CSE 6319 Notes 3: Mechanism Design

(Last updated 3/7/22 9:20 AM)

Karlin & Peres 10/11/12/13/14/15/16/17

Nisan 9/10/11/12/13/15/28

Roughgarden 2/3/4/5/6/7/8/9/10

3.A. Single-Peaked Preferences Over Policies (N 10.2)

n agents wish to choose a single point in a real-valued interval [0, 1].

Each agent has a single most-preferred point. Decreasing preferences to either side.

Desire a *strategy-proof* rule (dominant strategy to report truthful preferences) to decide the point.

Possible Rule Properties:

> *onto* = for every point in [0, 1], there is a strategy profile such that rule gives the point.
>
> *unanimity* = if all agents have the same peak, rule choses it
>
> *Pareto-optimal* = there is no point more preferred by all agents to the chosen point
>
> **Lemma 10.1** *Suppose f is* strategy-proof. *Then f is* onto *if and only if it is unanimous if and only if it is Pareto-optimal.*

Median-Voter Rule: Assume odd number of agents. Use median of their peaks.

> Strategy-proof (but so is choosing the *k*th highest peak)

Weighted average? (not strategy-proof in general, but could be dictatorial in the extreme)

anonymous = rule does not depend on order of input

> **Theorem 10.2** *A rule f is* strategy-proof, onto, *and* anonymous *if and only if there exist $y_1, y_2, \ldots, y_{n-1} \in [0, 1]$ such that for all $\succeq \in \mathcal{R}^n$,*

Aside:

$$f(\succeq) = \mathrm{med}\{p_1, p_2, \ldots, p_n, y_1, y_2, \ldots, y_{n-1}\}. \tag{10.1}$$

Aside: `https://en.wikipedia.org/wiki/The_Vital_Center`

Aside: `https://en.wikipedia.org/wiki/A_Theory_of_Justice`

STABLE MATCHING

Marriages and Gale-Shapley (KP 10.1-10.3; N 10.4; R 10.2-10.3)

Classical Problem Instance:

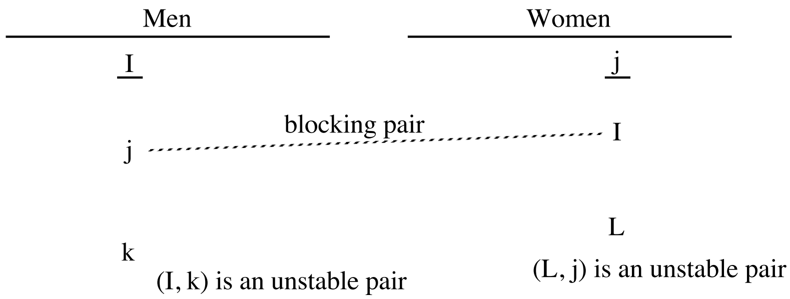
n men (A, B, C, . . .) with preference lists (ordered from most-preferred to least)

n women (1, 2, 3, . . .) with preference lists

Goal: Produce matching with n stable marriages.

A matching is *unstable* if there is a *blocking pair*:

Consider a matching with the pairs (I, k) and (L, j) based on preference lists:



I and j prefer each over their partners in the suggested matching . . . unstable situation

Applications:

Matching new M.D.s to internships (many-to-one, `http://www.nrmp.org/`)
Matching lawyers to federal clerkships (one-to-one)
Matching students to classes (many-to-many)
Centralized admissions decisions for universities (many-to-one)

Example 10.8 The preference orderings for the men and women are shown in the table below

$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$	$\succ_{w_1}$	$\succ_{w_2}$	$\succ_{w_3}$
w_2	w_1	w_1	m_1	m_3	m_1
w_1	w_3	w_2	m_3	m_1	m_3
w_3	w_2	w_3	m_2	m_2	m_2

Consider the matching $\{(m_1, w_1), (m_2, w_2), (m_3, w_3)\}$. This is an unstable matching since (m_1, w_2) is a blocking pair. The matching $\{(m_1, w_1), (m_3, w_2), (m_2, w_3)\}$, however, is stable.

Gale-Shapley (Deferred Acceptance) Algorithm:

Corresponds to most societies. (No `https://en.wikipedia.org/wiki/Sadie_Hawkins_Day`)

Men propose from the beginning of their lists.

Women always accept the first proposal, but may break the engagement later.

Example (from Sedgewick)

A	B	C	D	E	1	2	3	4	5
2	1	2	1	5	E	D	A	C	D
5	2	3	**3**	3	**A**	**E**	**D**	**B**	B
1	3	**5**	2	**2**	D	B	B	D	**C**
3	**4**	4	4	1	B	A	C	A	E
4	5	1	5	4	C	C	E	E	A

Observations:

1. There is at least one stable solution.

   (Once engaged, a woman is always engaged. A man could eventually propose to all women and can't be rejected by all of them.)

2. The set of currently engaged couples is stable.

3. As stated, Gale-Shapley algorithm gives *male-optimal* matching. Switching roles in algorithm gives *female-optimal* matching. (Example of rotations includes female-optimal matching for Sedgewick's example)

4. Truthfulness?

5. If male-optimal solution is the same as female-optimal solution, the solution is unique.

6. The order of proposals by the available men *makes no difference* in the outcome . . . leading to:

The "Rural Hospitals" Theorem: When the number of men and women differ (or preference lists may be incomplete), the set of agents included in every stable matching *is the same*.

Also possible to maintain n^2 nodes in *reduced* data structure instead of $2n^2$ nodes (i.e. each node is in two doubly-linked lists) - known as the *Extended Gale-Shapley* algorithm (MEGS = man-oriented, WEGS = woman-oriented).

Uses node deletion strategy to avoid some pain of rejection! For MEGS:

Man proposes from current beginning of reduced list . . . always accepted!

When woman receives proposal . . . she will always accept and also delete the nodes for all less-preferable men.

For the current set of engagements:

> A man is engaged to the woman at the beginning of his list.
>
> A woman is engaged to the man at the end of her list.

A	B	C	D	E	1	2	3	4	5
2	1	2	1	5	E	D	A	C	D
5	2	3	**3**	3	**A**	**E**	**D**	**B**	B
1	3	**5**	2	**2**	D	B	B	D	**C**
3	**4**	4	4	1	B	A	C	A	E
4	5	1	5	4	C	C	E	E	A

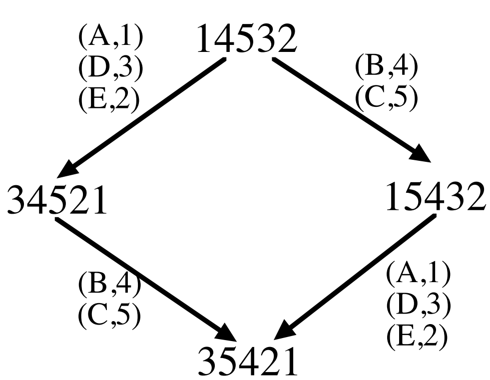
Lattice of Stable Matchings:

Given any pair of stable marriage matchings, another stable matching may be found by taking either:

1. The more preferred woman for every man (the “meet”).

2. The less preferred woman for every man (the “join”).

Mathematically, the result is a *distributive lattice*. (Also, note that any path from the male-optimal matching to the female-optimal matching includes each “rotation” exactly once.)



