Lie on the Fly: Iterative Voting Center with Manipulative Voters*

Lihi Naamani-Dery[1], **Svetlana Obraztsova**[2], **Zinovi Rabinovich**[3], **Meir Kalech**[4]
[1]Ariel University, Ariel, Israel. lihid@ariel.ac.il
[2]Tel Aviv University, Tel Aviv, Israel. svetlana.obraztsova@gmail.com
[3]Mobileye Vision Technologies Ltd., Jerusalem, Israel. zr@zinovi.net
[4]Ben-Gurion University, Beer Sheva, Israel. kalech@bgu.ac.il

Abstract

Manipulation can be performed when intermediate voting results are known; voters might attempt to vote strategically and try and manipulate the results during an iterative voting process. When only partial voting preferences are available, preference elicitation is necessary. In this paper, we combine two approaches of iterative processes: iterative preference elicitation and iterative voting and study the outcome and performance of a setting where manipulative voters submit partial preferences. We provide practical algorithms for manipulation under the Borda voting rule and evaluate those using different voting centers: the Careful voting center that tries to avoid manipulation and the Naive voting center. We show that in practice, manipulation happens in a low percentage of the settings and has a low impact on the final outcome. The Careful voting center reduces manipulation even further.

1 Introduction

Iterative Voting (see e.g. [Meir *et al.*, 2010]) proceeds in rounds. In every round the voters are allowed to change their profile, i.e., their reported preferences over the candidates. Voters might attempt to vote strategically, that is to state preferences that will increase the chance that the final outcome will match their truthful profile. Practical applications, such as Doodle (www.doodle.com) and SurveyMonkey (www.surveymonkey.com), provide a real world example of iterative voting processes. Currently, 20%-30% of the internet users use on-line calenders for their daily needs[1], making online scheduling a natural next step. Therefore the importance of research into iterative voting can hardly be overstated. Changing the profile is known as strategic voting or as manipulation [Farquharson, 1969; Laffont, 1987], and is readily found in the real world as well (see e.g. [Zou *et al.*, 2015]). Computing a manipulation is possible since the voters know the intermediate results of the voting process.

In this paper we claim that iterative voting can be applied when the voters' preferences are incomplete. Incomplete preferences are a realistic scenario for different reasons. First, due to privacy concerns full preference revelation should be treated with caution. Secondly, it is sometimes completely impractical to collect complete preferences due to the communication burden [Conitzer and Sandholm, 2005; Konczak and Lang, 2005] or to voter limitations. Consider, for example a meeting scheduling application whose purpose is to set a time for a conference [Kalech *et al.*, 2011], or an application that recommends movies, such as Netflix (www.netflix.com). It is impractical to expect the voters to provide their preferences on all available options as there might be hundreds available.

As a result, a separate direction has developed which studies the *iterative* elicitation from *non manipulative voters*. The assumption is that voters submit only a part of their preferences and cannot change already submitted preferences. Also, the voters are not aware of the intermediate result but are only shown the final outcome. In the worst case for most voting protocols all the preferences are needed in order to determine a winning item, i.e., an item that most certainly suits the group's joint preferences [Conitzer and Sandholm, 2005; Xia and Conitzer, 2011]. Nevertheless, recent works [Naamani-Dery *et al.*, 2014; 2015; Lu and Boutilier, 2011; 2013] show that in practice the required information can be cut by more than 50%. To the best of our knowledge, iterative preference elicitation has not been studied with *manipulative* voters before.

In this paper, we combine the two approaches of iterative processes: iterative preference elicitation and iterative voting and study the outcome and performance of a setting where manipulative voters submit partial preferences. In this scenario, the Voting Center proceeds in rounds. In each round the center selects one voter to query for her preference between two candidate items; we follow the terminology in [Lu and Boutilier, 2011] and term this a voter-item-item query. At the end of each round, the Voting Center exposes the set of possible winner candidates, i.e. the candidates that might win the elections. Thus the voters can attempt to manipulate by sending the Voting Center an insincere response in order to promote or avoid certain candidates according to their truth-

*This research was partially supported by: European Union (European Social Fund — ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) — Research Funding Program: THALES; RFFI 14-01-00156-a

[1] http://en.blog.doodle.com/2011/07/13

ful preferences. The voters' responses to queries must not contradict previously stated preferences.

In the iterative voting setting with manipulation there are two main challenges. First, from the Voting Center's point of view: how to select a voter-item-item query which is the most manipulation proof. The second challenge, from the manipulative voters' point of view: find a manipulation which will increase the chance to achieve a better selfish outcome, as is done in the voting literature since the classical works [Gibbard, 1973; Satterthwaite, 1975]. We assume only one query is executed in each iteration. Since combining preference elicitation with voting manipulation is a novel idea, we chose to begin with the Borda voting rule, as it can easily be expanded to other voting rules. It must be noted that several general results have been obtained regarding the complexity of manipulating Borda with partial information [Conitzer *et al.*, 2011; Xia and Conitzer, 2011]. However, these works assume that a complete preference order is submitted as a ballot. Since in our case only a very limited portion of an order is submitted, off hand it may still be possible to obtain a manipulation strategy efficiently. This, in turn, supports the need to resolve the first challenge.

