time was very fast because the voters didn’t really try to maximize their outcome. As they had an unlimited number of games, and their incentive being winning 5 bonus points, they preferred any quick consensus and playing another game to waiting for the last moment and possibly loosing the points.

**Phase II:** We ran another experiment with a few changes in order to address the findings in the first phase. We set a limit for the number of games per voter, so that the voters will remain motivated to play and “fight” for every point. In addition, we identified that by defining an odd number of voters (7) in the first phase, we made the game easier since ties between candidates were not possible. This meant the voters were required to make less decisions in order to reach a consensus. It is easy for a voter to understand he should change his vote if his preferred candidate is not the one with the most votes. But what should a voter do when there are two candidates tied in the first place, with an equal number of votes? In order to capture these situations, we changed the number of voters per game to be an even number of 8 voters.
A total of 72 students played a total of 264 games: 144 mixed games with bots, 137 of them converged, and 120 Real games, 105 of them converged. Note that humans had no idea whether they were playing against humans or against bots.

**Bot Phase:** We ran a set of bot-only games, 8 bots per game. A total of 10000 bots played 1250 games. As expected, all of the games converged.

### 5.4 Results Analysis

So how do humans behave when asked to reach a consensus? Are they rational? We want to identify how the irrational behavior of humans affects the games from different aspects. We compared between games with irrational actions (IRR1 or IRR2) to games without irrational actions (rational games). We examined all the data collected in the case studies - the first case with only humans, and the second with only humans, mixed games of humans and bots and bot-only games.

Figure 5 shows the influence of irrational actions on the convergence of the games. Axis y displays the percentage of converged games out of all games. As expected, we see that irrational actions lower the chances to reach an agreement by 10% – 15%. We also see that bot-only games always converge, and that mixed games converge in higher percentages than human-only games, since less voters can play irrational actions. An interesting comparison is between the human games of Phase1 and Phase2. We see that the convergence of rational games of Phase1 is higher than rational games of Phase2, however the convergence of irrational games of Phase1 is lower than irrational games of Phase2. We explain this phenomenon by the different configuration between the phases - in Phase1, the voters had an unlimited number of games so they were not so motivated to “fight”. When the results were good enough for them they played rationally, and when not they “gave up” and started to play irrationally. In the second phase, every voter had a limited number of games, so they were motivated to play even when the game was not in their favor, so more games with irrational actions converged. On the other hand, when playing rationally they tried to strategize, but they miscalculated the needed strategic moves, and less games converged.

Next, we examine whether a high convergence percentage necessarily results in a higher satisfaction, as theoretically a few converged game with high average reward might produce higher satisfaction than many converged game with low average reward.

Figure 6 shows the global satisfaction factor, i.e. the average reward points per game. We notice a full correlation with the convergence percentage (figure 5); we see a consistent decrease in the satisfaction factor in games with irrational votes. This is to be expected since when a voter plays irrationally, he plays against his interests, and more games do not converge, resulting in a decrease in the satisfaction. The global satisfaction of the rational games of Phase1 is higher than the games of Phase2, and the global satisfaction of irrational games of Phase1 is lower than the games of Phase2. This again correlates with the finding on the convergence percentage; humans are not a good strategists. When they play rationally and are motivated to play and fight, the global satisfaction decreases.

Our most important finding is found here: we see that the global satisfaction increases when bots play with humans, and reaches it’s highest value in bot-only games. In other words, bots are better than human players, even if they are rational human players. Bots in a game (a human-bots game or a bot-only game) increase convergence percentage and increase the satisfaction of all voters, bots and human alike.

Next, we examined the PoA and the convergence time factors between converged games with rational actions and converged games with irrational actions. We avoided comparing games that ended with a mistrial, as the factors for these games are not defined.

Figure 7 presents the comparison of the PoA factor for rational and irrational games that converged. The PoA factor represents the distance between the truthful majority winner and the actual unanimity winner. We see that irrational actions cause a higher PoA, since irrational voters play against their interests and may sometimes become the deciding factor, resulting in a consensus.
which is not the majority winner. We see differences between the PoA of rational games of Phase1 and Phase2. We explain it by the difference in the number of voters per game (7 voters in the first phase and 8 in the second phase). An even number of voters may cause ties between candidates, which leads to a higher PoA. We also identify a big influence of irrational actions in the mixed games since bots cannot cope with irrational behavior. We conclude that a single voter can manipulate and affect the results dramatically when playing with only bots. Interestingly, we see that the PoA of bot-only games is higher than the PoA of games with rational humans. In fact, a high PoA joint with a global satisfaction improvement is an indication of good strategic moves. Bots strategies change the majority winner to some other candidate, leading to an improvement in the global satisfaction.

Figure 8 demonstrates the comparison of the convergence time factor for rational and irrational games. The convergence time is the number of rounds the game took to converge. Convergence time 10 is the slowest, and it means the game converged only on the last round. We see that irrational actions cause the games to converge slower, as all other voters need to adapt their logic to the unexpected results. As described above, we identified that Phase2 human games converged slower than Phase1 games, as the voters were motivated to play due to the limitation in the number of games inflicted on the Phase2 voters. Phase2-mixed games converge even slower, and bot-only games converge the slowest, as bot logic makes them converge only when necessary. Humans converge faster since they are afraid of loosing all the point, so they prefer not to wait until the last round, while bots are not afraid since they assume the other voters will work with the same logic.

6 DISCUSSION AND FUTURE WORK

We set out to examine how humans reach an agreement, i.e. a consensus, within a tight deadline. Using the CUD framework, a game developed in order to collect and analyse human behavior, we compared human behavior to rational bot behavior. The CUD game initiates an iterative voting process, were voters can see the current score that each candidate received, and must decide whether or not to change their vote. The goal is to reach a unanimous agreement. We implemented voter bots, which follow the theoretical model suggested by [1], and compared it with data collected in a user study.

Not surprisingly, there are differences between humans and bots. Perhaps the bots main weakness is that they operate under the assumption that all other voters (who are also bots) have the same logic. However, humans do not hold the same assumption about other humans. Games played by humans converge faster since humans are concerned that the game will not converge and end in a mistrial (the worst option for all involved), so they prefer not to wait until the last round. For the same reason, a single voter playing against bots can manoeuvre them and change the outcome to his favor, meaning that the result is not necessarily the option preferred by the majority. However, bots increase the convergence percentage and increase the satisfaction of all voters, bots and human alike. Our inevitable conclusion is therefore that when we want confidence that a decision will converge, and that all voters will be as satisfied as possible with the result, then it is best for humans to leave bots to reach a decision on their behalf. A human can notify his personal bot of his preferences, and let the bot represent him in the decision making process.

Bots take an increasing part of our every day lives. We rely on them for navigation directions (e.g. Google Maps or Waze), we rely on them for information retrieval (e.g. any search engine) and we might one day rely on them when striving for a group decision. This research is one step forward in that direction. We see that the results of decisions made by bots only have a higher quality when then we leave it to us humans to reach decisions.

In the future we plan to extend our framework to accommodate real-life decisions instead of games. Hopefully, voters will be able to set their preferences, and then use the system to find a consensus. We also plan to explore majority scenarios, where a majority decision is needed instead of a unanimous consensus. Lastly, we set out to develop bots who change their rationalization pending on whether the other voters are humans or bots.

REFERENCES