Approximation Mechanisms: computation, representation, and incentives

Noam Nisan

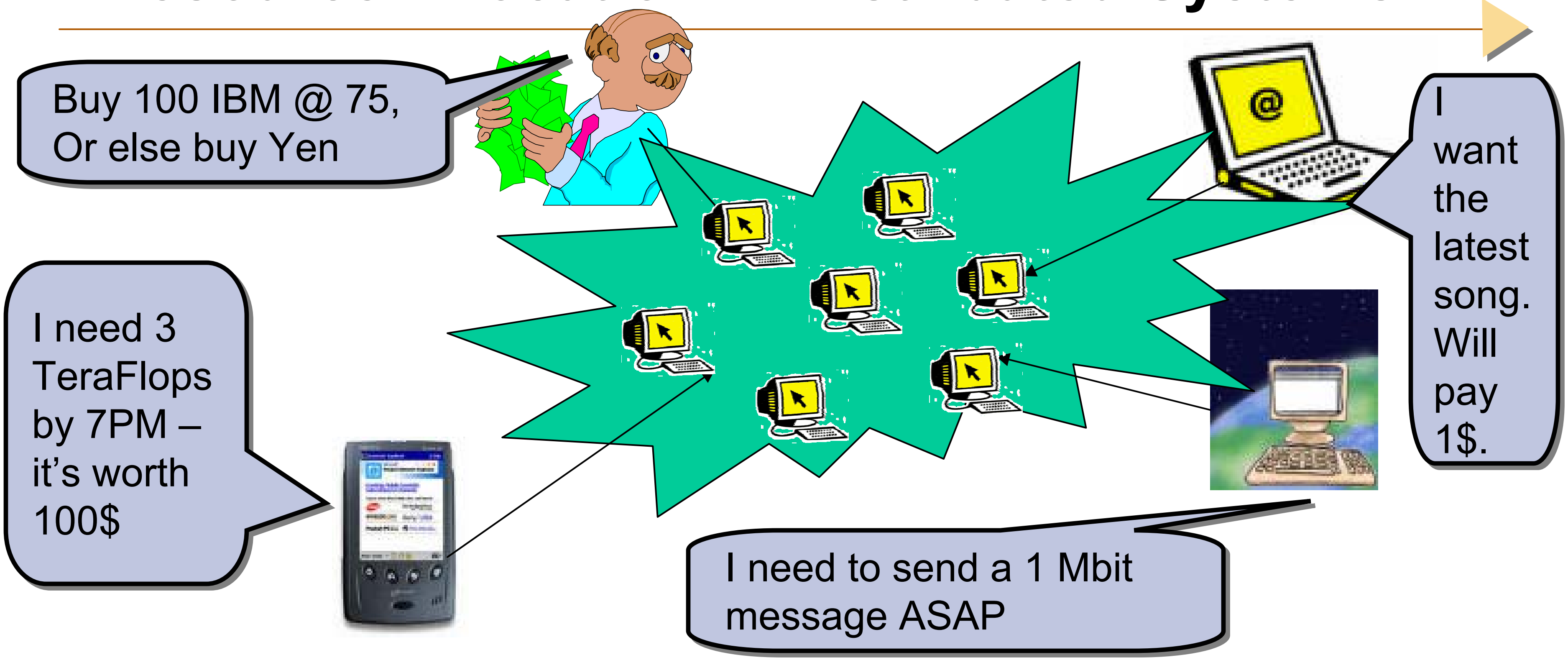
Hebrew University, Jerusalem

Based on joint works with Amir Ronen, Ilya Segal, Ron Lavi, Ahuva Mu'alem, and Shahar Dobzinski

Talk Structure

- Algorithmic Mechanism Design
- Example: Multi-unit Auctions
- Representation and Computation
- VCG mechanisms
- General Incentive-Compatible Mechanisms

Resource Allocation in Distributed Systems



- Each participant in today's distributed computation network has its own *selfish* set of goals and preferences.
- We, as designers, wish to optimize some common aggregated goal.
- Assumption: participant's will act in a rationally selfish way.

Mechanisms for Maximizing Social Welfare

- Set A of possible social alternatives (allocations of all resources) affecting n players.
- Each player has a valuation function $v_i : A \rightarrow \Re$ that specifies his *value* for each possible alternative.
- Our goal: maximize social welfare $\Sigma_i\, v_i(a)$ over all $a \varepsilon A$.
- Mechanism: Allocation Rule $a=f(v_1 \ldots v_n)$ and player payments $p_i(v_1 \ldots v_n) \in \Re$.
  - Incentive Compatibility: a rational player will always report his true valuation to the mechanism.