Contributions: For the first challenge, we present a Careful Voting Center which tries to prevent manipulation. For the second, we provide a *practical* manipulation algorithm for voters in an iterative voting setting. In order to evaluate the manipulation impact on the preference elicitation process, we constructed an experiment on real-world data. We compared manipulative voters to truthful voters in a Careful and in a Naive Voting Center setting and examined three research questions: (1) How often does manipulation occur in practice? (2) Does manipulation impact the final result? (3) How does manipulation impact the number of iterations required to reach the final result? We show that in practice, manipulation happens in a low percentage of the settings and has a low impact on the final outcome. A Careful Voting Center reduces the manipulation even further.

2 Related Work

Practical vote elicitation has been addressed recently. Ding and Lin define a candidate winning set as the set of queries needed in order to determine whether the candidate is a necessary winner and show that for rules other than the plurality voting, computing this set is NP-Hard [Ding and Lin, 2013].

Therefore heuristics are needed for preference elicitation. Practical elicitation heuristic algorithms that aim to minimize preference communication have been addressed in a few papers. Lu and Boutilier propose a practical elicitation process for the Borda voting protocol using the minmax regret concept. The outcome is a definite or an approximate winner [Lu and Boutilier, 2011; 2013]. Naamani-Dery et. al. suggest practical elicitation algorithms that incorporate probabilistic information for the Range and Borda voting protocol [Naamani-Dery *et al.*, 2014; 2015].

In iterative voting, it is assumed that rather than simply finding the election outcome, voters behave strategically [Farquharson, 1969]. Voters are allowed to iteratively examine the currently recorded intermediate election results and ballot alteration (e.g. [Meir *et al.*, 2010; Lev and Rosenschein, 2012; Branzei *et al.*, 2013; Kukushkin, 2011; Reyhani and Wilson, 2012]). Works in this direction can be roughly broken into two categories: the characterization of the stable points of the iterative voting process, echoing the older interest in equilibrium characterization (e.g. [Rabinovich *et al.*, 2015]), and the research into the conditions that guarantee the iterative voting dynamics to converge (e.g. [Reijngoud and Endriss, 2012; Grandi *et al.*, 2013; Obraztsova *et al.*, 2015; Meir *et al.*, 2014]).

It has been proven that it is computationally hard to compute the possible and necessary winners, or to manipulate elections in incomplete voting settings [Walsh, 2007; Pini *et al.*, 2007] Our paper is orthogonal to these claims and proposes a practical manipulation strategy for iterative voting. We challenge the separation of iterative preference elicitation and iterative voting and focus on the questions of how the interaction of these two concepts affects the elections' outcome.

3 The Model

A voting model consists of a set of candidates $C = \{c_1, ..., c_m\}$ and the set voters $V = \{v_1, ..., v_n\}$. At the beginning of the process, none of the voters' preferences are known. In an iterative process, the Voting Center selects a voter-item-item query, i.e., a voter v_i and two items $c_j, c_k \in C$. The selected voter is requested to submit her preferences between the two items. A voter's profile P_i is a vector of ordered preferences: $P_i = [c_{i_1}, ..., c_{i_m}]$ where c_{i_1} is v_i's most preferred candidate. The voter responds $c_j \succ c_k$ when c_j is preferred over c_k or vice versa. The Voting Center maintains a set $Q_i \in \mathcal{Q}$ of *partial preferences* of each voter. Based on the partial preferences of all voters $\mathcal{Q}$, the voting center computes a set of the Possible Winners PW. The center gradually expands the set until a Necessary Winner is found. We follow [Konczak and Lang, 2005] definitions of Possible and Necessary winners. The tie-breaking policy when a few Necessary Winners exist is that the winner with the smallest lexicographical order is chosen.

Previous work using an iterative center [Naamani-Dery *et al.*, 2014; Lu and Boutilier, 2013] assumed that the voters always submit their true preferences and that they have no knowledge of the voting process's intermediate results [Walsh, 2007]. However, we follow [Gibbard, 1973; Satterthwaite, 1975] and assume that voters attempt to manipulate the elicitation process in order to achieve a better selfish outcome. The voters are allowed to observe the current ballot summary similarly to [Conitzer *et al.*, 2011; Reijngoud and Endriss, 2012]. The summary is expressed by a set of Possible Winners PW.

Our model does not assume any specific query selection protocol and several possibilities for selecting the voter-item-item queries exist [Naamani-Dery *et al.*, 2015; Lu and Boutilier, 2011]. The interaction between the voting center and the voters proceeds as follows: At the beginning, each voter holds her true set of preferences $P_i = P_{i_{true}}$. As long as the necessary winner has not been identified: a) the Center selects a voter-item-item query, $\langle v_i \langle cj, ck \rangle \rangle$; b) the voter is provided with the current set of Possible Winners PW; c) the voter decides if to change her profile from P_i to P'_i; d) the voter responds with either $c_j \succ c_k$ or $c_k \succ c_j$; e) the Center updates the incomplete profile by incorporating v_i's answer and applying transitive closure.