House Allocation, Kidney Exchange, & Top-Trading Cycles (KP 10.4; N 10.3; R 10.1)

Like Stable Marriages, two types of agents - *applicants* (with preferences) and *houses* (without)

No notion of blocking pair

Expects some applicants' preference lists to be incomplete

Usual solution concept - Pareto optimality, excludes any *Pareto improvements*:

Matching an unmatched applicant with an acceptable unmatched house

Changing an applicant to a more-preferable house without changing another applicant to a less-preferable house (or leaving out entirely)

Simplest way to find a Pareto-optimal matching is the *Serial Dictatorship Mechanism*:

1. Choose an arbitrary order for the applicants.

2. Use the order to have each applicant choose their most-preferred house among the remaining unmatched houses.

Since the method is exhaustive and an early chooser would never trade with a later chooser, must be Pareto-optimal.

Due to different orderings and incomplete preference lists, different size matchings may occur!

Finding a maximum cardinality Pareto-optimal matching:

1. Find a maximum cardinality bipartite matching (e.g. using flow techniques or CLRS problem 26-6, p. 763), but *ignore the applicants' preferences*. (The next two phases are constructive proof that a Pareto-optimal matching of this size exists. This phase is the most expensive. It is not worthwhile to clutter it with details of the two later phases.)

2. Make the matching "trade-in free" - iteratively, find a matched applicant having a more-preferred house that is available and promote the applicant.

3. Address "cyclic coalitions" in a manner similar to rotations for S.M. The technique is known as Gale's Top Trading Cycles algorithm:

    a. Delete all unmatched houses.

    b. While applicants remain:

        1. Iteratively, find applicants matched with their most-preferred house. Make each match permanent and delete the applicant and house from data structures. (This may expose other most-preferred matches in the reduced instance.)

2. Create directed graph:

   a. Vertex for each applicant.
   b. Edge from vertex for applicant x to the vertex of the applicant who is (tentatively) matched with x's <u>most-preferred</u> house.

3. Properties of the generated graph:

   a. At least one cycle. (Aside: Similar to Boruvka's MST.
   https://ranger.uta.edu/~weems/NOTES5311/NEWNOTES/notes08.pdf
   https://en.wikipedia.org/wiki/Borůvka's_algorithm)
   b. Cycles do not intersect! Simply eliminate each cycle.

Example 1:

After Phase 1

A1: H1 H3 $\boxed{\text{H2}}$

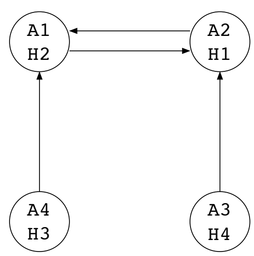
A2: H2 H4 $\boxed{\text{H1}}$

A3: H1 H2 $\boxed{\text{H4}}$

A4: H2 $\boxed{\text{H3}}$ H1

Phase 2 - no changes

Phase 3 graph



Final Result:

A1: $\boxed{\text{H1}}$

A2: $\boxed{\text{H2}}$

A3: $\boxed{\text{H4}}$

A4: $\boxed{\text{H3}}$

Example 2:

After Phase 1

A1: H3 H5 $\boxed{\text{H1}}$

A2: H1 H3 H4 $\boxed{\text{H2}}$

A3: H1 H4 H2 $\boxed{\text{H3}}$

A4: H1 H2 H3 $\boxed{\text{H4}}$

After Phase 2

A1: $\boxed{\text{H3}}$ H5 H1

A2: $\boxed{\text{H1}}$ H3 H4 H2

A3: H1 $\boxed{\text{H4}}$ H2 H3

A4: H1 $\boxed{\text{H2}}$ H3 H4

After Phase 3a

A1: $\boxed{\text{H3}}$ H1

A2: $\boxed{\text{H1}}$ H3 H4 H2

A3: H1 $\boxed{\text{H4}}$ H2 H3

A4: H1 $\boxed{\text{H2}}$ H3 H4

Phase 3b

The applicants will be deleted in the order: A1 A2 A3 A4

Final Result:

A1: $\boxed{\text{H3}}$

A2: $\boxed{\text{H1}}$

A3: $\boxed{\text{H4}}$

A4: $\boxed{\text{H2}}$

Example 3:

Find all maximum cardinality Pareto-optimal matchings for:

```
A1:  H1  H4  H2

A2:  H2  H3  H1

A3:  H1  H3  H4

A4:  H2  H4  H3
```

Maximum cardinality matchings (Phase 1) - Pareto-optimal ones are highlighted. The number below the non-Pareto-optimal ones is the number of Pareto-optimal ones that may be reached using later phases.

H1	H1	**H1**	H2	H2	H2	H4	**H4**	**H4**
H2	H2	**H3**	H1	H1	H3	H1	**H2**	**H3**
H3	H4	**H4**	H3	H4	H1	H3	**H1**	**H1**
H4	H3	**H2**	H4	H3	H4	H2	**H3**	**H2**
	1		1	1*	1	1		

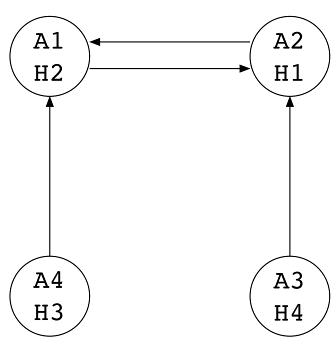
*Situation leading to 3.b iteration:

A1: H1 H4 $\boxed{H2}$

A2: H2 H3 $\boxed{H1}$

A3: H1 H3 $\boxed{H4}$

A4: H2 H4 $\boxed{H3}$



A1: $\boxed{H1}$

A2: $\boxed{H2}$

A3: H3 $\boxed{H4}$

A4: H4 $\boxed{H3}$

A1: $\boxed{H1}$

A2: $\boxed{H2}$

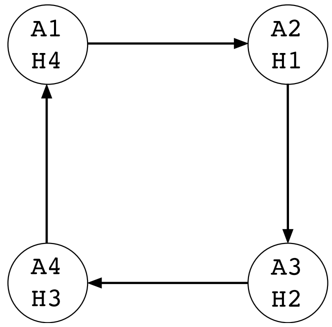
A3: $\boxed{H3}$

A4: $\boxed{H4}$

Example 4:

After Phases 1, 2, 3.a, 3.b.1, 3.b.2

A1:	H1	H2	H3	$\boxed{\texttt{H4}}$
A2:	H2	H3	H4	$\boxed{\texttt{H1}}$
A3:	H3	H4	H1	$\boxed{\texttt{H2}}$
A4:	H4	H1	H2	$\boxed{\texttt{H3}}$



3.B. Fair Division (KP 11)

Aside: `https://en.wikipedia.org/wiki/Ham_sandwich_theorem`

Cake Cutting (KP 11.1; `procaccia.2.pdf`; `cake.su.pdf`)

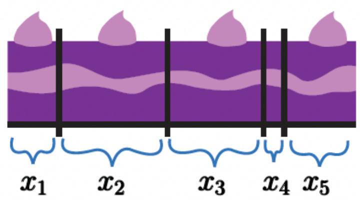


Figure 11.2. This figure shows a possible way to cut a cake into five pieces. The i^{th} piece is $B_i = [\sum_{k=1}^{i-1} x_k, \sum_{k=1}^{i} x_k)$. If the i^{th} piece goes to player j (i.e., $A_j := B_i$), then his value for this piece is $\mu_j(B_i)$.

Moving-knife Algorithm for fair division of a cake among n people

- Move a knife continuously over the cake from left to right until some player yells "Stop!"
- Give that player the piece of cake to the left of the knife.
- Iterate with the other $n-1$ players and the remaining cake.

