Voting and Mechanism Design

José M Vidal

Department of Computer Science and Engineering, University of South Carolina

March 26, 2010

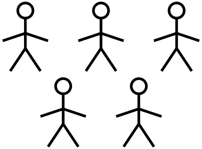
Abstract

Voting, Mechanism design, and distributed algorithmics mechanism design. Chapter 8.

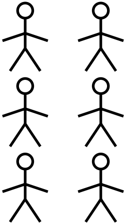
Why Vote?

- Common way of aggregating agents' preferences.
- Well understood.
- But, centralized.

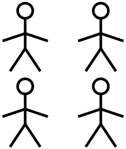
The Voting Problem



Beer
Wine
Milk



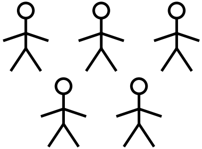
Milk
Wine
Beer



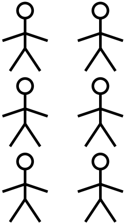
Wine
Beer
Milk

	Beer	Wine	Milk
Plurality			
Runoff			
Pairwise			

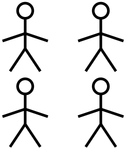
The Voting Problem



Beer
Wine
Milk



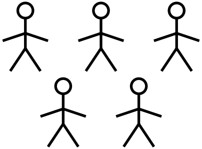
Milk
Wine
Beer



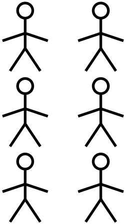
Wine
Beer
Milk

	Beer	Wine	Milk
Plurality	5	4	6
Runoff			
Pairwise			

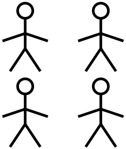
The Voting Problem



Beer
Wine
Milk



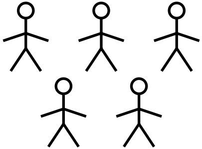
Milk
Wine
Beer



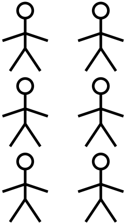
Wine
Beer
Milk

	Beer	Wine	Milk
Plurality	5	4	6
Runoff	5,9	4	6,6
Pairwise			

The Voting Problem

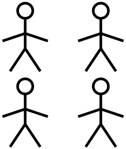


Beer
Wine
Milk



Milk
Wine
Beer

	Beer	Wine	Milk
Plurality	5	4	6
Runoff	5,9	4	6,6
Pairwise	1	2	0



Wine
Beer
Milk

Symmetry

- Reflectional symmetry: If one agent prefers A to B and another one prefers B to A then their votes should cancel each other out.
- Rotational symmetry: If one agent prefers A,B,C and another one prefers B,C,A and another one prefers C,A,B then their votes should cancel out.

Symmetry

- Reflectional symmetry: If one agent prefers A to B and another one prefers B to A then their votes should cancel each other out.
- Rotational symmetry: If one agent prefers A,B,C and another one prefers B,C,A and another one prefers C,A,B then their votes should cancel out.

- Plurality vote violates reflectional symmetry, so does runoff voting.
- Pairwise comparison violates rotational symmetry.

Borda Count

Jean-Charles de Borda. 1733–1799.

1. With x candidates, each agent awards x to points to his first choice, $x - 1$ points to his second choice, and so on.
2. The candidate with the most points wins.

Borda satisfies both reflectional and rotational symmetry.

Formalization

- There is a set of A agents, and O outcomes.
- Each agent i has a preference function $>_i$ over the set of outcomes.
- Let $>^*$ be the global set of social preferences. That is, what we want the outcome to be.