4 Locally Dominant Manipulation

The voter will change her current profile from P_i to P'_i if she recognizes that the change will increase the chance that the final outcome will match her truthful profile. The set $\mathcal{P}_i$ represents all the alternatives of P'_i, including the current P_i. For example, assume the set of possible winners is $PW = \{c_2, c_5\}$, and v_1 received a query for her preferences between c_2 and c_3, where v_1's current profile is: $P_1 = [c_3, c_2, c_5 ...]$. v_1 will recognize that c_3 has no chance of winning since it is not in the Possible Winners set. Therefore she will update her profile to be: $P'_1 = [c_2, ..., c_3, ...]$ and respond with $c_2 \succ c_3$. The voter's change of profile is always consistent with her set of previously stated preferences (Q_i).

We first describe under what conditions voters manipulate and then present algorithms for voter manipulation in an iterative voting process.

4.1 Conditions for Manipulation

The voters manipulate and state false preferences by employing a local dominance manipulation model (see e.g. [Meir *et al.*, 2014; Reijngoud and Endriss, 2012; Conitzer *et al.*, 2011]). Local Dominance is defined as:

Definition 1. [Local Dominance] *Let $P_{i_{true}}$ be v_i's true preference. A preference order P'_i is a local dominant over preference order P_i if in at least one outcome the Necessary Winner is ranked closer to v_i's true preferences, and in none of the outcomes the Necessary Winner is ranked lower.*

When queried, the voter is requested to submit her preference between two candidates only. However, in order to manipulate, more than a single change in preferences might be needed. From all of the possible changes in P_i, our algorithm selects the change that requires the minimal number of swaps, i.e., the voter performs the minimal change in her preferences. To compare two profiles, we use the *swap distance* [Kendall, 1938] defined for two linear orders P and P'. The distance counts the number of candidate pairs that are ordered differently by two ballots or linear orders.

The voter will change her profile to $P'_i \in \mathcal{P}_i$ under the following conditions only:

- **VM-Condition-1:** The new preference profile P'_i is a local dominant over P_i.
- **VM-Condition-2:** The new preference profile P'_i has the minimal swap distance out of all possible local dominant profiles. $P'_i \in \arg\min\limits_{P'_i \in \mathcal{P}_i} d_{swap}(P'_i, P_i)$.

We now define the scenarios were manipulation can be performed. For a given set of Possible Winners $PW = \{pw_1, ..., pw_l\}$ we define PW_i as the ordered vector of possible winners for voter v_i: $PW_i = [pw_{i_1}, ..., pw_{i_l}]$. We define E_i as the ordered vector of preferences which contains all possible winners and candidates between these possible winners; e.g, if $PW = \{c_2, c_5\}$ and $P_1 = [c_3, c_2, c_4, c_5, c_1]$ then $E_1 = [c_2, c_4, c_5]$.

Assume that for a given query $\langle v_i \langle c_j, c_k \rangle \rangle$, according to v_i's current profile $P_i : c_j \succ c_k$. We define two special subsets, $E_{i_{up}}$ and $E_{i_{down}}$, of candidates that come from E_i whose relative position wrt c_j and c_k we can not change due to our previous obligations recorded by Q_i:

$$E_{i_{up}} = \{c \in E_i | (c \succ c_k) \in Q_i\}$$
$$E_{i_{down}} = \{c \in E_i | (c_j \succ c) \in Q_i\}$$

In our running example, $E_{1_{up}} = [c_2, c_4]$, when the query is $\langle v_1 \langle c_3, c_5 \rangle \rangle$ and $Q_i = Closure(\{(c_2 \succ c_4), (c_4 \succ c_5)\})$.

To set the conditions for local dominance, we use the following set of *Common Givens*:

- The current preference profile of v_i is P_i
- The query is: $\langle v_i \langle c_j, c_k \rangle \rangle$, and according to P_i: $c_j \succ c_k$
- The ordered vector of Possible Winners of v_i is PW_i
- The closure of the set of current query responses is Q_i

Now, denote $SG(c, c', P_i)$ the set of candidates in preference profile P_i between two candidates, $c \succ c'$, inclusive of the two candidates themselves. The following Theorem states that a manipulative response to a query has to maintain the same implied order of possible winners. Furthermore, for at least one pair of consecutive possible winners the distance between them grows. In turn, the Corollary states that these ordering and distance properties can be consistently traced from the truth profile throughout all responses of a voter.

