**Louis Schlessinger**

**438259**

**4/25/17**

**CSE 516**

Homework 3

**Problem 1:**

**a)**

Assume the following definitions: A: (P, ~S) B: (~P, S) C: (~P, ~S) D: (P, S)

P: pool ~P: no pool

S: squash S: no squash

Assuming everyone is *hopeful*, ballot 1 votes (P, ~S), ballot 2 votes (~P, S), ballot 3 votes (P, S).

The vote for P is: 3 yes, 2 no. the vote for S is: 3 yes, 2 no.

This results in **(P, S)**, which is a socially bad result because 4 / 5 people wanted either P or S, but not both. Those 4 people had (P, S) at the bottom of their preferences.

**b)**

The Borda count results in: **A: 17, B: 16, C: 9, D: 8**.

So A wins, which is a good result in the context of the overall preferences.

|  |  |  |  |
| --- | --- | --- | --- |
| Ballot label | X | Y | Z |
| Number of voters | 2 | 2 | 1 |
| 1st choice | A | B | D |
| 2nd choice | B | A | A |
| 3rd choice | C | C | B |
| 4th choice | D | D | C |

**c)**

The Borda count method doesn’t guarantee that the candidate with the majority of first-place votes always wins.

Another problem is that if there is a candidate that would win when faced head-to-head with another candidate, she won’t always win.

**d)**

Assuming hopeful voting in terms of as-yet-undetermined outcomes, there are two voting order cases we must account for.

Case 1: Assume vote on P then S

Ballot X voters reason the following way:

* If the outcome of the first vote is P, then ballot Y voters vote for ~S and ballot Z voters vote for S
* If outcome for first vote is ~P, ballot Y voters vote S and ballot Z voters vote S, so the outcome is (~P, S)

Ballot Y voters:

* If the outcome for first vote is P: X ballot voters will vote for ~S and Z ballot voters vote for S.
* If outcome of first vote is ~P: X ballot voters and Z ballot voters vote for S, so the outcome will be (~P, S)

Ballot Z voters:

* If the outcome for first vote is P: X ballot voters vote for ~S and Y ballot voters vote for S.
* If outcome of first vote is ~P: X ballot voters vote for S and Y ballot voters vote for S, so the outcome will be (~P, S)

Case 2: assume vote on S then P

Ballot X voters:

* If first vote is S: ballot Y voters vote ~P and ballot Z voters vote P
* If first vote is ~S: ballot Y voters vote P and ballot Z voters vote P, so the outcome is (P, ~S)

Ballot Y voters:

* If first vote is S: ballot X voters vote ~P and ballot Z voters vote P
* If first vote is ~S: ballot X voters vote P and ballot Z voters vote P, so the outcome is (P, ~S)

Ballot Z voters:

* If first vote is S: ballot X voters vote ~P and ballot Y voters vote ~P, so the outcome is (~P, S)
* If first vote is ~S: ballot X voters vote P and ballot Y voters vote P, so the outcome is (P, ~S)

In summary, if the first vote is on P, the outcome will be **(P, S).** If the first vote is on S, the outcome will be **(~P, S)**.

There are some strategic issues. If the ballot Z voter is rational and has information on other voters’ preference, and if the first vote is on S, then she could vote for ~S, which would end up changing the outcome to (P, ~S), which is a better outcome for her. Another strategic issue is that if the vote starts with P, then ballot Y voters are better off voting ~P.

The sequence of the votes as stated above determine the outcome. In either case the squash court will be built though.

**Problem 2:**

**Implementations Summary:** I created an election simulator in Python to parse the election data and simulate the election per the different voting rules. I created few different models: Candidate, Vote, and Election. I first parse the file and populated my Election object for the different methods corresponding to the different voting rules to use. For each implementation, I created a voting results array to store the final voting results for each candidate. For the following analysis, I used the Aspen City Council 2009 Election data of type ‘Order with Ties – Incomplete List’ corresponding to the file

**'voting-rules/ED-00016-00000001.toi'**

Plurality vote: I took the first choice of all voters and summed them for each candidate. I then took the candidate with the maximum number of votes to be the winner

Instant runoff: I had an outer loop that ran the ‘number of candidates – 1’ times for each runoff election and an inner loop to account for each voter. I took the candidate with the least amount of 1st place votes, and eliminated him/her, passing the updated set of candidates to next round of voting. If a voter did not have any votes for any remaining candidate, I skipped that voter. If the voter had at least one voter who had not yet been eliminated in the ballot, I took their first choice. Whoever was remaining in the runoff elections at the last iteration by definition won them.

Borda count: I iterated over each voter and increased the Borda count for each candidate according to the number of voters and the position in the preference list. I used the formula given by the Borda count, which is the product of *n – i*, where n is the number of candidates and *i* is the preference position of the candidate ranging from 0 to *n – 1*. To account for the incomplete preference ballots, I gave every unranked candidate 1 point because that is how many points they would get for being ranked in the last position in someone’s ballot. I assumed the candidate with the highest Borda count won the election.