Definition (Desirable Voting Outcome Conditions)

1. $>^*$ exists for all possible inputs $>_i$

Definition (Desirable Voting Outcome Conditions)

1. $>^*$ exists for all possible inputs $>_i$
2. $>^*$ exists for every pair of outcomes

Definition (Desirable Voting Outcome Conditions)

1. $>^*$ exists for all possible inputs $>_i$
2. $>^*$ exists for every pair of outcomes
3. $>^*$ is asymmetric and transitive over the set of outcomes

Definition (Desirable Voting Outcome Conditions)

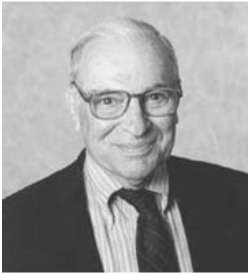
1. $>^*$ exists for all possible inputs $>_i$
2. $>^*$ exists for every pair of outcomes
3. $>^*$ is asymmetric and transitive over the set of outcomes
4. $>^*$ should be Pareto efficient.

Definition (Desirable Voting Outcome Conditions)

1. $>^*$ exists for all possible inputs $>_i$
2. $>^*$ exists for every pair of outcomes
3. $>^*$ is asymmetric and transitive over the set of outcomes
4. $>^*$ should be Pareto efficient.
5. The scheme used to arrive at $>^*$ should be independent of irrelevant alternatives.

Definition (Desirable Voting Outcome Conditions)

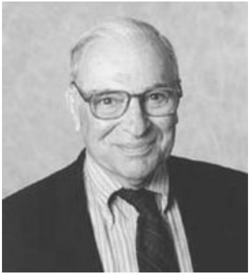
1. $>^*$ exists for all possible inputs $>_i$
2. $>^*$ exists for every pair of outcomes
3. $>^*$ is asymmetric and transitive over the set of outcomes
4. $>^*$ should be Pareto efficient.
5. The scheme used to arrive at $>^*$ should be independent of irrelevant alternatives.
6. No agent should be a dictator in the sense that $>^*$ is always the same as $>_i$, no matter what the other $>_j$ are.



Kenneth Arrow

Theorem (Arrow's Impossibility)

There is no social choice rule that satisfies the six conditions.



Kenneth Arrow

Theorem (Arrow's Impossibility)

There is no social choice rule that satisfies the six conditions.

- Plurality voting relaxes 3 and 5. Adding a third candidate can wreak havoc.
- Pairwise relaxes 5.
- Borda violates 5.

Borda Example

1. $a > b > c > d$
2. $b > c > d > a$
3. $c > d > a > b$
4. $a > b > c > d$
5. $b > c > d > a$
6. $c > d > a > b$
7. $a > b > c > d$

Borda Example

1. $a > b > c > d$
2. $b > c > d > a$
3. $c > d > a > b$
4. $a > b > c > d$
5. $b > c > d > a$
6. $c > d > a > b$
7. $a > b > c > d$

1. c gets 20 points
2. b gets 19 points
3. a gets 18 points
4. d gets 13 points

Borda Example

Let's get rid of d.

1. $a > b > c > d$
2. $b > c > d > a$
3. $c > d > a > b$
4. $a > b > c > d$
5. $b > c > d > a$
6. $c > d > a > b$
7. $a > b > c > d$

Borda Example

Let's get rid of d.

1. $a > b > c$
2. $b > c > a$
3. $c > a > b$
4. $a > b > c$
5. $b > c > a$
6. $c > a > b$
7. $a > b > c$

Borda Example

Let's get rid of d.

1. $a > b > c$
2. $b > c > a$
3. $c > a > b$
4. $a > b > c$
5. $b > c > a$
6. $c > a > b$
7. $a > b > c$

1. a gets 15 points
2. b gets 14 points
3. c gets 13 points

Voting Summary

1. Use Borda count whenever possible.
2. Practically, Borda requires calculating all preferences: often computationally hard.

Painting the House

Name	Wants house painted?
Alice	Yes
Bob	No
Caroline	Yes
Donald	Yes
Emily	Yes

Formal Definition

- Each agent i has a type $\theta_i \in \Theta_i$ which is private.

Formal Definition

- Each agent i has a type $\theta_i \in \Theta_i$ which is private.
- $\theta = \{\theta_1, \theta_2, \ldots, \theta_A\}$.