Theorem 1. *A preference profile P'_i is a local dominant over P_i if and only if the following holds:*

- $pw_l \succ pw_{l+1}$ *for all* $l \in [1, ..., k-1]$
- $|SG(pw_l, pw_{l+1}, P'_i)| \geq |SG(pw_l, pw_{l+1}, P_i)|$ *for all* $l \in [1, ..., k-1]$
- *Exists* $l \in [1, ..., k-1]$ *so that* $|SG(pw_l, pw_{l+1}, P'_i)| \gneqq |SG(pw_l, pw_{l+1}, P)|$

Corollary 1. *Let $\tau > t$, and P^t_i, P^τ_i are the preference profiles of voter v_i at times t and τ respectively. Then the set of possible winners PW at time τ will be ordered the same way by $P_{i_{true}}$ (the truthful preference of v_i), P^t_i and P^τ_i. Furthermore, the size of each segment between consecutive possible winners in PW will monotonically grow from $P_{i_{true}}$ to P^t_i to P^τ_i, and the total size of these segments will grow strictly monotonically.*

It is possible to devise an algorithm that implements the properties described by Theorem 1, i.e. looks for a possible manipulation. Taking into account the swap distance limitations, it essentially means that for a given query we will be testing whether one candidate of the query is in E_i and the other is not. If it is the case, we will test which ordering between them would imply the smallest distance between P_i and P'_i constructed by adding the considered query response. It is possible to give a set of formal lemmata that provide the sufficient and the necessary conditions used by the aforementioned tests. We omit them due to space limitations, however we do provide the algorithms they entail. The algorithms find all possible manipulations that meet VM-Conditions 1&2.

4.2 Manipulation Algorithms

We identify six situations where a profile can be changed. Given a query $\langle v_i \langle c_j, c_k \rangle \rangle$, when according to P_i, $c_j \succ c_k$ and a set of l Possible Winners PW, the situations are grouped into two symmetrical families:

- **Family I:** $c_j \succ pw_{i_1}$
  - **Case-A:** $c_k \in PW \setminus \{pw_{i_1}, pw_{i_l}\}$,
  - **Case-B:** $c_k = pw_{i_1}$,
  - **Case-C:** $c_k \in E_i \setminus PW$.
- **Family II:** $pw_{i_l} \succ c_k$
  - **Case-D:** $c_j \in PW \setminus \{pw_{i_1}, pw_{i_l}\}$,
  - **Case-E:** $c_j = pw_{i_l}$,
  - **Case-F:** $c_j \in E_i \setminus PW$.

We describe algorithms for Family I only, since the Family II algorithms are symmetric to Family I. Algorithms 1-3 correspond to Case-A through Case-C respectively.

Algorithm 1 VoterManipulation function: Case-A

```
Require:
    Common Givens
    Holds $c_j \succ pw_{i_1}$, $c_k = pw_{i_k}$
if $E_{i_{down}} \cap \ SG(pw_{i_1}, c_k, P_i) \neq \emptyset$ then
    return $P_i$
end if
$S_{good} \leftarrow \{c \in C | c \succ p_1)\} \setminus (E_{i_{down}} \cup \{c_k\})$
$S_{bad} \leftarrow \{c \in C | c \succ p_1)\} \cap \ E_{i_{down}}$
$Y_{head} \leftarrow \ E_{i_{up}}$
$Y_{tail} \leftarrow \{c \in C | c_k \succ c)\}$
Order $S_{good}, S_{bad}, Y_{head}$ and $Y_{tail}$ by $P_i$
Compose $P_i' = (S_{good}, Y_{head}, x, S_{bad}, Y_{tail})$
return $P_i'$.
```

Algorithm 2 VoterManipulation function: Case-B

```
Require:
    Common Givens
    Holds $c_j \succ pw_{i_1}$, $c_k = pw_{i_1}$
Set $O \leftarrow \emptyset$
Set $Z \leftarrow [c_j, ..., c_k]$
for $d \in [1 : length(Z)]$ do
    $X_{good} \leftarrow \{c \in C | c \succ z_d\} \setminus (E_{i_{down}} \cup \{c_j\})$
    $X_{bad} \leftarrow \{c \in C | c \succ z_d\} \cap E_{i_{down}}$
    $Y_{good} \leftarrow \{c \in C | z_d \succ c\} \vee c = z_j\} \setminus \left(E_{i_{up}} \cup \{c_k\}\right)$
    $Y_{bad} \leftarrow \{c \in C | z_d \succ c\} \vee c = z_j\} \cap E_{i_{up}}$
    Order $X_{good}$, $X_{bad}$, $Y_{good}$ and $Y_{bad}$ by $P_i$
    Compose $P_i' = (X_{good}, Y_{bad}, y, x, X_{bad}, Y_{good})$
    Set $O \leftarrow O \cup \{P_i'\}$
end for
Set $P_{i_{best}} = \arg\min_{P_i \in O} d_{swap}(P_i, P_i')$
return $P_{i_{best}}$
```

Algorithm 1 identifies that since c_j is not a Possible Winner and $c_k \in PW$, it is better to switch between them. The algorithm first checks whether any of voter v_i's previously declared preferences in Q_i, are inconsistent with switching the order of c_j and c_k. If it is consistent, it shifts c_j and all candidates that are declared in Q_i as candidates with a lower preference than c_j, to be positioned as less preferred than c_k in the new preference profile P_i'.

