Investment Incentives in Near-Optimal Mechanisms

Mohammad Akbarpour, *Stanford*
Scott Duke Kominers, *Harvard*
Shengwu Li, *Harvard*
Paul Milgrom, *Stanford*

October 2020

Resource allocation with transfers

Who gets what, and how much should they pay?

Use a Vickrey-Clarke-Groves mechanism!

1. Strategy-proof
2. Efficient allocations
3. Efficient investment incentives
    - Change just one bidder’s value.
    - Under VCG, Δ bidder’s payoff = Δ social welfare.
    - So the bidder gains by investing (paying a cost to change his value) if and only if net social welfare increases.

Resource allocation with transfers

Who gets what, and how much should they pay?

Use a Vickrey-Clarke-Groves mechanism!

1. Strategy-proof
2. Efficient allocations
3. Efficient investment incentives
   - Change just one bidder’s value.
   - Under VCG, Δ bidder’s payoff = Δ social welfare.
   - So the bidder gains by investing (paying a cost to change his value) if and only if net social welfare increases.

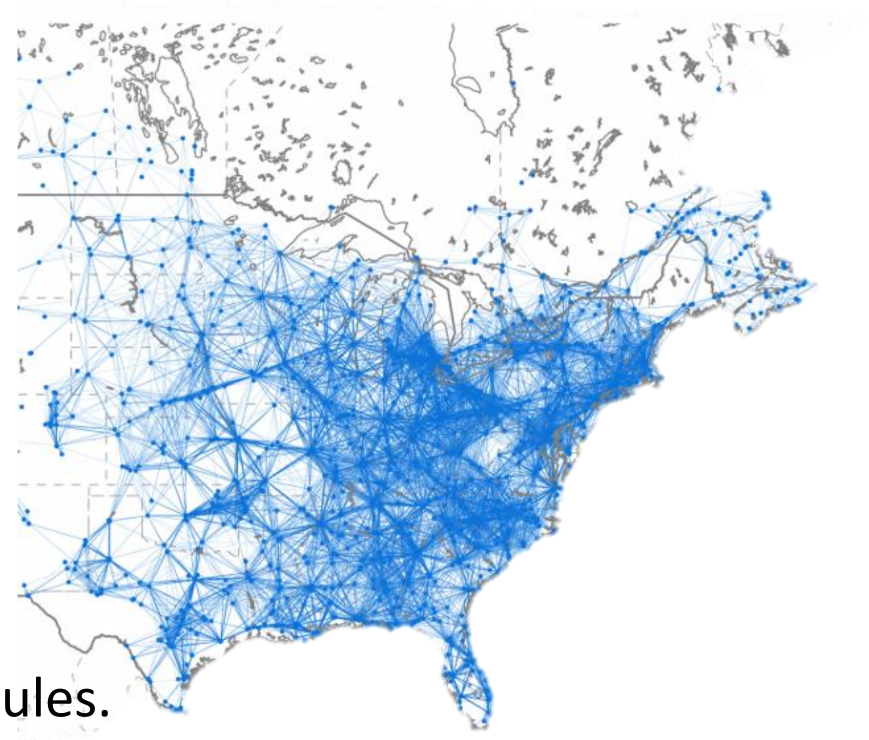
Caveats:

1. To invest efficiently, bidder must know his prices.
2. With multiple investors, coordination issues arise.
3. VCG mechanisms rely on exact optimization.

Exact optimization is sometimes infeasible.

Many real-world allocation problems are NP-hard.

- Spectrum auctions with local interference constraints
- Combinatorial auctions
- Electricity auctions



In practice, we use heuristic allocation rules.

Research question: If we use a heuristic allocation rule in a strategy-proof mechanism, what happens to incentives to invest efficiently?

Informal Overview of Results

Q: inefficient allocations + SP ⇒ inefficient investments?

Yes! Inefficient allocation rules have ‘algorithmic externalities’, which cause inefficient investments.

Informal Overview of Results

Q: inefficient allocations + SP ⇒ inefficient investments?