Formal Definition

- Each agent i has a type $\theta_i \in \Theta_i$ which is private.
- $\theta = \{\theta_1, \theta_2, \ldots, \theta_A\}$.
- The protocol results in some outcome $o \in O$.

Formal Definition

- Each agent i has a type $\theta_i \in \Theta_i$ which is private.
- $\theta = \{\theta_1, \theta_2, \ldots, \theta_A\}$.
- The protocol results in some outcome $o \in O$.
- Each agent i gets a value $v_i(o, \theta_i)$ for outcome o.

Formal Definition

- Each agent i has a type $\theta_i \in \Theta_i$ which is private.
- $\theta = \{\theta_1, \theta_2, \ldots, \theta_A\}$.
- The protocol results in some outcome $o \in O$.
- Each agent i gets a value $v_i(o, \theta_i)$ for outcome o.
- The social choice function $f(\theta)$ tells us the outcome we want to achieve. For example,

$$f(\theta) = \arg\max_{o \in O} \sum_{i=1}^{n} v_i(o, \theta_i) \tag{1}$$

Painting the House

Name	Type (θ_i)	$v_i(Paint, \theta_i)$	$v_i(NoPaint, \theta_i)$
Alice	WantPaint	10	0
Bob	DontNeedPaint	0	0
Caroline	WantPaint	10	0
Donald	WantPaint	10	0
Emily	WantPaint	10	0

Painting the House

Name	Type (θ_i)	$v_i(Paint, \theta_i)$	$v_i(NoPaint, \theta_i)$
Alice	WantPaint	10	0
Bob	DontNeedPaint	0	0
Caroline	WantPaint	10	0
Donald	WantPaint	10	0
Emily	WantPaint	10	0

- Try this: Everyone votes Y/N. If majority votes Y then paint house. All pay 1/5 of cost (4 each).

Painting the House

Name	Type (θ_i)	$v_i(Paint, \theta_i)$	$v_i(NoPaint, \theta_i)$
Alice	WantPaint	10	0
Bob	DontNeedPaint	0	0
Caroline	WantPaint	10	0
Donald	WantPaint	10	0
Emily	WantPaint	10	0

- Try this: Everyone votes Y/N. If majority votes Y then paint house. All pay 1/5 of cost (4 each).
- Bob must pay for a paint job he didn't want.

Painting the House

Name	Type (θ_i)	$v_i(Paint, \theta_i)$	$v_i(NoPaint, \theta_i)$
Alice	WantPaint	10	0
Bob	DontNeedPaint	0	0
Caroline	WantPaint	10	0
Donald	WantPaint	10	0
Emily	WantPaint	10	0

- Try this: Everyone votes Y/N. Split cost among those who voted Y.

Painting the House

Name	Type (θ_i)	$v_i(Paint, \theta_i)$	$v_i(NoPaint, \theta_i)$
Alice	WantPaint	10	0
Bob	DontNeedPaint	0	0
Caroline	WantPaint	10	0
Donald	WantPaint	10	0
Emily	WantPaint	10	0

- Try this: Everyone votes Y/N. Split cost among those who voted Y.
- There is an incentive for all but Bob to lie.

Definition (g Implements f)

A mechanism $g : S_1 \times \cdots \times S_A \rightarrow O$ implements social choice function $f(\cdot)$ if there is an equilibrium strategy profile $(S_1^*(\cdot), \ldots, S_A^*(\cdot))$ of the game induced by g such that $g(S_1^*(\theta_1), \ldots, S_A^*(\theta_A)) = f(\theta_1, \ldots, \theta_A)$ for all $\theta \in \Theta$.

Where we let $S_i(\theta_i)$ be agent i's strategy given that it is of type θ_i.

Definition (Dominant Strategy Equilibrium)

A strategy profile $(S_1^*(\cdot), \ldots, S_A^*(\cdot))$ of the game induced by g is a dominant strategy equilibrium if for all i and all θ_i,

$$v_i(g(s_i^*(\theta_i), s_{-i}), \theta_i) \geq v_i(g(s_i', s_{-i}), \theta_i)$$

for all $s_i' \in S_i$ and all $s_{-i} \in S_{-i}$.