Dominant-strategy Incentive-compatibility

For <u>every</u> profile of valuations, you do not gain by lying:

$$\forall i \ \forall v_1 \ldots v_n \ \forall v'_i: \ v_i(a)-p \geq v_i(a')-p'$$

Where: $a=f(v_i\, v_{-i})$, $p=p_i(v_i\, v_{-i})$, $a'=f(v'_i\, v_{-i})$, $p'=p_i(v'_i\, v_{-i})$.

We will not consider weaker notions:

- Randomized
- Bayesian
- Approximate
- Computationally-limited
- …

There is no loss of generality relative to any mechanism with ex-post-Nash equilibria.

The classic solution -- VCG

1. Find the welfare-maximizing alternative *a*
2. Make every player pay "VCG prices":
   - Pay $\Sigma_{k \neq i} v_k(a)$ to each player *i*
   - Actually, a 2nd, non-strategic, term makes player payments ≥ 0.
   - But we don't worry about revenue or profits in this talk.

Proof: Each player's utility is identified with the social welfare.

Problem: (1) is often computationally hard.

CS approach: approximate or use heuristics.

Problem: VCG idea doesn't extend to approximations.

Running Example: Multi-unit Auctions

- There are *m* identical units of some good to allocate among *n* players.

- $v_i(q)$ – value to player *i* if he gets exactly *q* units

- Valid allocation: $(q_1 \ldots q_n)$ such that $\Sigma_i q_i \leq m$

- *Social welfare:* $\Sigma_i v_i(q_i)$

Representing the valuation

- Single-minded: *(p,q)* – value is *p* for at least *q* units.
- "k-minded" / "XOR-bid": a sequence of k increasing pairs (p_j,q_j) – value is p_j, for $q_j \le q < q_{j+1}$ units.
    - Example: "(5$ for 3 items), (7$ for 17 items)"
- General, "black box": can answer queries $v_i(q)$.
    - Example: $v(q) = 3q^2$

What can be done efficiently?

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints			
Incentive compatible VCG payments			
General incentive compatible			

What can be done efficiently?

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints		**Computational Benchmark**	
Incentive compatible VCG payments		**Existing Ideas**	
General incentive compatible		**Our Goal**	

What can be done efficiently?

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints			
Incentive compatible VCG payments			
General incentive compatible			

← **Strategic** → **complexity gap**

← **Representation** → **Complexity gap**

Approximation quality levels

- How well can a computationally-efficient (polynomial time) mechanism approximate the optimal solution?
  - A: Exact Optimization
  - B: Fully Polynomial Time Approximation Scheme (FPTAS)-- to within any $\varepsilon>0$, with running time polynomial in $1/\varepsilon$.
  - C: Polynomial Time Approximation Scheme (PTAS)-- to within any fixed $\varepsilon>0$.
  - D: To within some fixed constant $c>0$ (this talk $c=2$).
  - E: Not to within any fixed constant.

- What we measure is the worst-case ratio between the quality (social welfare) of the optimal solution and the solution that we get.

Rest of the talk…

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints	B	B	B
Incentive compatible VCG payments	C	C	D
General incentive compatible	B	Conjecture: C	Conjecture + Partial result: D

Computational Status

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints	**Not A** **NP-compete**	**Not A**	

The SM case is exactly Knapsack:

Input: $(p_1,q_1) \dots (p_n,q_n)$

Maximize $\Sigma_{i \in S}\, p_i$ where $\Sigma_{i \in S}\, q_i \leq m$

$v_i(q) = p_i$ iff $q \geq q_i$ $(0$ otherwise$)$

Computational Status: general valuations

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints			**Not A** **Exponential**

Proof:

- Consider two players with $v_1(q)=v_2(q)=q$ except for a single value of q^* where $v_1(q^*)=q+1$.
- $v_1(q_1)+v_2(q_2)=m$ except for $q_1=q^*; q_2=m-q^*$.
- Finding q^* requires exponentially many (i.e. m) queries.

THM (N+Segal): Lower bound holds for all types of queries.

Proof: Reduction to Communication Complexity

Computational Status: Approximation

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints	B	B	B **FPTAS**

Knapsack has an FPTAS – works in general:

1. Round ***prices*** $v_i(q)$ down to integer multiple of δ
2. For all $k= 1 \ldots n$ for all $p = \delta \ldots L\delta$
   - Compute $Q(k,p)$ = minimum $\Sigma_{i\le k} q_i$ such that $\Sigma_{i\le k} v_i(q_i) \ge p$

(Requires binary search to find minimum q_k with $v_k(q_k) \ge p'$.)

Incentives vs. approximation

Two players; Three unit m=3

v_1: (1.9$ for 1 unit), (2$ for 2 units), (3$ for 3 units)

v_2: (2$ for 1 item), (2.9$ for 2 units), (3$ for 3 units)

Best allocation: 1.9$+2.9$ = 4.8$.