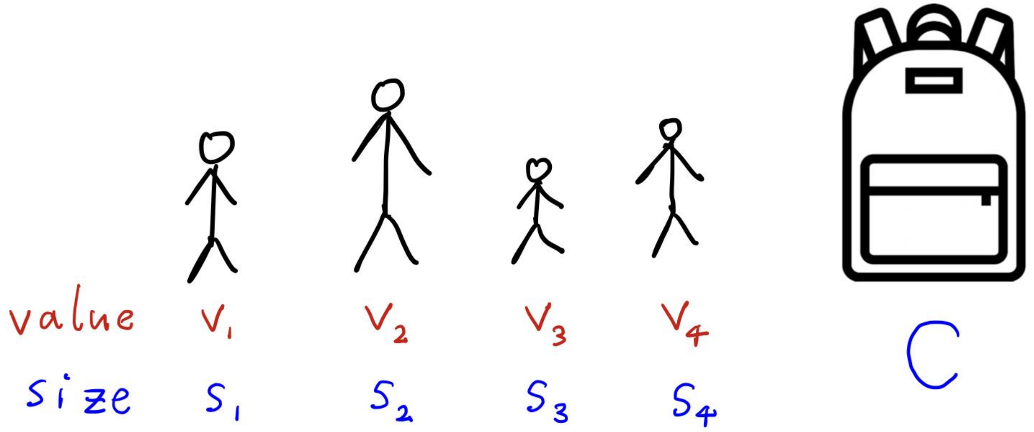
Yes! Inefficient allocation rules have ‘algorithmic externalities’, which cause inefficient investments.

Q: near-efficient allocations + SP ⇒ near-efficient investments?

1. No! Arbitrarily good allocation rules can have arbitrarily bad investment incentives.
2. If the allocation rule ‘excludes bossy negative externalities’ (XBONE), then investment incentives are ‘OK’.
3. A characterization result: ‘weak XBONE’ is necessary and sufficient.

Running example: The knapsack problem

An **instance** of the knapsack problem specifies a set of bidders N, a *value* and a *size* for each bidder, and knapsack capacity C.

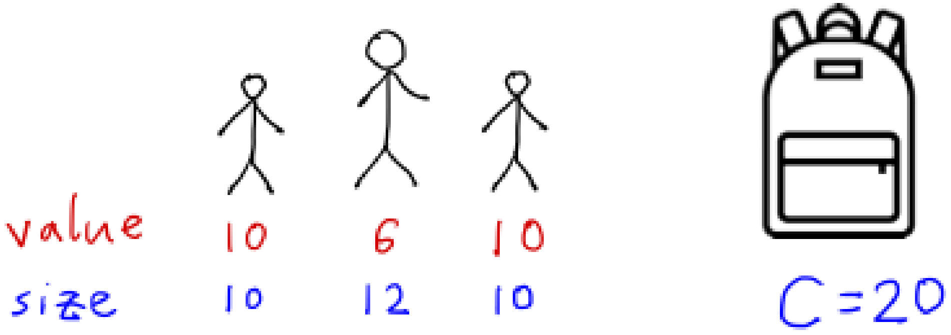


Choose $a \subseteq N$ to maximize $\sum_{n \in a} v_n$ subject to $\sum_{n \in a} s_n \leq C$.

Knapsack optimization is NP-hard. No known algorithm computes efficient allocations and runs in polynomial time on all instances.

A fast algorithm for the knapsack problem

Consider this instance:

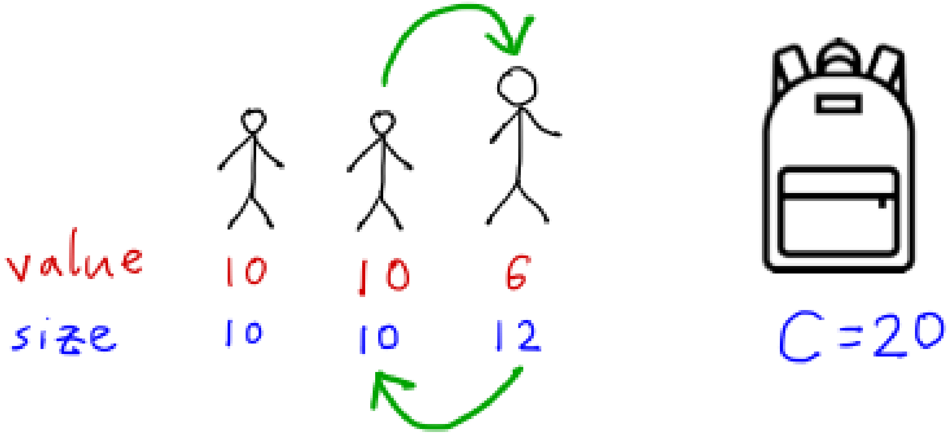


value	10	6	10
size	10	12	10

C=20

A fast algorithm for the knapsack problem

Consider this instance:

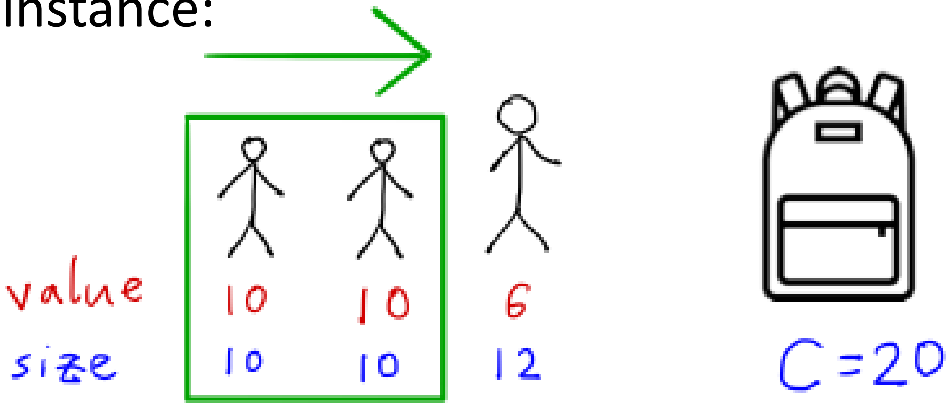


The **Greedy** algorithm (Dantzig 1957)

1. Sort bidders by $\frac{value}{size}$, in descending order.

A fast algorithm for the knapsack problem

Consider this instance:

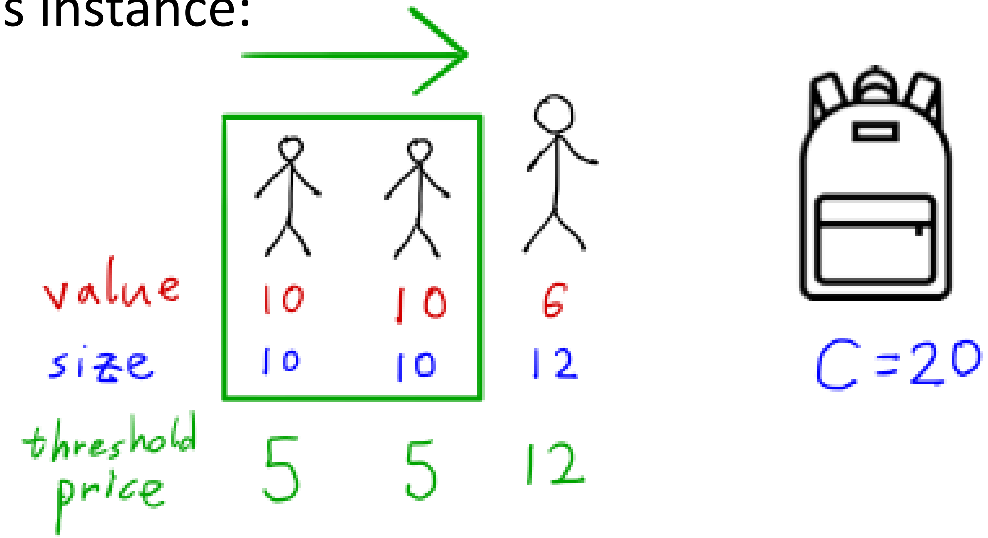


The **Greedy** algorithm (Dantzig 1957)

1. Sort bidders by $\frac{value}{size}$, in descending order.
2. Pack bidders from left to right.
3. Stop when the next bidder doesn't fit.

Running the Greedy algorithm when values are private

Consider this instance:



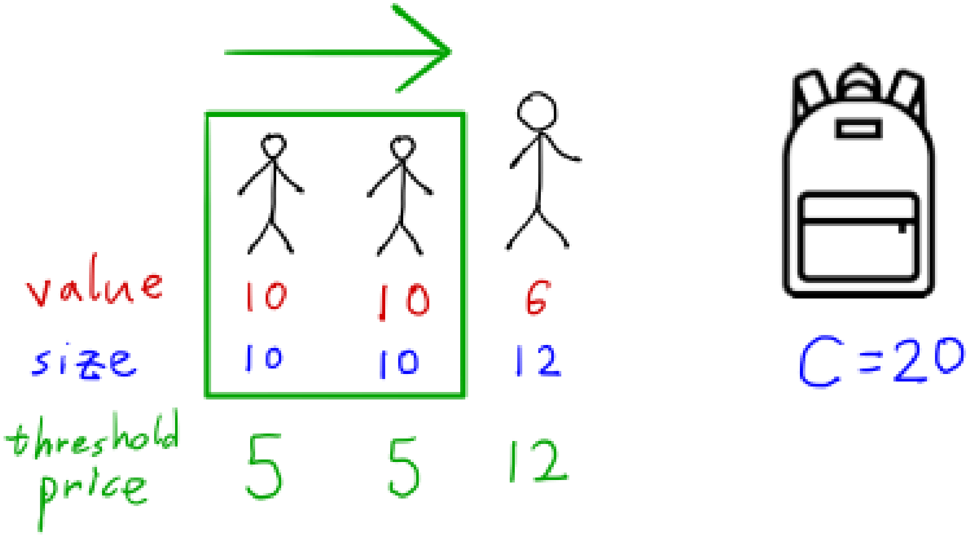
value	10	10	6
size	10	10	12
threshold price	5	5	12

Suppose sizes are observable, but values are private info.