Algorithm 3 VoterManipulation function: Case-C

```
Require:
    Common Givens
    Holds $c_j \succ pw_{i_1}$, $c_k \in E_i \setminus PW$
 1: Let $pw_l \in PW$ be closest to $c_k$ from above
 2: if $E_{i_{down}} \cap SG(pw_{i_1}, pw_{l+1}, P_i) \neq \emptyset$ then
 3:     return $P_i$
 4: end if
 5: Set $O \leftarrow \emptyset$
 6: Set $Z \leftarrow \ [pw_{i_l}, ..., c_k]$
 7: for $d \in [1 : length(Z)]$ do
 8:     $X_{good} \leftarrow \{c \in C | c \succ z_d\} \setminus (E_{i_{down}} \cup \{c_j\})$
 9:     $X_{bad} \leftarrow \{c \in C | c \succ z_d\} \cap E_{i_{down}}$
10:     $Y_{good} \leftarrow \{c \in C | z_d \succ c\} \setminus \left(E_{i_{up}} \cup \{c_k\}\right)$
11:     $Y_{bad} \leftarrow \{c \in C | z_d \succ c\} \vee c = z_d\} \cap \ E_{i_{up}}$
12:     Order $X_{good}$, $X_{bad}$, $Y_{good}$ and $Y_{bad}$ by $P_i$
13:     Compose $P_i' = (X_{good}, Y_{bad}, y, x, X_{bad}, Y_{good})$
14:     Set $O \leftarrow O \cup \{P_i'\}$
15: end for
16: Set $P_{i_{best}} = \arg\min_{b \in O} d_{swap}(P_i, P_i')$
17: return $P_{i_{best}}$
```

Algorithm 2 and Algorithm 3 attempt the same shift of priority for c_j and relevant candidates. However, the location of c_k in the current profile P_i makes the process more intricate. We therefore demonstrate the operation of Algorithm 3 with an example. Consider a situation where the query is: $\langle v_1 \langle c_5, c_2 \rangle \rangle$ and the current preference profile of v_1 has the following order of preferences:

$$P_1 = [c_5, pw_{1_1}, c_1, pw_{1_2}, c_3, c_4, c_6, c_8, c_7, c_2, pw_{1_3}]$$

This means that: $c_5 \succ c_2$ and the ordered vector of Possible Winners for v_1 is: $PW_1 = [pw_{1_1}, pw_{1_2}, pw_{1_3}]$. Assume that $Q_1 = \{(c_5 \succ c_1)\}$. Then $E_{1_{down}} = [c_1]$. Hence, the preference profile remains unchanged, since the intersection between $E_{1_{down}}$ and $[pw_{1_1}, c_1, pw_{1_2}]$ is not empty.

Let us now assume a different scenario, where Q_1 is more complex: $Q_1 = \{(c_5 \succ c_4), (c_5 \succ c_8), (c_6 \succ c_2)\}$. The safety check in line 2 of the algorithm is now satisfied, since $E_{1_{down}} = [c_4, c_8, c_2]$ and $E_1 = [pw_{1_1}, c_1, pw_{1_2}]$. The algorithm now searches for the best new preference profile P_1', where v_1 will manipulate and declare that $c_2 \succ c_5$. First, different positions for the sub-sequence $[c_2, c_5]$ are considered. The possible positions are all the positions that are above c_2 and below the possible winner that is closest to c_2 from above, in this case pw_{1_2}. Thus in our case the possible positions are: $[c_3, c_4, c_6, c_8, c_7, c_2]$. If $[c_2, c_5]$ appears before c_3 in the new preference profile, then $c_{5_{good}} = [pw_{1_1}, c_1, pw_{1_2}]$ will be placed before c_5. Similarly, $c_{2_{good}} = [c_3, c_4, c_8, c_7, pw_{1_3}]$ will be placed after c_2. We now check for inconsistencies with Q_1. If c_5 is moved to be preferred over c_3 and is positioned directly before c_3, the set $c_{5_{bad}}$ is empty, thus this move is consistent with Q_1. However, $c_{2_{bad}} = [c_6]$ meaning we cannot move c_6 to be preferred over c_2, since $(c_6 \succ c_2) \in Q_1$.

As a result, the generated new preference profile is:

$$P'_{1_a} = [pw_{1_1}, c_1, pw_{1_2}, c_6, c_2, c_5, c_3, c_4, c_8, c_7, pw_{1_3}]$$

In P'_{1_a}, c_5 is lower in 5 position than it was in P_1, c_3 and c_4 are lower in 2 positions and c_6 and c_2 are lower in 1 position. Hence, the swap distance is: $d_{swap}(P'_{1_a}, P_1) = 5 + 2 + 2 + 1 + 1 = 11$.

Another option is to create a preference profile where $[c_2, c_5]$ appears before $[c_4, c_6, c_8]$ or before $[c_7]$. Considering the constraints induced by $c_{5_{bad}}$ and $c_{2_{bad}}$, the new preference profile is:

$$P'_{1_b} = (pw_{1_1}, c_1, pw_{1_2}, c_3, c_6, c_2, c_5, c_4, c_8, c_7, pw_{1_3})$$

This results in swap distance of $d_{swap}(P'_{1_b}, P_1) = 10$.