Definition (g Implements f)

A mechanism $g : S_1 \times \cdots \times S_A \to O$ implements social choice function $f(\cdot)$ in dominant strategies if there is a dominant strategy equilibrium strategy profile $(S_1^*(\cdot), \ldots, S_A^*(\cdot))$ of the game induced by g such that $g(S_1^*(\theta_1), \ldots, S_A^*(\theta_A)) = f(\theta_1, \ldots, \theta_A)$ for all $\theta \in \Theta$.

Definition (Strategy-Proof)

The social choice function $f(\cdot)$ is truthfully implementable in dominant strategies (or strategy-proof) if $s_i^*(\theta_i) = \theta_i$ (for all $\theta_i \in \Theta_i$ and all i) is a dominant strategy equilibrium of the direct revelation mechanism $f(\cdot)$. That is, if for all i and all $\theta_i \in \Theta_i$,

$$v_i(f(\theta_i, \theta_{-i}), \theta_i) \geq v_i(f(\hat{\theta}_i, \theta_{-i}), \theta_i)$$

for all $\hat{\theta}_i \in \Theta_i$ and all $\theta_{-i} \in \Theta_{-i}$.

Theorem (Revelation Principle)

If there exists a mechanism g that implements the social choice function f in dominant strategies then f is truthfully implementable in dominant strategies.

Selling Example

- $\theta_i \in \Re$: types are the valuations.

Selling Example

- $\theta_i \in \Re$: types are the valuations.
- $o \in \{1, \ldots, n\}$: index of agent who gets the item.

Selling Example

- $\theta_i \in \Re$: types are the valuations.
- $o \in \{1, \ldots, n\}$: index of agent who gets the item.
- $v_i(o, \theta_i) = \theta_i$ if $o = i$, and 0 otherwise.

Selling Example

- $\theta_i \in \Re$: types are the valuations.
- $o \in \{1, \ldots, n\}$: index of agent who gets the item.
- $v_i(o, \theta_i) = \theta_i$ if $o = i$, and 0 otherwise.
- $f(\theta) = \arg\max_i(\theta_i)$

Selling Example

- $\theta_i \in \Re$: types are the valuations.
- $o \in \{1, \ldots, n\}$: index of agent who gets the item.
- $v_i(o, \theta_i) = \theta_i$ if $o = i$, and 0 otherwise.
- $f(\theta) = \arg\max_i(\theta_i)$
- Each agent gets a $p_i(o)$ so that $u_i(o, \theta_i) = v_i(o, \theta_i) + p_i(o)$.

Selling Example

- $\theta_i \in \Re$: types are the valuations.
- $o \in \{1, \ldots, n\}$: index of agent who gets the item.
- $v_i(o, \theta_i) = \theta_i$ if $o = i$, and 0 otherwise.
- $f(\theta) = \arg\max_i(\theta_i)$
- Each agent gets a $p_i(o)$ so that $u_i(o, \theta_i) = v_i(o, \theta_i) + p_i(o)$.

Set $p(o)$ such that the agent who wins must pay a tax equal to the second highest valuation. No one else pays/gets anything.

$$u_i(o, \theta_i) = \begin{cases} \theta_i - \max_{j \neq i} \theta_j & \text{if } o = i \\ 0 & \text{otherwise.} \end{cases}$$

Truth-Telling is Dominant in Vickrey Payments Example.

1. Let $b_i(\theta_i)$ be i's bid given that his true valuation is θ_i.

□

Truth-Telling is Dominant in Vickrey Payments Example.

1. Let $b_i(\theta_i)$ be i's bid given that his true valuation is θ_i.
2. Let $b' = \max_{j \neq i} b_j(\theta_j)$ be the highest bid amongst the rest.

□

Truth-Telling is Dominant in Vickrey Payments Example.