No problem! Here’s a **strategy-proof** mechanism:

1. Ask bidders to report their values.
2. Run the Greedy algorithm on the reported values.
3. Charge each packed bidder his ***threshold price.***

Profitable investments can reduce welfare

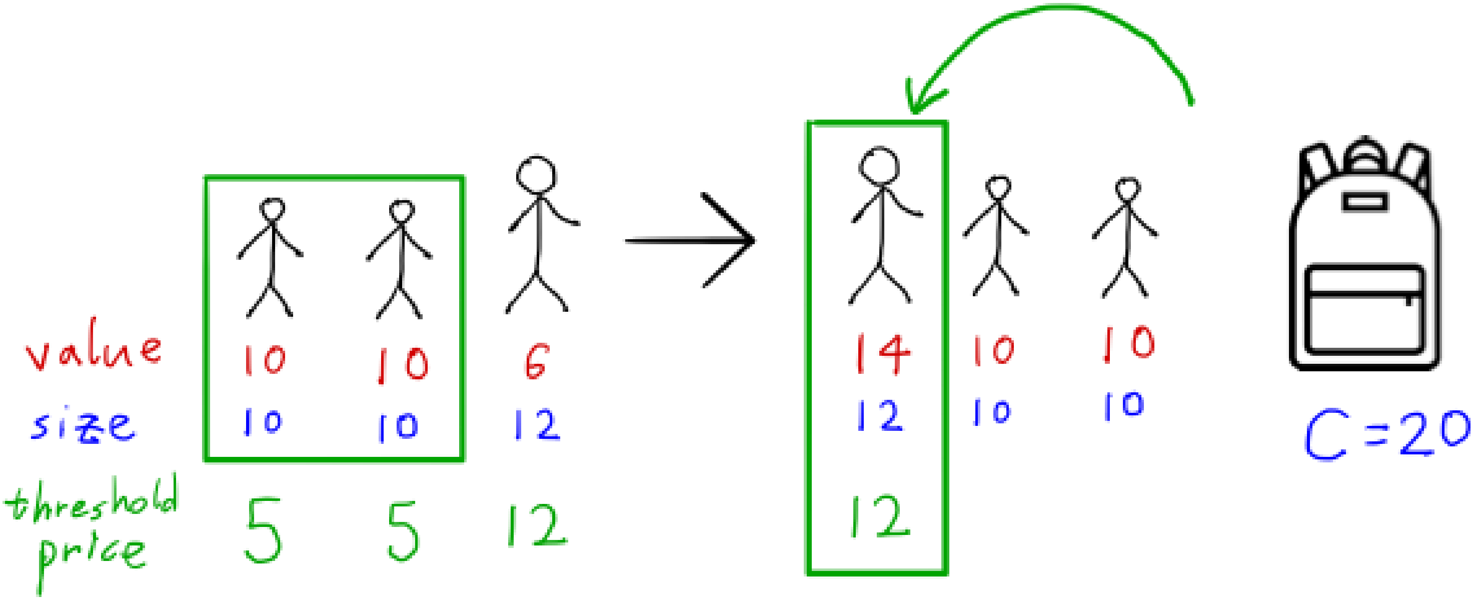


value	10	10	6
size	10	10	12
threshold price	5	5	12

The big bidder is not packed, has utility 0.

What if he can invest to raise his value to 14 at cost 1?

Profitable investments can reduce welfare



The big bidder is not packed, has utility 0.

What if he can invest to raise his value to 14 at cost 1?

This causes him to be packed and is profitable: $14\ -12\ -1>0$

The investment reduces welfare from 20 to $13=14\ -\ 1$.

Model primitives

An **allocation instance** (v, A) specifies, for a set of bidders N,

1. value profile $v \in \mathbb{R}^N_{\geq 0}$ (generalize to multi-dimensional types in paper)
2. feasible allocations $A \subseteq \mathrm{powerset}(N)$

e.g. “all sets of bidders whose sizes sum to less than 10”
e.g. “all sets of TV stations that can broadcast without interference”

Model primitives

An **allocation instance** (v, A) specifies, for a set of bidders N,

1. value profile $v \in \mathbb{R}^N_{\geq 0}$
2. feasible allocations $A \subseteq \text{powerset}(N)$

An **allocation problem** Ω is a collection of allocation instances.