Not envy-free

Simmons' Method Based on Sperner's Lemma (KP 11.1.1; `cake.su.pdf` p. 930-937)

Sperner's Lemma for Triangles. *Any Sperner-labelled triangulation of T must contain an odd number of elementary triangles possessing all labels. In particular, there is at least one.*

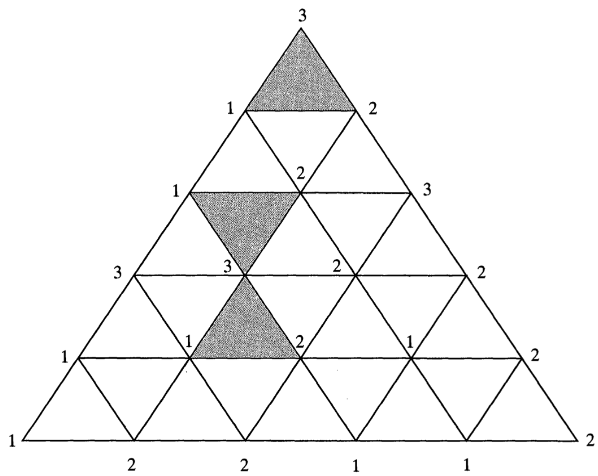


Figure 1. A Sperner labelling, with (1,2,3)-triangles marked.

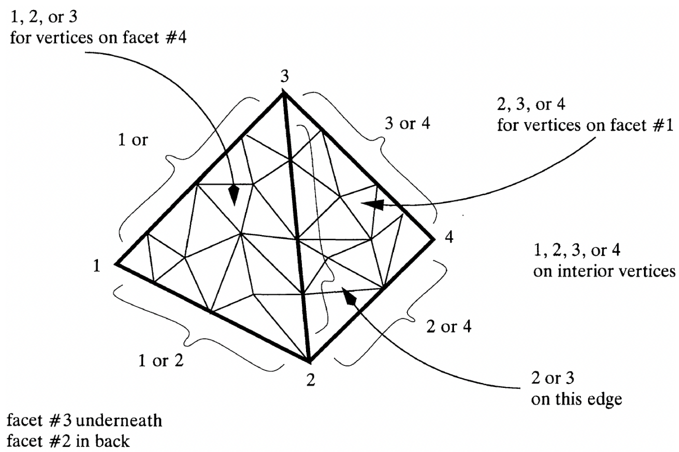


Figure 3. A triangulated tetrahedron, with Sperner labelling.

Sperner's Lemma. *Any Sperner-labelled triangulation of a n-simplex must contain an odd number of fully labelled elementary n-simplices. In particular, there is at least one.*

Trap-door walk (`https://en.wikipedia.org/wiki/Doubly_connected_edge_list` `https://ranger.uta.edu/~weems/dt.html`) as constructive proof and as fundamental procedure:

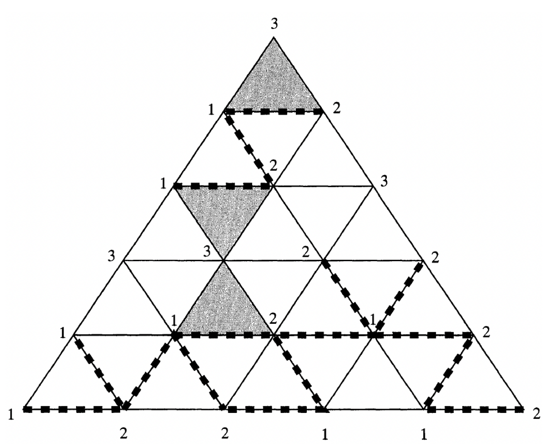


Figure 4. House, rooms, and doors indicated by dotted lines.

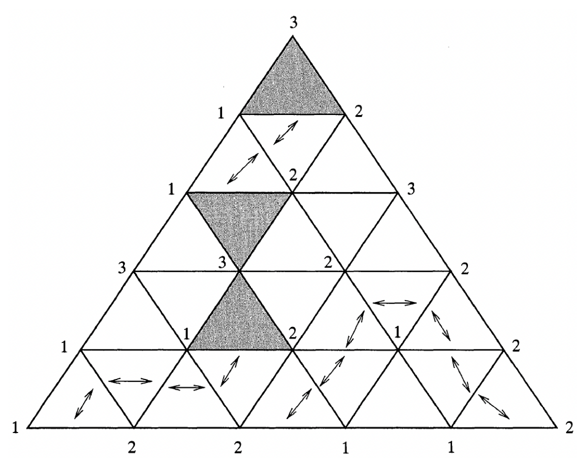


Figure 5. Walking through doors.

Simmons' Algorithm:

Construct initial triangulation ($\in$ and η issue for diameters with KP?)

Repeat until happy:

Properly color ("fully label") small triangles/tetrahedrons/simplices (vertex ownership, not Sperner)

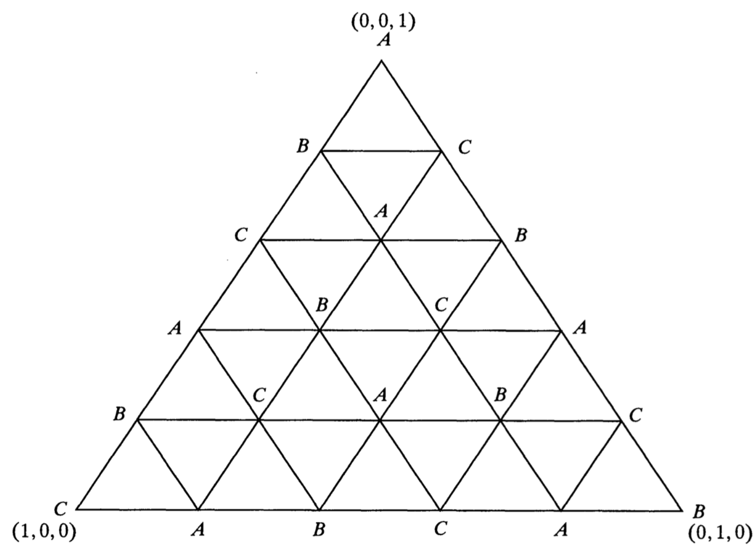


Figure 7. Labelling by ownership.

Determine the preferred slice for the owner of each vertex
(This is a Sperner labeling. Consider segment between (0, 0, 1) and (0, 1, 0).)

Find "fully labeled" simplices to use in happiness check

Refine triangulation using "barycentric subdivision"

Fair division of rent (indivisible objects) later

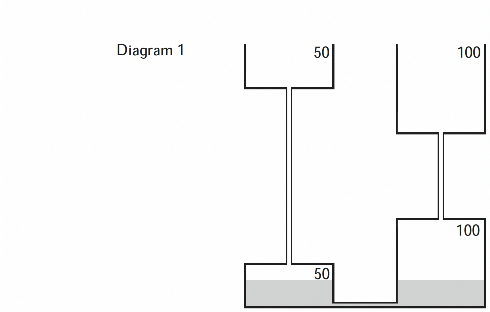
Bankruptcy (or divorce settlement or inheritance)

Chapter 4 of `https://ebookcentral.proquest.com/lib/utarl/detail.action?docID=377897` (`https://www.amazon.com/dp/0521696925`) is useful.

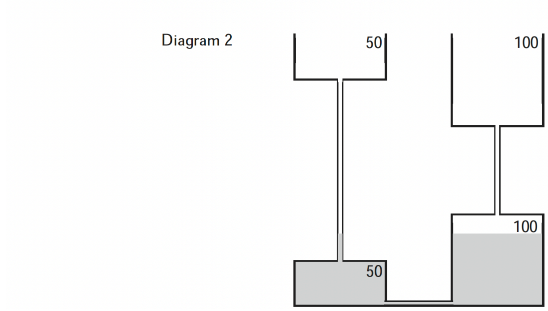
Four Laws are presented:

1. Contested-Garment Principle (p. 5). Also has "physical interpretation" diagrams for equivalent Rule of Linked Vessels.