1. Let $b_i(\theta_i)$ be i's bid given that his true valuation is θ_i.
2. Let $b' = \max_{j \neq i} b_j(\theta_j)$ be the highest bid amongst the rest.
3. If $b' < \theta_i$ then any bid $b_i(\theta_i) > b'$ is optimal since

$$u_i(i, \theta_i) = \theta_i - b' > 0$$

□

Truth-Telling is Dominant in Vickrey Payments Example.

1. Let $b_i(\theta_i)$ be i's bid given that his true valuation is θ_i.
2. Let $b' = \max_{j \neq i} b_j(\theta_j)$ be the highest bid amongst the rest.
3. If $b' < \theta_i$ then any bid $b_i(\theta_i) > b'$ is optimal since

$$u_i(i, \theta_i) = \theta_i - b' > 0$$

4. If $b' > \theta_i$ then any bid $b_i(\theta_i) < b'$ is optimal since

$$u_i(i, \theta_i) = 0$$

□

Truth-Telling is Dominant in Vickrey Payments Example.

1. Let $b_i(\theta_i)$ be i's bid given that his true valuation is θ_i.
2. Let $b' = \max_{j \neq i} b_j(\theta_j)$ be the highest bid amongst the rest.
3. If $b' < \theta_i$ then any bid $b_i(\theta_i) > b'$ is optimal since

$$u_i(i, \theta_i) = \theta_i - b' > 0$$

4. If $b' > \theta_i$ then any bid $b_i(\theta_i) < b'$ is optimal since

$$u_i(i, \theta_i) = 0$$

5. Since we have that if $b' < \theta_i$ then i should bid $> b'$ and if $b' > \theta_i$ then i should bid $< b'$, and we don't know b' then i should bid θ_i.

□

Theorem (Groves-Clarke Mechanism)

If

$$f(\theta) = \arg\max_{o \in O} \sum_{i=1}^{n} v_i(o, \theta_i)$$

then calculating the outcome using

$$f(\tilde{\theta}) = \arg\max_{o \in O} \sum_{i=1}^{n} v_i(o, \tilde{\theta}_i)$$

(where $\tilde{\theta}$ *are reported types) and giving the agents payments of*

$$p_i(\tilde{\theta}) = \sum_{j \neq i} v_j(f(\tilde{\theta}), \tilde{\theta}_j) - h_i(\tilde{\theta}_{-i})$$

(where $h_i(\theta_{-i})$ *is an arbitrary function) results in a strategy-proof mechanism.*

Groves-Clarke Payments for House Painting

Name	$v_i(o,\tilde{\theta})$	$v_i(o,\theta) + \sum_{j \neq i} v_j(\tilde{\theta})$
Alice	$10 - \frac{20}{4} = 5$	
Bob	$0 - 0 = 0$	
Caroline	$10 - \frac{20}{4} = 5$	
Donald	$10 - \frac{20}{4} = 5$	
Emily	$10 - \frac{20}{4} = 5$	

Assuming all tell the truth.

Groves-Clarke Payments for House Painting

Name	$v_i(o, \tilde{\theta})$	$v_i(o, \theta) + \sum_{j \neq i} v_j(\tilde{\theta})$
Alice	$10 - \frac{20}{4} = 5$	$5 + 15 = 20$
Bob	$0 - 0 = 0$	$0 + 20 = 20$
Caroline	$10 - \frac{20}{4} = 5$	$5 + 15 = 20$
Donald	$10 - \frac{20}{4} = 5$	$5 + 15 = 20$
Emily	$10 - \frac{20}{4} = 5$	$5 + 15 = 20$

Assuming all tell the truth.