Assumption: For all A, the set $\{v \mid (v, A) \in \Omega\}$ is a product of closed intervals.

An allocation rule (**algorithm**) chooses a feasible allocation for each instance, $x(v, A) \in A$. $x_n(\cdot)$ is an indicator $= 1$ if n is packed.

A **payment rule** specifies a payment for each bidder, $p(v, A) \in \mathbb{R}^N$.

A **mechanism** (x, p) consists of an algorithm and a payment rule.

How to convert an algorithm into a strategy-proof mechanism

Strategy-proof (SP)

$$\forall (v, A) : \forall n : v_n \in \arg\max_{\tilde{v}_n} v_n x_n(\tilde{v}_n, v_{-n}, A) - p_n(\tilde{v}_n, v_{-n}, A)$$

x is **monotone** if for any bidder n, there exists a packing threshold $t_n(v_{-n}, A)$ such that

$$v_n > t_n(v_{-n}, A) \Rightarrow n \in x(v, A)$$

$$v_n < t_n(v_{-n}, A) \Rightarrow n \notin x(v, A)$$

We charge each packed bidder a price equal to her threshold.

How to convert an algorithm into a strategy-proof mechanism

Strategy-proof (SP)

$$\forall (v, A) : \forall n : v_n \in \arg\max_{\tilde{v}_n} v_n x_n(\tilde{v}_n, v_{-n}, A) - p_n(\tilde{v}_n, v_{-n}, A)$$

x is **monotone** if for any bidder n, there exists a packing threshold $t_n(v_{-n}, A)$ such that

$$v_n > t_n(v_{-n}, A) \Rightarrow n \in x(v, A)$$

$$v_n < t_n(v_{-n}, A) \Rightarrow n \notin x(v, A)$$

We charge each packed bidder a price equal to her threshold.

Proposition (Lehmann *et al* 2006): There exists a payment rule p such that (x, p) is strategy-proof if and only if x is monotone.

Henceforth restrict attention to monotone algorithms.

The SP payment rule is essentially unique.

Proposition: If (x, p) is strategy-proof, then there exist functions $f_n(v_{-n}, A)$ such that

$$p_n(v, A) = \underbrace{x_n(v, A)}_{\text{indicator}} \quad \underbrace{t_n(v_{-n}, A)}_{\text{packing threshold}} \quad + f_n(v_{-n}, A)$$

Corollary: Any two SP mechanisms for the same x yield the same investment choices.

Investment incentives in SP mechanisms are entirely pinned down by the allocation algorithm.

Efficiency of Allocation and Investment

Welfare achieved by algorithm $W_x(v, A) \equiv \sum_{n \in x(v,A)} v_n$

Efficient allocations (EA)

$$\forall (v, A) : W_x(v, A) = \max_{a \in A} \sum_{n \in a} v_n$$

Efficient investments (EI)

Suppose bidder n, knowing v_{-n} , can invest to change his value.

He should profit from that investment if and only if the welfare gain exceeds the cost.

Interpretation: long-run adjustment to prevailing prices.

Efficiency of Allocation and Investment

Welfare achieved by algorithm $W_x(v, A) \equiv \sum_{n \in x(v,A)} v_n$

Efficient allocations (EA)

$$\forall (v, A) : W_x(v, A) = \max_{a \in A} \sum_{n \in a} v_n$$

Efficient investments (EI)

For any instance (v, A), any $\tilde{v}_n$, and any $c \geq 0$,

$$\tilde{v}_n x_n(\tilde{v}_n, v_{-n}, A) - p_n(\tilde{v}_n, v_{-n}, A) - c \geq v_n x_n(v, A) - p_n(v, A)$$

$$\Updownarrow$$

$$W_x(\tilde{v}_n, v_{-n}, A) - c \ \geq W_x(v, A)$$

Interpretation: long-run adjustment to prevailing prices.