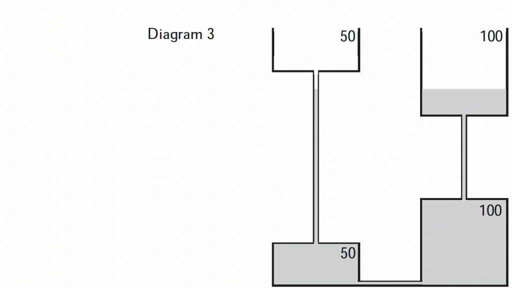
1. The estate is 80 and the debts are 100 and 200.



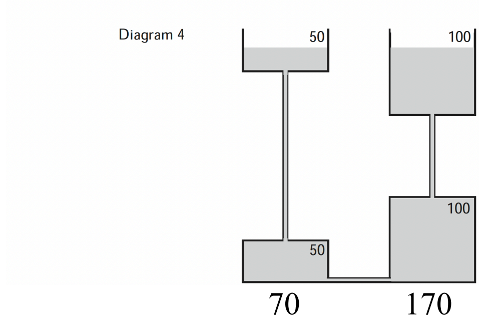
2. The estate is 140 and the debts are 100 and 200.



3. The estate is 180 and the debts are 100 and 200.



4. The estate is 240 and the debts are 100 and 200.



2. Rif's Law - order claimants in ascending order, then proceed evenly in levels

Claims \ Estate		300		500		600
100	α	100	100	100	100	100
200	α	100	100+β	200	200	200
300	α	100	100+β	200	200+γ	300

3. Proportional division - proportion of investment determines proportion of proceeds

4. O'Neill's law - race to the bank (Shapley value)

COOPERATIVE GAMES (KP 12; N 15.6)

Cooperative Game with Transferable Utility

Glove Market

Player 1 has a left glove. Player 2 has a right glove, so does Player 3.

$$v(123) = v(12) = v(13) = 1$$

Characteristic Function on subsets of *n* players: $v : 2^S \to \mathbb{R}$

- $v(\varnothing) = 0.$
- *Monotonicity:* If $S \subseteq T$, then $v(S) \leq v(T)$.

An allocation vector $\boldsymbol{\psi} = \boldsymbol{\psi}(v)$ is in the **core** if it satisfies the following two properties:

- **Efficiency:** $\sum_{i=1}^{n} \psi_i = v(\{1, \ldots, n\})$. This means, by monotonicity, that, between them, the players extract the maximum possible total value.
- **Stability:** Each coalition is allocated at least the payoff it can obtain on its own; i.e., for every set S,

$$\sum_{i \in S} \psi_i \geq v(S).$$

For the glove market, an allocation vector in the core must satisfy

$$\begin{aligned} \psi_1 + \psi_2 &\geq 1, \\ \psi_1 + \psi_3 &\geq 1, \\ \psi_1 + \psi_2 + \psi_3 &= 1. \end{aligned}$$

This system has only one solution: $\psi_1 = 1$ and $\psi_2 = \psi_3 = 0$.

Miners carrying gold bars in pairs. Core solution iff number of miners is even.

Example 12.2.2 (**Splitting a dollar:**). A parent offers his two children \$100 if they can agree on how to split it. If they can't agree, they will each get \$10. In this case $v(12) = 100$, whereas $v(1) = v(2) = 10$. The core conditions require that

$$\psi_1 \geq 10 \qquad \psi_2 \geq 10 \qquad \text{and} \qquad \psi_1 + \psi_2 = 100,$$

which clearly has multiple solutions.

Shapley Value

12.3.1. Shapley's axioms.

(1) **Symmetry**: If $v(S \cup \{i\}) = v(S \cup \{j\})$ for all S with $i, j \notin S$, then $\psi_i(v) = \psi_j(v)$.
(2) **Dummy**: A player that doesn't add value gets nothing; i.e., if $v(S \cup \{i\}) = v(S)$ for all S, then $\psi_i(v) = 0$.
(3) **Efficiency**: $\sum_{i=1}^{n} \psi_i(v) = v(\{1, \ldots, n\})$.
(4) **Additivity**: $\psi_i(v + u) = \psi_i(v) + \psi_i(u)$.

S-Veto Game:

Controlling coalition *S*. Characteristic function $w_S(\bullet) = 1$ when subset includes *S*.

$$\psi_i(w_S) = 0 \qquad \text{if } i \notin S$$

for $i, j \in S$, the symmetry axiom gives $\psi_i(w_S) = \psi_j(w_S)$

$$\psi_i(w_S) = \frac{1}{|S|} \qquad \text{if } i \in S.$$

Shapley Value Solution to Glove Market:

Player 1 gets $\psi_1(v) = 2/3$. $\psi_2(v) = \psi_3(v) = 1/6$.

LEMMA 12.3.2. *For any characteristic function $v : 2^{[n]} \to \mathbb{R}$, there is a unique choice of coefficients c_S such that*

$$v = \sum_{S \neq \varnothing} c_S w_S.$$

Four Stockholders and Shapley-Shubik Power Index (KP p. 208)

Shapley's Theorem

Define allocation by marginal contribution based on a permutation π:

$$\phi_i(v, \pi) = v\big(\pi\{1, \ldots, k\}\big) - v\big(\pi\{1, \ldots, k-1\}\big) \quad \text{where } \pi(k) = i.$$

satisfies the last three of Shapley's axioms. To satisfy symmetry, determine the expected value:

$$\psi_i(v) = \frac{1}{n!} \sum_{\pi \in S_n} \phi_i(v, \pi)$$

https://en.wikipedia.org/wiki/Airport_problem

(NASH) BARGAINING . . . skip

3.C. SOCIAL CHOICE (KP 13; N 9.2)

"Every nation gets the government it deserves" (https://en.wikiquote.org/wiki/Joseph_de_Maistre#Quotes)

"If . . . voters . . . knew the distribution . . ."

"Vote early and vote often"

https://en.wikipedia.org/wiki/Ross_Perot

https://en.wikipedia.org/wiki/Electronic_voting_in_Estonia

https://www.amazon.com/Gödels-Proof-Ernest-Nagel/dp/0814758371/

https://en.wikipedia.org/wiki/Gödel%2C_Escher%2C_Bach

https://en.wikipedia.org/wiki/Proofs_and_Refutations

https://en.wikipedia.org/wiki/Order_dimension

F. Brandl, et.al. "Proving the Incompatibility of Efficiency and Strategyproofness via SMT Solving". *J. ACM* 65, 2, Article 6 (January 2018), 28 pages. https://dl.acm.org/doi/10.1145/3125642

Condorcet Paradox - order of pairwise contests can make a difference

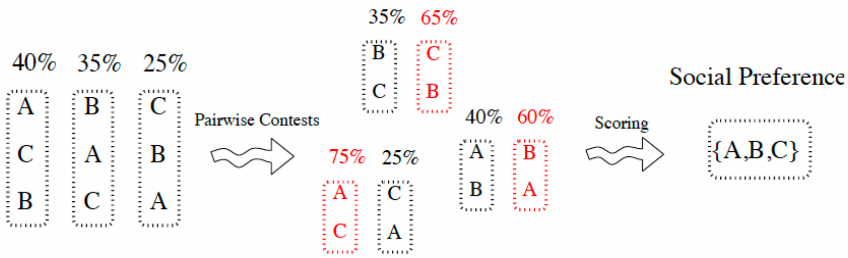


FIGURE 13.1. In pairwise contests A defeats C and C defeats B, yet B defeats A.

Mechanisms

Plurality voting (KP p. 217, figures 13.2 and 13.3)

Runoff elections (KP p. 217, figures 13.4 and 13.5)

Voting rule - produces a single winner from preference profile

Ranking rule - produces a total order from preference profile

Arrow's Properties

A social welfare function F satisfies *unanimity* if for every $\prec \in L$, $F(\prec, \ldots, \prec) = \prec$. That is, if all voters have identical preferences then the social preference is the same.