Approximation algorithm with $\delta=1$ will get only 2$+2$=4$.

Manipulation by player 1: say $v_1(1\ unit)=5\$$.

- Improves social welfare ➔ (with VCG payments) improves player 1's utility

Where can VCG take us?

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints	B	B	B
Incentive compatible VCG payments	**Not B** **Not better than *n/(n-1)* approximation**	**Not B**	**Not C** **Not better than 2 approximation**

Limitation of VCG-based mechanisms

THM (N+Ronen): A VCG-based mechanism is incentive compatible iff it *exactly* optimizes over its own range of allocations. (almost)

Proof:

- (If) exactly VCG theorem on the range
- (only if) Intuition: if players can improve outcome, they will…
- (only if) proof idea: hybrid argument (local opt → global opt)

Corollary (N+Dobzinski): No better than *2*-approximation for general valuations, or *n/(n-1)*-approximation for SM valuations.

Proof (of corollary):

- If range is full ➔ exact optimization ➔ we saw impossibility
- If range does not include $[q_1\ q_2 \dots q_n]$ then will loose factor of *n/(n-1)* on profile v_1=(1$ for q_1) … v_n=(1$ for q_n).

Where can VCG take us?

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints	**B**	**B**	**B**
Incentive compatible VCG payments	**C**	**C** PTAS	**D** 2-approximation

An incentive-compatible VCG-based mechanism

Algorithm (N+Dobzinski): bundle the items into n^2 bundles of size $t=m/n^2$ (+ a single remainder bundle).

Lemma 1: This is a 2-approximation

Proof: Re-allocate items of one bidder among others

Lemma 2: Can be computed in poly-time:

For all $k = 1 \ldots n$ for all $q = t \ldots m/t$

Compute $P(k,q)$ = maximum $\Sigma_{i \le k} v_i(tq_i)$ such that $\Sigma_{i \le k} q_i \le q$

PTAS for k-minded case: all players except for $O(1/\varepsilon)$ ones get round bundles.

General Incentive Compatibility

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints	B	B	B
Incentive compatible VCG payments	C	C	D
General incentive compatible			

The single-minded case

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints	B	B	B
Incentive compatible VCG payments	C	C	D
General incentive compatible	B FPTAS		

Single parameter Incentive-Compatibility

THM (LOS): A mechanism for the Single-minded case is incentive compatible iff it is

1. Monotone increasing in p_i and monotone decreasing in q_i
2. Payment is critical value: minimum p_i needed to win q_i

Proof (if):

- Payment does not depend on declared *p;* win iff *p* > *payment*
- Lying with lower *q* is silly; higher *q* can only increase payment

Corollary (almost): Incentive compatible FPTAS for SM case.

The FPTAS that rounds the prices to integer multiples of δ satisfies 1&2.

Problem: Choosing δ…

Solution: Briest, Krysta and Vöcking, STOC 2005….

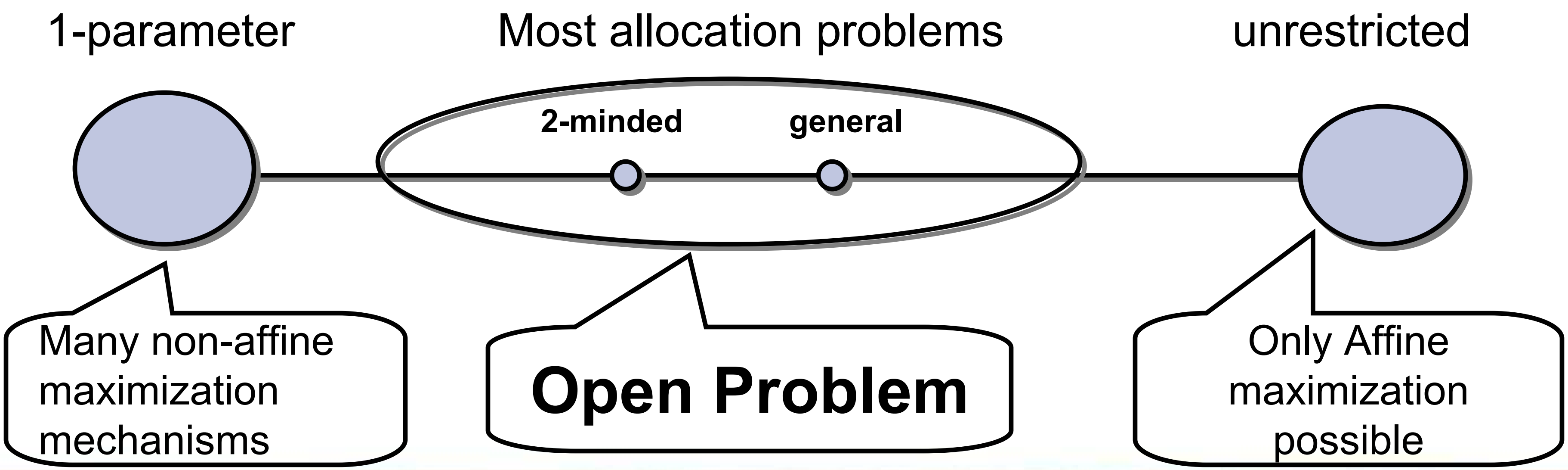
What can be implemented?