VCG mechanisms yield efficient investments

- Efficient allocations (EA)
- Strategy-proof (SP)
- Efficient investments (EI)

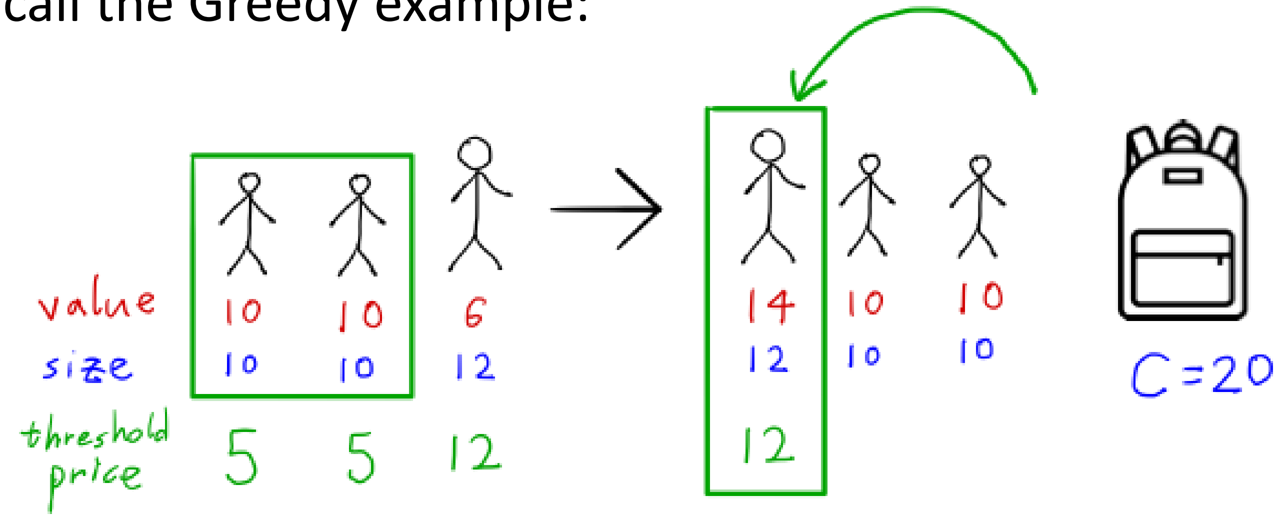
Proposition (Rogerson 1992): The VCG mechanism satisfies EA, SP, and EI.

Intuition: The VCG mechanism internalizes all externalities. Each bidder’s threshold price equals the welfare loss to others when he is packed.

Heuristic mechanisms can have ‘algorithmic externalities’.

For heuristic algorithms, threshold prices may not reflect the full social cost (welfare losses to other bidders).

Recall the Greedy example:



For which allocation algorithms do threshold prices internalize all externalities? (yielding efficient investments)

A weakening of efficient allocations

Efficient allocations (EA)

$$\forall (v, A) : W_x(v, A) = \max_{a \in A} \sum_{n \in a} v_n$$

Range-efficient allocations (REA)

$$\forall A : \exists R \subseteq A : \forall v : W_x(v, A) = \max_{a \in R} \sum_{n \in a} v_n$$

Note order of quantifiers: R depends on A but not on v.
If for some generic v, the algorithm outputs a^*, we can infer $a^* \in R$.
So for all v', the algorithm must do no worse than a^*.
e.g. For knapsack, Greedy does **not** satisfy REA.

Only REA algorithms avoid algorithmic externalities.

- Range-efficient allocations (REA)
- Strategy-proof (SP)
- Efficient investments (EI)

Theorem: For any mechanism with monotone x,

1. REA and SP ⇒ EI
2. REA and EI ⇒ SP
3. EI and SP ⇒ REA

Only REA algorithms avoid algorithmic externalities.

- Range-efficient allocations (REA)
- Strategy-proof (SP)
- Efficient investments (EI)

Theorem: For any mechanism with monotone x,

1. REA and SP ⇒ EI
2. REA and EI ⇒ SP
3. EI and SP ⇒ REA

A strategy-proof mechanism can have efficient investments only if it constrains the allocations enough to make exact optimization feasible.

In computationally hard problems, this often leads to poor allocative efficiency.