Voter i is a *dictator* in social welfare function F if for all $\prec_1 \ldots \prec_n \in L$, $F(\prec_1, \ldots, \prec_n) = \prec_i$. The social preference in a dictatorship is simply that of the dictator, ignoring all other voters. F is not a *dictatorship* if no i is a dictator in it.

A social welfare function satisfies *independence of irrelevant alternatives* if the social preference between any two alternatives a and b depends only on the voters' preferences between a and b. Formally, for every $a, b \in A$ and every $\prec_1, \ldots, \prec_n, \prec'_1, \ldots, \prec'_n \in L$, if we denote $\prec = F(\prec_1, \ldots, \prec_n)$ and $\prec' = F(\prec'_1, \ldots, \prec'_n)$ then $a \prec_i b \Leftrightarrow a \prec'_i b$ for all i implies that $a \prec b \Leftrightarrow a \prec' b$.

DEFINITION 13.2.4. A ranking rule R is **strategically vulnerable** at the profile $\boldsymbol{\pi} = (\succ_1, \ldots, \succ_n)$ if there is a voter i and alternatives A and B so that $A \succ_i B$ and $B \rhd A$ in $R(\boldsymbol{\pi})$, yet replacing $\succ_i$ by $\succ_i^*$ yields a profile $\boldsymbol{\pi}^*$ such that $A \rhd^* B$ in $R(\boldsymbol{\pi}^*)$.

Arrow's Impossibility Theorem

THEOREM 13.3.1. *Any ranking rule that satisfies unanimity and independence of irrelevant alternatives is a dictatorship.*

Gibbard-Satterthwaite Theorem

DEFINITION 13.4.1. A voting rule f from profiles to Γ is **strategy-proof** if for all profiles $\boldsymbol{\pi}$, candidates A and B, and voters i, the following holds: If $A \succ_i B$ and $f(\boldsymbol{\pi}) = B$, then all $\boldsymbol{\pi}'$ that differ from $\boldsymbol{\pi}$ only in voter i's ranking satisfy $f(\boldsymbol{\pi}') \neq A$.

THEOREM 13.4.2. *Let f be a strategy-proof voting rule onto Γ, where $|\Gamma| \geq 3$. Then f is a **dictatorship**. That is, there is a voter i such that for every profile $\boldsymbol{\pi}$ voter i's highest ranked candidate is equal to $f(\boldsymbol{\pi})$.*

Desirable Properties

(1) **Anonymity (i.e., symmetry):** The identities of the voters should not affect the results. I.e., if the preference orderings of voters are permuted, the society ranking should not change. This is satisfied by most reasonable voting systems, but not by the US electoral college or other regional based systems. Indeed, switching profiles between very few voters in California and Florida would have changed the results of the 2000 election between Bush and Gore.

(2) **Monotonicity:** If a voter moves candidate A higher in his ranking without changing the order of other candidates, this should not move A down in the society ranking.

(3) **Condorcet winner criterion:** If a candidate beats all other candidates in pairwise contests, then he should be the winner of the election. A related, and seemingly[2] weaker, property is the **Condorcet loser criterion**: The system should never select a candidate that loses to all others in pairwise contests.

(4) **IIA with preference strengths:** If two profiles have the same preference strengths for A versus B in all voter rankings, then they should yield the same preference order between A and B in the social ranking. (The *preference strength* of A versus B in a ranking is the number of places where A is ranked above B, which can be negative.)

(5) **Cancellation of ranking cycles:** If there is a subset of N candidates, and N voters whose rankings are the N cyclic shifts of one another (e.g. three voters each with a different ranking from Figure 13.1), then removing these N voters shouldn't change the outcome.

Analysis of Specific Voting Rules

Instant Runoff - remove candidate with fewest first-place votes after each round. Issues with monotonicity, IIA with preference strengths, and Condorcet winner criterion.

Borda Count - *m* candidates gives descending weights *m*, *m* - 1, . . . , 1 for ranks. Issues with Condorcet winner criterion, IIA, and strategic vulnerability.

Positional Voting - generalizes Borda count.

Approval Voting - vote for as many candidates as one wishes without ranking.

3.D. AUCTION CONCEPTS (R p. 11)

Welfare-Maximizing Auction - maximizes the utilities(s) of the winner(s) (assuming truthfulness)

(expected) utility = (expected) value - (expected) payment

Single-Parameter Domain/Environment/Setting (N p. 228?; R p. 24)

Each agent *i* has a private value-per-unit v_i (distribution is assumed public)

(Terminology: *k* units are homogeneous / indistinguishable, *k* items are heterogeneous / distinguishable. Concept of substitutes complicates this.)

(Desired quantity)

Feasible set of allocation vectors

Examples (and allocation vectors) (R p. 24)

Single Item (basis vectors)

k Units (0-1 vectors)

Sponsored Search (click-through rates) (R 2.6)

Public Project (nobody-or-everybody)

Single Item Auctions and Independent Private Values (review of Notes 1.D; KP 14.1-14.2, Def. 14.1.1)

First-Price

Ascending - Dominant Strategy: stop bidding when price exceeds value

Sealed-Bid - Dominant Strategy: bid $\beta(v) = \frac{n-1}{n} v$ (KP p. 241)

Sealed-Bid Second-Price (R Theorem 2.4) - Dominant Strategy: bid value

Dominant-Strategy Incentive Compatible (DSIC; R p. 15)

Truthful bidding is always a dominant strategy

Truthful bidders obtain non-negative utility

Bayes-Nash Equilibrium - For every bidder *i*, utility is maximized by bidding $\beta_i(V_i)$

Revenue in Single-Item Auctions (KP 14.3)

EXAMPLE 14.3.1. We return to our earlier example of two bidders, each with a value drawn independently from a U[0,1] distribution. From that analysis, we know that if the auctioneer runs a first-price auction, then in equilibrium his expected revenue will be

$$\mathbb{E}\left[\max\left(\frac{V_1}{2}, \frac{V_2}{2}\right)\right] = \frac{1}{3}.$$

On the other hand, suppose that in the exact same setting, the auctioneer runs a second-price auction. Since we can assume that the bidders will bid truthfully, the auctioneer's revenue will be the expected value of the second-highest bid, which is

$$\mathbb{E}\left[\min(V_1, V_2)\right] = \frac{1}{3},$$

Revenue Equivalence (KP 14.4)

Argubly, the central result in single-object auction theory is the *revenue equivalence theorem* . . . (Bichler, p. 61)

Perhaps the most remarkable theoretical result in auction theory is *revenue equivalence*, a principle which can be loosely expressed as follows: for a broad class of auctions, bidders, ideally, adjust their behavior to the rules in such a way that the expected revenue to the seller remains the same. (Steiglitz, p. 180)

THEOREM 14.4.2 (**Revenue Equivalence**). *Suppose that each agent's value V_i is drawn independently from the same strictly increasing distribution $F \in [0, h]$. Consider any n-bidder single-item auction in which the item is allocated to the highest bidder, and $u_i(0) = 0$ for all i. Assume that the bidders employ a symmetric strategy profile $\beta_i := \beta$ for all i, where β is strictly increasing in $[0, h]$.*

(i) *If $(\beta, \ldots, \beta)$ is a Bayes-Nash equilibrium, then for a bidder with value v,*

$$a(v) = F(v)^{n-1} \quad \text{and} \quad p(v) = va(v) - \int_0^v a(w)dw. \tag{14.9}$$