Representation → Incentives ↓	Single-minded	k-minded	general
No incentive constraints	B	B	B
Incentive compatible VCG payments	C	C	D
General incentive compatible	B	Conjecture: C No better than VCG	Conjecture + Partial result: D No better than VCG

Efficiently Computable Approximation Mechanisms?

Theorem (Roberts '77): If the space of valuations is unrestricted and $|A| \geq 3$ then the only incentive compatible mechanisms are *affine maximizers*: $\Sigma_i\, \alpha_i v_i(a) + \beta_a$

Comment: weighted versions of VCG. Easy to see that Weights cannot help computation/approximation.



Partial Lower Bound

Theorem (Lavi+Mu'alem+N): Every efficiently computable incentive compatible mechanism among two players that always allocates all units has approximation ratio ≥2.

Proof idea: If range is full, must be (essentially) affine maximizer.

- Non-full range → no better than 2-approximation
- Full range → computationally as hard as exact social welfare maximization

Rest of talk: proof assuming full range even after a single player is fixed.

Characterization of incentive compatibility

Notation: The algorithm allocates *a=f(v w) units to player 1.*

Player *1* pays*:* $p_1(v\ w)$

Characterization 1: For every *w* there exist payments p_a (for all *a*) such that for all *v: f(v w)* maximizes $v(a)- p_a$

Proof: $p_a(w) = p_1(v\ w)$, with *f(v w)=a,* can not depend on *v.*

Characterization 2 (*WMON*): If: f(v w)=a≠b=f(v' w)

Then: v(a)-v(b)≥v'(a)-v'(b)

Proof: $v(a)- p_a \geq v(b)- p_b$

$v'(a)- p_a \leq v'(b)- p_b$

➔ $v(a)-v'(a) \geq v(b)-v'(b)$

Properties of $p_a(w)$

Our Goal: for all a, $p_a(w) = \beta_a - \alpha\, w(m-a)$

Proof (Goal→Theorem): By characterization 1, f(v w) maximizes $v(a) - p_a = \beta_a + v(a) + \alpha\, w(m-a)$

Lemma: If: $w(m-a)-w(m-b) > w'(m-a)-w'(m-b)$

Then: $p_a(w)-p_b(w) \le p_a(w')-p_b(w')$

Proof (Lemma → Goal): Math (next slide)

Proof (of Lemma): Otherwise choose v such that:

$p_a(w)-p_b(w) > v(a)-v(b) > p_a(w')-p_b(w')$ (and low other $v(c)$)

By characterization 1: $f(v\ w)=b$ and $f(v\ w')=a$. Contradiction to WMON.

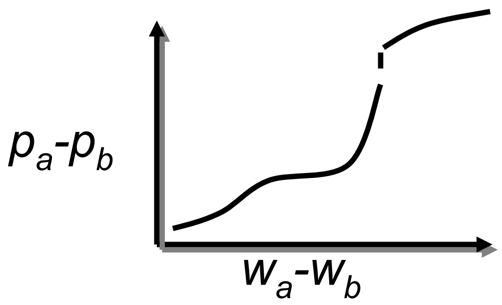
Monotonicity in differences (sketch)

Lemma: If $p : \mathfrak{R}^m \rightarrow \mathfrak{R}^m$ $(m \geq 3)$ satisfies

$w_a - w_b > w'_a - w'_b \Rightarrow p_a(w) - p_b(w) \geq p_a(w') - p_b(w')$

Then for all a, $p_a(w) = \beta_a + \alpha\, w_a$

Proof:



➔ $p_a(w)-p_b(w)$ depends only on $w_a - w_b$ (except for countably many values.)

Claim: $\partial p_a / \partial w_a = \partial p_b / \partial w_b$ (except for measure 0 of w)

Proof: $p_a(w)-p_b(w)$ stays constant when w_a *and* w_b are increased by the same amount.

Corollary: $\partial p_a / \partial w_a$ is constant

Remaining Open Problem:

Are there any useful non-VCG mechanisms for CAs, MUAs, or other resource allocation problems?

(E.g. poly-time approximations or heuristics)