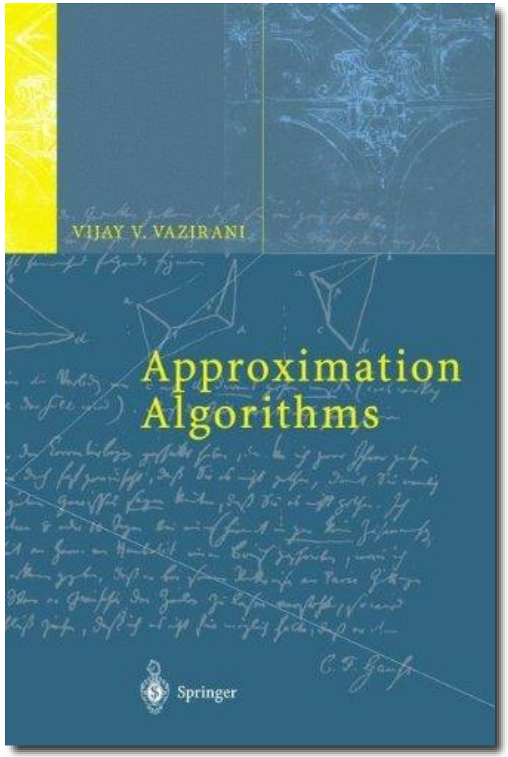
Recap

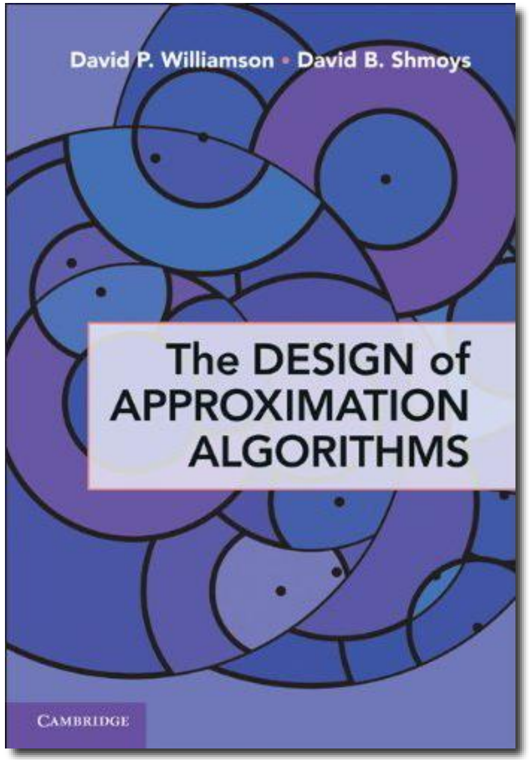
1. Monotone algorithms can be made into strategy-proof mechanisms.
2. For these mechanisms, investment incentives are entirely determined by the algorithm.
3. Algorithmic externalities cause inefficient investments.
4. The only algorithms with efficient investments are those that exactly optimize (possibly on a restricted subset of allocations).

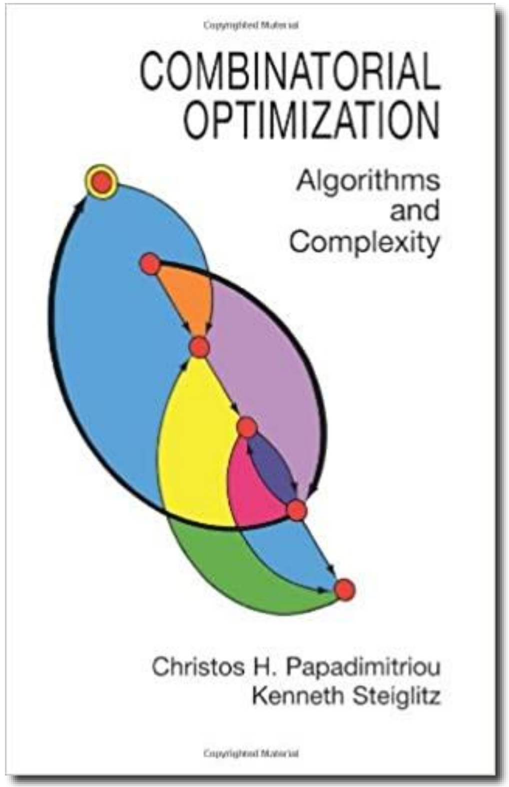
When exact optimization is infeasible, we approximate.

A vast CS literature studies fast algorithms for hard problems.

These typically violate REA but are approximately optimal.




David P. Williamson • David B. Shmoys
The DESIGN of APPROXIMATION ALGORITHMS
CAMBRIDGE


Copyrighted Material
COMBINATORIAL OPTIMIZATION
Algorithms and Complexity
Christos H. Papadimitriou
Kenneth Steiglitz
Copyrighted Material

When exact optimization is infeasible, we approximate.

A vast CS literature studies fast algorithms for hard problems.

These typically violate REA but are approximately optimal.

For $\beta \in [0, 1]$, algorithm x is a **β-approximation** if for every instance $(v, A) \in \Omega$, we have that

$$W_x(v, A) \geq \beta \underbrace{W^*(v, A)}_{\substack{\text{welfare under} \\ \text{efficient allocation}}}$$

E.g. the SmartGreedy algorithm packs the max of the Greedy output and the single bidder with the highest value.

SmartGreedy is a $\frac{1}{2}$-approximation for the knapsack problem.

Which algorithms lead to good investment incentives?