The last option is to position c_5 directly after c_2, resulting in a preference profile:
$P'_{1_c} = (pw_{1_1}, c_1, pw_{1_2}, c_3, c_6, c_7, c_2, c_5, c_4, c_8, pw_{1_3})$
with a swap distance: $d_{swap}(P'_{1_c}, P_1) = 12$.

Out of the set of locally dominant alternative preference profiles $O = \{P'_{1_a}, P'_{1_b}, P'_{1_c}\}$ the profile with the smallest swap distance is selected: $P'_1 = P'_{1_b}$.

The resulting new preference profile P'_1 satisfies both $VM - Condition - 1$ and $VM - Condition - 2$: (1) the profile is a local dominant over P_1, since both pw_1 and pw_2 now have a higher relative preference; and (2) the profile is the most similar to P_1 according to the swap-distance measure.

Theorem 2. *Assume Common Givens. Algorithms 1-3 and their symmetric counterparts for Family II always find a new preference profile P'_i that satisfies both $VM - Condition - 1$ and $VM - Condition - 2$, if such a profile exists. The computational complexity of finding the profile is $O(m^3)$.*

Having mapped all possible cases of manipulation, we now construct a voting center which is less prone to manipulation by prevention: the center actively selects queries which will not enable the voter to manipulate. Specifically, we define a Careful Voting Center as a Center that selects queries where both candidates are in the Possible Winners set.

5 Experimental Validation

In order to evaluate the manipulation impact on the preference elicitation process, we compared manipulative voters to truthful voters in a Careful and in a Naive Voting Center setting and examined:(1) How often manipulations occur; and (2) The manipulation impact on the final result. Experiments were performed on the real-world Sushi dataset [Kamishima *et al.*, 2005]. The dataset contains 5000 preference rankings over 10 kinds of sushi. The dataset was used to generate responses to elicitation queries, assuming a Borda voting rule. A random set of preference profiles (P) was chosen out of the Sushi dataset, according to the amount of users in the experiment setting. For each experiment setting, 20 sets of random profiles were evaluated. For each set of profiles, the experiment was conducted 40 times. Thus we reach an amount of 800 experiments for each experiment setting.

Algorithms that perform preference elicitation in iterations can be found in [Naamani-Dery *et al.*, 2014; 2015; Lu and Boutilier, 2011; 2013]. In this paper we use the Expected Scored (ES) algorithm found in [Naamani-Dery *et al.*, 2015]. The ES algorithm selects a voter-item-item pair where one of the items is the item with the current maximum score. This algorithm is publicly available whereas some others are used commercially and cannot be tampered with [Lu and Boutilier, 2011; 2013]. As a baseline we used an algorithm which randomly chooses the next query (denoted as RANDOM). Each algorithm (ES and RANDOM) was studied in three states: (a) the voters always answer truthfully (ES+T, RANDOM+T), (b) the voters attempt to manipulate (ES+M, RANDOM+M), and (c) manipulative voters with a Careful Voting Center (Careful-ES+M, Careful-RANDOM+M). For RANDOM we used a Center that selects queries where both candidates are in the Possible Winners set or when both candidates are not in the Possible Winners set. If none such queries exist (since they have been previously used), the Voting Center stops being careful. The amount of candidates was set to 10, which is the maximum amount of candidates in the Sushi dataset. The amount of voters was varied on a range of 10, 50, 100, 150, 200, 250.

In order to conclude which algorithm performs best over multiple datasets, we followed a robust non-parametric procedure proposed by [García *et al.*, 2010]. 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.

5.1 Results

Manipulation impact on the final result: Experiments show that when the amount of voters is 50-250 the manipulations do not alter the final result, i.e., in all experiments for all three states of each algorithm, the same necessary winner was found. However, for 10 voters, the outcome changes in 4.25%, 0.37%, 15%, 0.37% of the experiments for ES+M, Careful-ES+M, RANDOM+M, Careful-RANDOM+M respectively.

Manipulation rate in a Careful and a regular Voting Center: Manipulations occur in a small portion of the queries. The manipulation rate is a mere 0.001–0.01. Figure 1 illustrates the average manipulation rate per experiment for a changing number of voters. Fewer manipulations occur at the ES+M and Careful-ES+M than at the RANDOM algorithms. The ES algorithm variations reach the stopping condition of the algorithm earlier and thus less queries and manipulations are used. Furthermore, Careful-ES+M reduces the number of manipulations since it tends to select queries where both items are possible winners, i.e., queries in which manipulations are not possible. The algorithms are significantly different according to a Friedman test with at a 95% confidence level. The Careful-ES+M Center significantly reduces the manipulation percentage when compared with the regular ES+M algorithm, at a 95% confidence level according to the Bonefferi-Dunn test (Figure 2 shows a zoom on these two variations). The RANDOM+M and Careful-RANDOM+M algorithms are not significantly different.