(ii) *If (14.9) holds for the strategy profile $(\beta, \ldots, \beta)$, then for any bidder i with utility $u(\cdot|\cdot)$ and all $v, w \in [0, h]$,*

$$u(v|v) \geq u(w|v). \tag{14.10}$$

Theorem 9.46 (The Revenue Equivalence Principle) *Under certain weak assumptions (to be detailed in the proof body), for every two Bayesian–Nash implementations of the same social choice function f, we have that if for some type t_i^0 of player i, the expected (over the types of the other players) payment of player i is the same in the two mechanisms, then it is the same for every value of t_i. In particular, if for each player i there exists a type t_i^0 where the two mechanisms have the same expected payment for player i, then the two mechanisms have the same expected payments from each player and their expected revenues are the same.*

Ascending

Descending (Dutch)

Second-Price Sealed-Bid

First-Price Sealed-Bid

All-pay

War of attrition (second-price all-pay)

When i.i.d. valuations for n bidders are uniformly distributed over [0, 1], the expected revenue is $(n - 1)/(n + 1)$

https://en.wikipedia.org/wiki/Winner's_curse

Reserve Prices

Vickrey Auction with a Reserve Price (KP p. 243; R exercise 5.1)

> Perhaps surprisingly, an auctioneer may want to impose a reserve price even if his own value for the item is zero. For example, we have seen that for two bidders with values independent and drawn from $U[0,1]$, all auctions that allocate to the highest bidder have an expected auctioneer revenue of $1/3$.
>
> Now consider the expected revenue if, instead, the auctioneer uses the Vickrey auction with a reserve of r. Relative to the case of no reserve price, the auctioneer loses an expected revenue of $r/3$ if both bidders have values below r, for a total expected loss of $r^3/3$. On the other hand, he gains if one bidder is above r and one below. This occurs with probability $2r(1-r)$, and the gain is r minus the expected value of the bidder below r; i.e., $r - r/2$. Altogether, the expected revenue is
>
> $$\frac{1}{3} - \frac{r^3}{3} + 2r(1-r)\frac{r}{2} = \frac{1}{3} + r^2 - \frac{4}{3}r^3.$$
>
> Differentiating shows that this is maximized at $r = 1/2$, yielding an expected auctioneer revenue of $5/12$. (This is not a violation of the Revenue Equivalence Theorem because imposition of a reserve price changes the allocation rule.)

Bidder-Specific Reserve (R p. 78)

Revenue Equivalence (KP 14.5.1)

Entry Fee vs Reserve Price (KP 14.5.2)

Evaluation Fee (KP 14.5.3) - must pay fee to know your value (similarly eBay shipping)

Knapsack Auctions (R 4.1, 4.2; Chapters 1/2/3 of https://www-degruyter-com.ezproxy.uta.edu/document/doi/10.7312/milg17598/html)

Seller has capacity w

Each bidder has public size w_i ($\leq w$) and private valuation v_i (which is also the bid b_i)

Since knapsack is NP-hard, apply greedy heuristic of ordering bid-to-size ratios descending

Achieves at least 50% of the maximum social welfare . . .

From CSE 3318 Notes 6:

Observe that applying fractional concept to 0/1 problem gives an upper bound on what may be achieved optimally (OPT) for 0/1.

$$\text{Suppose } \sum_{k=1}^{i} w_k > W \text{ and } \sum_{k=1}^{i-1} w_k \le W$$

$$\text{OPT} \le \sum_{k=1}^{i-1} v_k + \frac{W - \sum_{k=1}^{i-1} w_k}{w_i} v_i$$

By taking the larger of the revenue for the first *i* - 1 items or the revenue for item *i*, we will achieve at least 1/2 of OPT. (Why?)

Aside: eBay

`https://en.wikipedia.org/wiki/EBay#Bidding`

Individually Rational (KP 14.5.4; N 9.4.1)

Each bidder's expected utility is nonnegative (at that decision time)

Ex-ante - bidders know only the value distributions

Ex-interim - bidders know the value distributions and their own value (entry-fee, evaluation fee)

Ex-post - bidders know after the auction (first-price and second-price)

3.E. Auction Concepts (continued)

Myerson's Lemma (R 3)

Definition 3.5 (Implementable Allocation Rule) An allocation rule ... is *implementable* if there is a payment rule such that the direction-revelation mechanism is DSIC.

Definition 3.6 (Monotone Allocation Rule) An allocation rule ... is *monotone* if for every agent the allocation is nondecreasing in her bid.

Theorem 3.7 (Myerson's Lemma) . . .

(a) Allocation rule is implementable if and only if it is monotone.

(b) Such a monotone allocation rule has a *unique* payment rule for which the direct-revelation mechanism is DSIC and the payment is zero whenever the bid is zero.

(c) The payment rule in (b) has an *explicit* formula.

Highlights of the proof: expressions (3.4) and (3.5/3.6), along with diagrams on (R p. 32).

Example: Knapsack Auction and critical bid (R p. 41)

Characterization of Bayes-Nash Equilibrium (KP 14.6)

Myerson's Lemma, but for distributions

KP proof and diagram (KP p. 247-248) mirror R proof and diagram (R p. 28-32)

POA in Auctions (KP 14.7)

See Roughgarden et.al, "The Price of Anarchy in Auctions"

Revelation Principle

Theorem 4.3 (Revelation Principle for DSIC Mechanisms): For every mechanism in which every participant always has a dominant strategy, there is an equivalent direct-revelation DSIC mechanism. (R p. 47)

THEOREM 14.8.2 (**The Revelation Principle**). *Let $\mathcal{A}$ be a direct auction where $\{\beta_i\}_{i=1}^n$ is a Bayes-Nash equilibrium. Recall Definition 14.1.1. Then there is another direct auction $\tilde{\mathcal{A}}$, which is BIC and has the same winners and payments as $\mathcal{A}$ in equilibrium; i.e., for all $\mathbf{v} = (v_1, \dots, v_n)$, if $b_i = \beta_i(v_i)$ and $\mathbf{b} = (b_1, \dots, b_n)$, then*

$$\boldsymbol{\alpha}^{\mathcal{A}}[\mathbf{b}] = \boldsymbol{\alpha}^{\tilde{\mathcal{A}}}[\mathbf{v}] \quad \text{and} \quad \mathscr{P}^{\mathcal{A}}[\mathbf{b}] = \mathscr{P}^{\tilde{\mathcal{A}}}[\mathbf{v}].$$

EXAMPLE 14.8.3. Recall Example 14.2.1, a first-price auction with two bidders with U[0, 1] values. An application of the Revelation Principle to this auction yields the following BIC auction: Allocate to the highest bidder and charge him half of his bid.