Groves-Clarke Payments for House Painting

Name	$v_i(o, \tilde{\theta})$	$v_i(o, \theta) + \sum_{j \neq i} v_j(\tilde{\theta})$
Alice	$0 - 0 = 0$	$10 + (\frac{10}{3} \cdot 3) = 20$
Bob	$0 - 0 = 0$	$0 + (\frac{10}{3} \cdot 3) = 10$
Caroline	$10 - \frac{20}{3} = \frac{10}{3}$	$10 + (\frac{10}{3} \cdot 2) = 10$
Donald	$10 - \frac{20}{3} = \frac{10}{3}$	$10 + (\frac{10}{3} \cdot 2) = 10$
Emily	$10 - \frac{20}{3} = \frac{10}{3}$	$10 + (\frac{10}{3} \cdot 2) = 10$

Alice lies.

Theorem (Vickrey-Clarke-Groves (VCG) Mechanism)

If

$$f(\theta) = \arg\max_{o \in O} \sum_{i=1}^{n} v_i(o, \theta_i)$$

then calculating the outcome using

$$f(\tilde{\theta}) = \arg\max_{o \in O} \sum_{i=1}^{n} v_i(o, \tilde{\theta}_i)$$

(where $\tilde{\theta}$ are reported types) and giving the agents payments of

$$p_i(\tilde{\theta}) = \sum_{j \neq i} v_j(f(\tilde{\theta}_{-i}), \tilde{\theta}_j) - \sum_{j \neq i} v_j(f(\tilde{\theta}), \tilde{\theta}_j)$$

(where $h_i(\theta_{-i})$ is an arbitrary function) results in a strategy-proof mechanism.

VCG Payments for House Painting

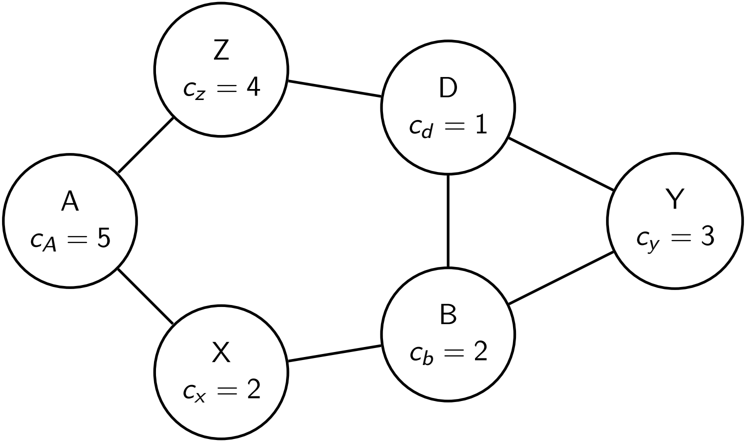
Name	$v_i(o, \tilde{\theta})$	$\sum_{j \neq i} v_j(f(\tilde{\theta}_{-i}), \tilde{\theta}_j)$	$\sum_{j \neq i} v_j(\cdot) - \sum_{j \neq i} v_j(\cdot)$
Alice	$10 - \frac{20}{4} = 5$	$(10 - \frac{20}{3}) \cdot 3 = 10$	$10 - 15 = 5$
Bob	$0 - 0 = 0$	$(10 - \frac{20}{4}) \cdot 4 = 20$	$20 - 20 = 0$
Caroline	$10 - \frac{20}{4} = 5$	$(10 - \frac{20}{3}) \cdot 3 = 10$	$10 - 15 = 5$
Donald	$10 - \frac{20}{4} = 5$	$(10 - \frac{20}{3}) \cdot 3 = 10$	$10 - 15 = 5$
Emily	$10 - \frac{20}{4} = 5$	$(10 - \frac{20}{3}) \cdot 3 = 10$	$10 - 15 = 5$

Alice tells the truth.

Distributed Algorithmic Mechanism Design

- Algorithmic mechanism design: make mechanism polynomial time
- Distributed algorithmic mechanism design: make mechanism distributed

Inter-Domain Routing Problem



Conclusion

- GC and VCG payment equations are useful ready-made solutions to many problems.

Conclusion

- GC and VCG payment equations are useful ready-made solutions to many problems.
- But, we still need more research into how to distribute the mechanisms and how to make their calculation computationally tractable.