Manipulations impact on the number of iterations: Figures 3 and 4 illustrate the average percentage of the dataset queried until a necessary winner is found, for a changing number of voters. All variations of the ES algorithm (Fig-

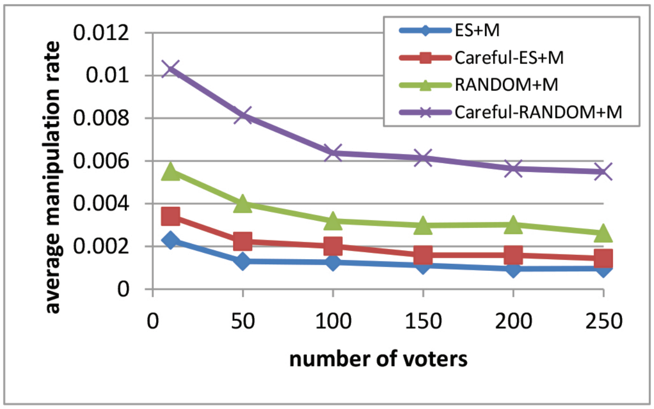

Figure 1: Manipulation percentage under a changing number of voters.

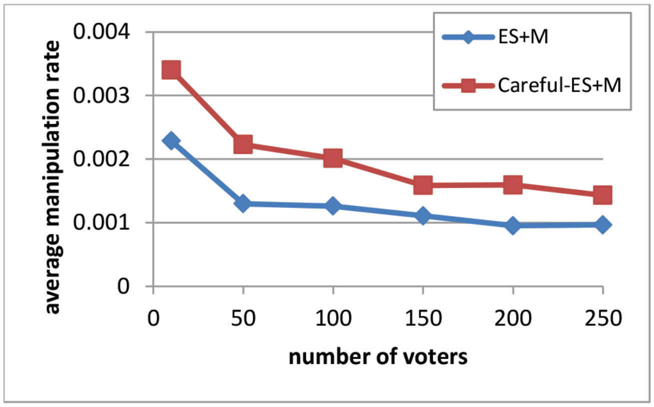

Figure 2: Manipulation percentage under a changing number of voters: zoom in on the significant difference between ES+M and Careful-ES+M.

ure 3) significantly outperform all variations of the RANDOM algorithm (Figure 4). There is no significant difference between ES+M and Careful-ES+M. However, there is a significant difference to the favor of an ES+T algorithm with no manipulations: when no manipulations occur, the result is reached faster. A possible explanation is that the Voting Center is confused by the manipulations; although the manipulations occur in a small percentage of the queries, they do impact the algorithms performance.

Surprisingly, for the RANDOM algorithms the trend is opposite: the non-manipulating algorithm performs worse than the manipulating one. It seems that when the queries are chosen at Random, the manipulations actually assist the Voting Center. This definitely requires further investigation.

We can conclude that in practice, manipulations do not pose a big threat on an incremental iterative voting process when the voters submit one preference at a time, since they rarely modify the outcome. A Careful Voting Center can reduce the manipulation rate, but the tradeoff is that in turn, there is an increase in the amount of queries needed in order to end the iterative process and find a winner.

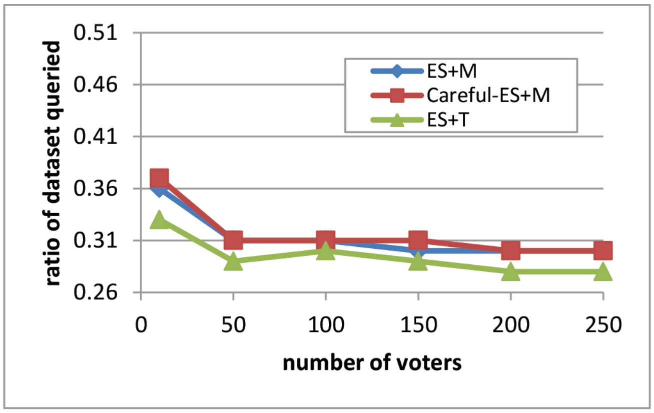

Figure 3: ES based algorithms: the percentage of the dataset queried for a changing number of voters.

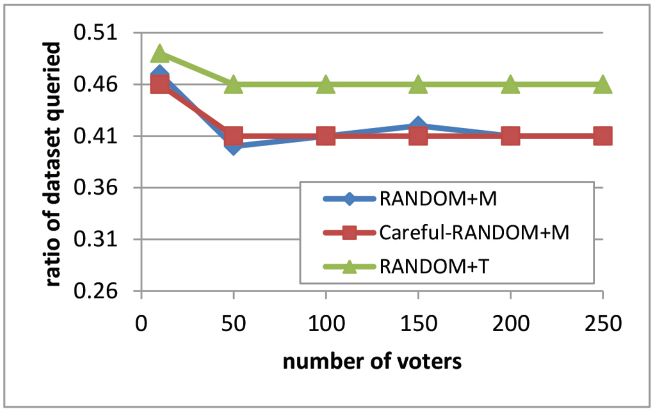

Figure 4: Random based algorithms: the percentage of the dataset queried for a changing number of voters.