Myerson's Optimal Auction/Revenue Maximization (R 5; KP 14.9)

"Take it or leave it" by setting a reserve price *r*

One Bidder, One Item (Monopoly Price)

Expected Revenue: $r \cdot (1 - F(r))$ (F is the distribution function, the probability that the value is at most the argument)

Uniform distribution [0, 1] for buyer's value - set r at 1/2

Virtual Valuation ("regular" if non-decreasing)

$$\varphi_i(v_i) = \underbrace{v_i}_{\text{what you'd like to charge } i} - \underbrace{\frac{1 - F_i(v_i)}{f_i(v_i)}}_{\text{“information rent” earned by bidder } i}$$

Expected Revenue = Expected Virtual Welfare

$$\mathbf{E}_{\mathbf{v}}\left[\sum_{i=1}^{n} \varphi_i(v_i) \cdot x_i(\mathbf{v})\right]$$

DEFINITION 14.9.12. The Myerson auction for distributions with strictly increasing virtual value functions is defined by the following steps:

(i) Solicit a bid vector $\mathbf{b}$ from the agents.
(ii) Allocate the item to the bidder with the largest virtual value $\psi_i(b_i)$ if positive, and otherwise, do not allocate. That is[23],

$$\alpha_i(b_i, \mathbf{b}_{-i}) = \begin{cases} 1, & \text{if } \psi(b_i) > \max_{j \neq i} \psi(b_j) \text{ and } \psi(b_i) \geq 0; \\ 0, & \text{otherwise,} \end{cases} \tag{14.29}$$

(iii) If the item is allocated to bidder i, then she is charged her *threshold bid* $t_*(\mathbf{b}_{-i})$, the minimum value she could bid and still win, i.e.,

$$t_*(\mathbf{b}_{-i}) := \min\{b \ : \ \psi_i(b) \geq \max(0, \{\psi_j(b_j)\}_{j \neq i})\}. \tag{14.30}$$

THEOREM 14.9.13. *Suppose that the bidders' values are independent with strictly increasing virtual value functions. Then the Myerson auction is optimal; i.e., it maximizes the expected auctioneer revenue in Bayes-Nash equilibrium. Moreover, bidding truthfully is a dominant strategy.*

COROLLARY 14.9.14. *The Myerson optimal auction for i.i.d. bidders with strictly increasing virtual value functions is the Vickrey auction with a reserve price of* $\psi^{-1}(0)$.

Approximately (Near-) Optimal Auctions

What if you don't have the agents' distributions?

Prophet Inequality (R 6.2) and Simple Single-Item Auction (R 6.3)

`https://cacm.acm.org/magazines/2017/12/223050-lousy-advice-to-the-lovelorn`

Bulow-Klemperer (KP 14.10.1; R 6.4)

THEOREM 14.10.1. *Let F be a distribution for which virtual valuations are increasing. The expected revenue in the optimal auction with n i.i.d. bidders with values drawn from F is upper bounded by the expected revenue in a Vickrey auction with $n+1$ i.i.d. bidders with values drawn from F.*

Lookahead Auction (KP 14.10.2)

(i) Solicit bids from the agents. Suppose that agent i submits the highest bid b_i. (If there are ties, pick one of the highest bidders arbitrarily.)

(ii) Compute the conditional distribution $\tilde{F}_i$ of V_i given the bids $\mathbf{b}_{-i}$ and the event $V_i \geq \max_{j \neq i} b_j$. Let $p_i = p_i(\mathbf{b}_{-i})$ be the price p that maximizes $p(1 - \tilde{F}_i(p))$.

(iii) Run the optimal single-bidder auction with agent i, using his previous bid b_i and the distribution $\tilde{F}_i$ for his value: This auction sells the item to agent i at price p_i if and only if $b_i \geq p_i$.

Lookahead Auction is Approximately Optimal (KP 14.10.3)

PROPOSITION 14.10.3. *The Lookahead auction is optimal among truthful auctions that allocate to the highest bidder (if at all).*

THEOREM 14.10.4. *The Lookahead (LA) auction yields an expected auctioneer revenue that is at least half that of the optimal truthful and ex-post individually rational auction even when bidders have dependent values.*

3.F. MULTI-UNIT AUCTIONS

Uniform-Price for Budgeted Bidders (R 9.2)

m = number of items

Besides private valuation v_i, each buyer has a public budget B_i

Demand of bidder i at price p:
$$D_i(p) = \begin{cases} \min\left\{\lfloor \frac{B_i}{p} \rfloor, m\right\} & \text{if } p < v_i \\ 0 & \text{if } p > v_i \end{cases}$$

Utility is still $v_i - p$

Not DSIC (due to “ambiguity” for $p = v_i$, see R p. 115 Example 9.1) - demand reduction for reporting a lower v_i

Clinching for Budgeted Bidders (R 9.3; `dobzinski.pdf`)

Price gradually rises to point where demand for all but one buyer (with largest residual demand) is below remaining supply.

If total demand exceeds supply, allocate one item to largest-residual-demand buyer. Otherwise, allocate remaining supply at current price.

DSIC

Clinching with Descending Valuations (`ausubel.pdf`)

Ascending auction with private valuations, but payments are based on prices where demand falls below remaining supply

Agent i valuations are downward-sloping $v_{i1} \geq v_{i2} \geq \ldots \geq v_{ik}$ for auction with k units

Nisan survey (Algorithmic Mechanism Design: Through the lens of Multi-unit auctions)

- We have m identical indivisible units of a single good to be allocated among n strategic players also called *bidders*.
- Each bidder i has a privately known *valuation function* $v_i : \{0, ..., m\} \rightarrow \Re^+$, where $v_i(k)$ is the value that this bidder has for receiving k units of the good. We assume that $v_i(0) = 0$ and free disposal: $v_i(k) \leq v_i(k+1)$ (for all $0 \leq k < m$).
- Our goal (as the auctioneer) is to allocate the units among the bidders in a way that optimizes *social welfare*: each bidder i gets m_i units and we aim to maximize $\sum_i v_i(m_i)$, where the feasibility constraint is that $\sum_i m_i \leq m$.

Representations

Bidding Languages

1. **Single Minded Bids:** This language allows only representing "step functions", valuations of the form $v(k) = 0$ for $k < k^*$ and $v(k) = w^*$ for $k \geq k^*$. Clearly to represent such a valuation we only need to specify two numbers: k^* and w^*. Clearly, also, this is a very limited class of valuations.
2. **Step Functions:** This language allows specifying an arbitrary sequence of pairs $(k_1, w_1), (k_2, w_2), \ldots, (k_t, w_t)$ with $0 < k_1 < k_2 \cdots < k_t$ and $0 < w_1 < w_2 < \cdots < w_t$. In this formalism $v(k) = w_j$ for the maximum j such that $k \geq k_j$. Thus, for example, the bid $((2,7),(5,23))$ would give a value of \$0 to 0 or 1 items, a value of \$7 to 2, 3, or 4 items, and a value of \$23 to 5 or more items. Every valuation can be represented this way, but most valuations will take length m. Simple ones – ones that have only a few "steps" – will be succinctly represented.
3. **Piece-Wise Linear:** This language allows specifying a sequence of marginal values rather than of values. Specifically, a sequence $(k_1, p_1), (k_2, p_2), \ldots, (k_t, p_t)$ with $0 < k_1 < k_2 \cdots < k_t$ and $p_j \geq 0$ for all j. In this formalism p_j is the marginal value of item k for $k_j \leq k < k_{j+1}$. Thus $v(k) = \sum_{l=1}^{k} u_l$ with $u_l = p_j$ for the largest j such that $l \geq k_j$. In this representation, the bid $((2,7),(5,23))$ would give a value of \$7 to 1 item, \$14 to 2 items, \$37 to 3 items, \$60 to 4 items, and \$83 to 5 or more items.

Value Queries

Communication Queries

Intractable General Allocation Algorithm (DP) - $O(nm^2)$ time

1. Fill the $(n+1)*(m+1)$ table s, where $s(i,k)$ is the maximum value achievable by allocating k items among the first i bidders:

   (a) For all $0 \le k \le m$: $s(0,k) = 0$

   (b) For all $0 \le i \le n$: $s(i,0) = 0$

   (c) For $0 < i \le n$ and $0 < k \le m$: $s(i,k) = max_{0 \le j \le k}[v_i(j) + s(i-1, k-j)]$

2. The total value of the optimal allocation is now stored in $s(n,m)$

3. To calculate the m_i's themselves, start with $k = m$ and for $i = n$ down to 1 do:

   (a) Let m_i be the value of j that achieved $s(i,k) = v_i(j) + s(i-1, k-j)$

   (b) $k = k - m_i$

Tractable Downward Sloping Valuations via Market Equilibrium and Multiple Binary Searches

$v_i(k+1) - v_i(k) \le v_i(k) - v_i(k-1)$ for all bidders i and number of items $1 \le k \le m-1$

Algorithmically, given a potential price p, we can calculate players' demands using binary search to find the point where the marginal value decreases below p. This allows us to calculate the total demand for a price p, determining whether it is too low or too high, and thus search for the right price p using binary search. For clarity of exposition we will assume below that all values of items are distinct, $v_i(k) \ne v_{i'}(k')$ whenever $(i,k) \ne (i',k')$.