Approval rating: For each vote, I incremented the candidates total number of times appearing on a ballot by 1. This method was similar to the plurality voting rule’s implementation. The only difference was that on the approval rating, we simply used the whole preference list instead of only looking at the first preference in the plurality vote. I took the candidate with the highest approval rating to win the election.

**Results**:

Plurality vote:

**Derek Johnson (8)** won the election with *465* votes

Jackie Kasabach (1) had 249 votes

Jack Johnson (2) had 456 votes

Adam Frisch (3) had 418 votes

Torre (4) had 385 votes

Michael Behrendt (5) had 354 votes

Jason Lasser (6) had 27 votes

Michael Wampler (7) had 95 votes

Derek Johnson (8) had 465 votes

Brian D. Speck (9) had 38 votes

Write In 1 (10) had 0 votes

Write In 2 (11) had 0 votes

Instant runoff:

**Torre (4)** won the election with *990* votes

Write In 1 (10) was eliminated from the election with 0 votes

Write In 2 (11) was eliminated from the election with 0 votes

Jason Lasser (6) was eliminated from the election with 27 votes

Michael Wampler (7) was eliminated from the election with 28 votes

Derek Johnson (8) was eliminated from the election with 40 votes

Brian D. Speck (9) was eliminated from the election with 49 votes

Jackie Kasabach (1) was eliminated from the election with 352 votes

Jack Johnson (2) was eliminated from the election with 469 votes

Adam Frisch (3) was eliminated from the election with 584 votes

Michael Behrendt (5) was eliminated from the election with 936 votes

Borda count:

**Derek Johnson (8)** won the election with a borda count of *15943*

Jackie Kasabach (1) had a borda count of 13584

Jack Johnson (2) had a borda count of 13531

Adam Frisch (3) had a borda count of 14338

Torre (4) had a borda count of 15806

Michael Behrendt (5) had a borda count of 15134

Jason Lasser (6) had a borda count of 7899

Michael Wampler (7) had a borda count of 10698

Derek Johnson (8) had a borda count of 15943

Brian D. Speck (9) had a borda count of 8829

Write In 1 (10) had a borda count of 2647

Write In 2 (11) had a borda count of 2551

Approval rating:

**Torre (4)** won the election by appearing on *1744* ballots

Jackie Kasabach (1) had 1525 votes

Jack Johnson (2) had 1549 votes

Adam Frisch (3) had 1551 votes

Torre (4) had 1744 votes

Michael Behrendt (5) had 1625 votes

Jason Lasser (6) had 1024 votes

Michael Wampler (7) had 1303 votes

Derek Johnson (8) had 1696 votes

Brian D. Speck (9) had 1137 votes

Write In 1 (10) had 31 votes

Write In 2 (11) had 11 votes

This experience tells me that slight ambiguities in implementations of voting rules such as handling incomplete preferences in the Borda count can have a big impact on the result. I also learned that each of the voting rules such as a simple plurality vote have some big drawbacks, especially as the number of candidates increase. It seemed like instant runoff voting gave the fairest and best social outcome. One negatives of it though is that is a little more expensive to run.

**Problem 3:**

Let be agent 's bid, which could be a function of

Let

Consider each alternative separately.

**Alternative 1**:

The payment for employee is

Therefore, the total payment for alternative 1 is

This is incentive compatible because the payment for each agent does not depend on her own bid, which is the same as in the VCG mechanism.

This mechanism does not make any positive transfers because it does not need to pay anything to the employees (beyond the money from the first stage).

It is individually rational because each payment is greater than or equal to zero.

Pf:

Assume

For all , , which means

So, for every employee ,

The VCG payment is , so the fraction of it given back is the agent's payment above divided by the VCG payment. The total fraction is the same, but just using the total payment for alternative 1.

Additionally, for **alternative 2**,

Let be the th highest bid from an agent other than

Assume

The payment for employee 𝑖 is

Therefore, the total payment is

Alternative 2 is also incentive compatible for the same reason as alternative 1.

It does not make any positive transfers for the same reason as in alternative 1.

Each agent always receives a payment that is 0, which make it incentive compatible

Pf:

Assume

Assume

For all , a\_i≥0, , which means

So, for every employee ,

The VCG payment is , so the fraction of it given back is the agent's payment above divided by the VCG payment. The total fraction is the same, but just using the total payment for alternative 2.

**Discussion:**

Neither mechanism is always better than the other, in terms of how much it gives back. If n is big, then the alternative 2 is better. In the best case for both, it is possible to achieve 100% payback to the employees.

The total money given back in alternative 1 is:

If then it is the same as the total VCG payback of .

In alternative 2, the total given back is

If it also equals the total VCG payback of

In the worst case for alternative 1, the total payment back to the employees is the same as in the best case.

If the payback is

In the worst case for alternative 2, the total payment back to the employees is the same as in the best case above.

If the payback is

We can simply divide by the VCG payment to get the relative difference of payments for each alternative.

By inspection, it seems that the alternative 2 has a better worst-case performance than alternative 1, if the goal is to pay as much back as possible. As *n* gets much bigger than *m*, you would want to use alternative 2.