6 Conclusions and Future Work

In this work we have developed and studied a novel combination of two iterative processes found in social choice: iterative preference elicitation using a Voting Center, and the manipulative modification of preferences by voters in Iterative Voting. The design of the former intends to reduce the amount of query requests the Voting Center sends in order to obtain the elections outcome, but assumes voters will reveal their true preference. The latter, on the other hand, presumes that voters may change their preferences. We illustrated how a voter may attempt to manipulate a Voting Center. We provided a set of algorithms to detect and exploit manipulation opportunities that would drive the Voting Center to declare an election outcome that is more beneficial to the manipulating voter. We experimentally showed that manipulation has only a slight impact on the outcome of an election when advanced elicitation schemes such as the one found in [Naamani-Dery *et al.*, 2015] are used. The surprising finding of our experiments is that manipulation does not necessarily have a negative effect. In fact, for a Voting Center with random query generation, manipulation attempts speed-up the convergence process. Nonetheless, our manipulation detection algorithms allow us to build a Careful Voting Center that avoids manipulable queries. Our experiments show that a Careful Voting Center is effective. In the future, we would like to drill down and study cases with less than 50 voters, since our finding imply that these cases are more manipulation prone. We would like to investigate in what settings and which manipulation scenarios lead to a change in the elections outcome.

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.

[Conitzer and Sandholm, 2005] V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In *ACM EC*, pages 78–87, 2005.

[Conitzer *et al.*, 2011] V. Conitzer, T. Walsh, and L. Xia. Dominating manipulations in voting wih partial information. In *AAAI*, pages 638–643, 2011.

[Ding and Lin, 2013] N. Ding and F. Lin. Voting with partial information: What questions to ask? In *AAMAS*, pages 1237–1238, 2013.

[Farquharson, 1969] R. Farquharson. *Theory of Voting*. Yale University Press, 1969.

[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, 2010.

[Gibbard, 1973] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–602, 1973.

[Grandi *et al.*, 2013] U. Grandi, A. Loreggia, F. Rossi, K. B. Venable, and T. Walsh. Restricted manipulation in iterative voting: Condorcet efficiency and borda score. In *ADT*, pages 181–192, 2013.

[Kalech *et al.*, 2011] M. Kalech, S. Kraus, G. A. Kaminka, and C. V. Goldman. Practical voting rules with partial information. *JAAMAS*, 22(1):151–182, 2011.

[Kamishima *et al.*, 2005] T. Kamishima, H. Kazawa, and S. Akaho. Supervised ordering - an empirical survey. In *IEEE ICDM*, pages 219–228, 2005.

[Kendall, 1938] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[Konczak and Lang, 2005] K. Konczak and J. Lang. Voting procedures with incomplete preferences. *Workshop on Advances in Preference Handling @ IJCAI*, 20, 2005.

[Kukushkin, 2011] N. S. Kukushkin. Acyclicity of improvements in finite game forms. *Int. J. Game Theory*, 40(1):147–177, 2011.

[Laffont, 1987] J.-J. Laffont. Incentives and the allocation of public goods. In *Handbook of Public Economics*, volume 2, chapter 10, pages 537–569. Elsevier, 1987.

[Lev and Rosenschein, 2012] O. Lev and J. S. Rosenschein. Convergence of iterative voting. In *AAMAS*, pages 611–618, 2012.

[Lu and Boutilier, 2011] T. Lu and C. Boutilier. Robust approximation and incremental elicitation in voting protocols. In *IJCAI*, pages 287–293, 2011.

[Lu and Boutilier, 2013] T. Lu and C. Boutilier. Multi-winner social choice with incomplete preferences. In *IJCAI*, pages 263–270, 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.

[Naamani-Dery *et al.*, 2014] L. Naamani-Dery, M. Kalech, L. Rokach, and B. Shapira. Reaching a joint decision with minimal elicitation of voter preferences. *Inf. Sci.*, 278:466–487, 2014.

[Naamani-Dery *et al.*, 2015] L. Naamani-Dery, I. Golan, M. Kalech, and L. Rokach. Preference elicitation for group decisions using the borda voting rule. *Group Decision and Negotiation*, pages 1–19, 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.

[Pini *et al.*, 2007] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *IJCAI*, pages 1464–1469, 2007.

[Rabinovich *et al.*, 2015] Z. Rabinovich, S. Obraztsova, O. Lev, E. Markakis, and J. S. Rosenschein. Analysis of equilibria in iterative voting schemes. In *AAAI*, pages 1007–1013, 2015.

[Reijngoud and Endriss, 2012] A. Reijngoud and U. Endriss. Voter response to iterated poll information. In *AAMAS*, pages 635–644, 2012.

[Reyhani and Wilson, 2012] R. Reyhani and M. Wilson. Best reply dynamics for scoring rules. In *ECAI*, pages 672–677, 2012.

[Satterthwaite, 1975] M. A. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *J. of Econ. Theory*, 10(2):187–217, 1975.

[Walsh, 2007] T. Walsh. Uncertainty in preference elicitation and aggregation. In *AAAI*, pages 3–8, 2007.

[Xia and Conitzer, 2011] L. Xia and V. Conitzer. Determining possible and necessary winners given partial orders. *JAIR*, 41:25–67, 2011.

[Zou *et al.*, 2015] J. Zou, R. Meir, and D. Parkes. Strategic voting behavior in doodle polls. In *ACM Conference on CSCW)*, 2015. to appear.