Allocation Algorithm for Downward Sloping Valuations

1. Using binary search, find a clearing price p in the range $[0, V]$, where $V = max_i[v_i(1)]$:

   (a) For each $1 \le i \le n$, use binary search over the range $\{0, 1, ..., m\}$, to find m_i such that $v_i(m_i) - v_i(m_{i-1}) \ge p > v_i(m_{i+1}) - v_i(m_i)$

   (b) If $\sum_i m_i > m$ then p is too low; if $\sum_i m_i < m$ then p is too high, otherwise we have found the right p.

2. The optimal allocation is given by $(m_1, ..., m_n)$.

Intractability for value queries model

Theorem 3 *Every algorithm for the multi-unit allocation problem, in the value-queries model, needs to make at least* $2m - 2$ *queries for some input.*

Let us consider the simplest possible bidding language, that of Single Minded Bids. In this language each valuation v_i is represented as (k_i, p_i) with the meaning that for $k < k^*$ we have $v_i(k) = 0$ and for $k \geq k_i$ we have $v_i(k) = p_i$. The optimal allocation will thus need to choose some subset of the players $S \subseteq \{1 \dots n\}$ which it would satisfy. This would give total value $\sum_{i \in S} p_i$ with the constraint that $\sum_{i \in S} k_i \leq m$. This optimization problem is a very well known problem called the knapsack problem, and it is one of a family of problems known to be "NP-complete". In particular, no computationally-efficient algorithm exists for solving the knapsack problem unless "P=NP", which is generally conjectured not to be the case (the standard reference is still Garey and Johnson [1979]). It follows that no computationally-efficient algorithm exists for our problem when the input valuations are presented in any of the bidding languages mentioned above all of which contain the single-minded one as special cases. We thus have:

Theorem 4 *Assuming $P \neq NP$, there is no polynomial-time algorithm for the multi-unit allocation problem, in any of the bidding language models.*

Communication Complexity Results

Theorem 5 *(Nisan and Segal [2006]) Every algorithm for the multi-unit allocation problem, in any query model, needs to ask queries whose total answer length is at least $m-1$ in the worst case.*

3.G. Multi-Unit Auctions (continued)

DP-based Approximation Scheme (with binary searches)

At this point we seem to have very strong and general impossibility results for algorithms attempting to find the optimal allocation. However, it turns out that if we just slightly relax the optimality requirement and settle for "approximate optimality" then we are in much better shape. Specifically, an *approximation scheme* for an optimization problem is an algorithm that receives, in addition to the input of the optimization problem, a parameter ϵ which specifies how close to optimal do we want to solution to be. For a maximization optimization problem (like our allocation problem) this would mean a solution whose value is at least $1 - \epsilon$ times the value of the optimal solution. It turns out that this is possible to do, computationally-efficiently, for our problem using dynamic programming.

The basic idea is that there exists an optimal algorithm whose running time is polynomial in the *range of values*. (This is a different algorithm than the one presented in section 4.1 whose running time was polynomial in the number of items m.) Our approximation algorithm will thus truncate all values $v_i(k)$ so that they become small integer multiples of some δ and run this optimal algorithm on the truncated values. Choosing $\delta = \epsilon V/n$ where $V = max_i[v_i(m)]$ strikes the following balance: on one hand, all values are truncated to an integer multiple $w\delta$ for an integer $0 \le w \le n/\epsilon$, and so the range of the total value is at most the sum of n such terms, i.e. at most n^2/ϵ. On the other hand, the total additive error that we make is at most $n\delta \le \epsilon V$. Since V is certainly a lower bound on the total value of the optimal allocation this implies that the fraction of total value that we loose is at most $(\epsilon V)/V = \epsilon$ as required.

The allocation algorithm among the truncated valuations uses dynamic programming, and fills a $(n+1)*(W+1)$-size table K, where $W = n^2/\epsilon$. The entry $K(i,w)$ holds the minimum number of items that, when allocated optimally between the first i players, yields total value of at least $w\delta$. Each entry in the table can be computed efficiently from the previous ones, and once the table is filled we can recover the actual allocation.

1. Fix $\delta = \epsilon V/n$ where $V = max_i[v_i(m)]$, and fix $W = n^2/\epsilon$.

2. Fill an $(n+1)*(W+1)$ table K, where $K(i,w)$ holds the minimum number of items that, when allocated optimally between the first i players, yields total value of at least $w\delta$:

   (a) For all $0 \le i \le n$: $K(i,0) = 0$

   (b) For all $w = 1, \ldots, W$: $K(0,w) = \infty$

   (c) For all $i = 1, \ldots, n$ and $w = 1, \ldots, W$ compute $K(i,w) = min_{0 \le j \le w} val(j)$, for $val(j)$ defined by:

      i. Use binary search to find minimum number of items k such that $v_i(k) \ge j\delta$

      ii. $val(j) = k + K(i-1, w-j)$

3. Let w be the maximum index such that $K(n,w) \le m$

4. For i going from n down to 1:

   (a) Let j be so that $K(i,w) = val(j)$ (as computed above)

   (b) Let m_i be the minimum value of k such that $v_i(k) \ge j\delta$

   (c) Let w = w - j

Our algorithm's running time is dominated by the time required to fill the table K which is of size $n \times n^2/\epsilon$ where filling each entry in the table requires going over all possible n/ϵ values j and for each performing a binary search that takes $O(\log m)$ queries. Thus the total running time is polynomial in the input size as well as in the precision parameter ϵ which often called a fully polynomial time approximation scheme. For most realistic optimization purposes such an arbitrarily good approximation is essentially as good as an optimal solution. We have thus obtained:

Theorem 6 *There exists an algorithm for the approximate multi-unit allocation problem (in any of the models we considered) whose running time is polynomial in n, $\log m$, and ϵ^{-1}. It produces an allocation $(m_1, ..., m_n)$ that satisfies $\sum_i v_i(m_i) \geq (1-\epsilon) \sum_i v_i(opt_i)$, where $(opt_1, ..., opt_n)$ is the optimal allocation.*

Payments, Incentives, and Mechanisms

Direct Revelation Mechanism

VCG Mechanism

Approximation and Incentives

Maximum in Range Mechanisms

Single Parameter Mechanisms

Multi-Parameter Mechanisms Beyond VCG?

Randomization

3.H. Truthful Auctions in Win/Lose Settings

Win/Lose Allocation Settings

Social Surplus and the VCG Mechanism

Applications of VCG

3.I. Truthful Auctions in Win/Lose Settings (continued)

Sponsored Search Auctions

3.J. VCG and Scoring Rules

Examples

Social Surplus Maximization and the General VCG mechanism

Scoring Rules

3.K. Combinatorial Auctions

Introduction

Single-Minded Case

Walrasian Equilibrium and the LP Relaxation

Bidding Languages

Iterative Auctions: The Query Model, Value vs. Demand

Communication Complexity

Ascending Auctions

3.L. Matching Markets

Maximum Weighted Matching (Assignment Problem)

Envy-Free Prices

Envy-Free Division of Rent

Maximum Matching by Ascending Auctions

Matching Buyers and Sellers (Assignment Games)

Application to Weighted Hide-and-Seek Games