Suppose we start with a monotone algorithm and run the corresponding strategy-proof mechanism.

efficient allocations + SP ⇒ efficient investments

Is it true that

near-efficient allocations + SP ⇒ near-efficient investments?

Let’s measure net welfare including investment.

Suppose some bidder chooses between

- value $\overline{v}_n$ at cost $\overline{c} \geq 0$
- value $\underline{v}_n$ at cost 0 (normalized)

This is an **investment technology**, denoted I_n.

Given I_n and his threshold price, the bidder chooses an investment to maximize his payoff under the SP mechanism.

$$B^* \equiv \underset{(v_n, c) \in I_n}{\arg\max}\; v_n x_n(v_n, v_{-n}, A) - p_n(v_n, v_{-n}, A) - c$$

Welfare including investment is

$$\overline{W}_x(I_n, v_{-n}, A) \equiv \min_{(v_n, c) \in B^*} W_x(v_n, v_{-n}, A) - c$$

Approximately-efficient investments

What's our benchmark? Efficient investments, efficient allocations.

$$\overline{W}^*(I_n, v_{-n}, A) \equiv \max_{(v_n, c) \in I_n} \underbrace{W^*(v_n, v_{-n}, A)}_{\substack{\text{welfare under} \\ \text{efficient allocation}}} - c$$

Algorithm x is a **β-approximation with investment** if for any **investment instance** (I_n, v_{-n}, A) we have

$$\overline{W}_x(I_n, v_{-n}, A) \geq \beta \overline{W}^*(I_n, v_{-n}, A)$$

Allows inefficient investments, provided welfare net of investment costs is at least β of the first-best (with $\beta \leq 1$).

What happens to guarantees when we add investments?

They can't get better.

Proposition: For any β, if x is not a β-approximation, then x is not a β-approximation with investment.

They can get a lot worse.

Proposition: For all $\beta < 1$, there exists a β-approximation for the knapsack problem that is no more than a 0-approximation with investment.

β-approx allocations + SP $\not\Rightarrow$ β-approx with investments

How can guarantees be worse under investment?

Consider this 'satisficing' algorithm. If Bidder 1's value exceeds 99% of the total, pack her only. Otherwise, optimize exactly.

Suppose two bidders and we can fit both in knapsack.

$$x(v_1, v_2) = \begin{cases} \{1\} & \text{if } \frac{v_1}{v_1+v_2} \geq .99 \\ \{1,2\} & \text{otherwise} \end{cases}$$

Bidder 1 is always packed, so her threshold price is 0.

Suppose she can have value 0 at cost 0, or value $K + \epsilon$ at cost K.

Profitable to invest.

For large enough K, only bidder 1 is packed.

Welfare is $K + \epsilon - K = \epsilon$. First-best is $v_2 + \epsilon$.

Approximation ratio with investment is $\leq \dfrac{\epsilon}{v_2 + \epsilon}$.

How can guarantees be worse under investment?

Consider this ‘satisficing’ algorithm. If Bidder 1’s value exceeds 99% of the total, pack her only. Otherwise, optimize exactly.

Suppose two bidders and we can fit both in knapsack.

$$x(v_1, v_2) = \begin{cases} \{1\} & \text{if } \frac{v_1}{v_1+v_2} \geq .99 \\ \{1,2\} & \text{otherwise} \end{cases}$$

This algorithm has a ‘bossy negative externality’.

Raising v_1 from 0 to $K + \epsilon$ doesn’t change whether bidder 1 is packed, yet it reduces the welfare of bidder 2!

Not just a problem for our toy algorithm. The canonical FPTAS for the knapsack has the same vulnerability.

When is a guarantee robust to investment?

"If we raise the value of a packed bidder or lower the value of an unpacked bidder, then the total welfare of *other* bidders does not fall."

x is **XBONE** (eXcludes BOssy Negative Externalities) if for any allocation instance (v, A) and any v'_n, if

1. $n \in x(v, A)$ and $v_n < v'_n$
2. or: $n \notin x(v, A)$ and $v_n > v'_n$

then we have

$$\sum_{m \in x(v,A) \setminus \{n\}} v_m \leq \sum_{m \in x(v'_n, v_{-n}, A) \setminus \{n\}} v_m$$

Theorem: For any β, if x is XBONE and a β-approximation, then x is a β-approximation with investment.

Examples of XBONE algorithms

- Any exactly optimal algorithm is XBONE.
- Greedy is XBONE.
- The max of any set of XBONE algorithms is XBONE.
- SmartGreedy is XBONE and a $\frac{1}{2}$-approximation.

$$\mathrm{SmartGreedy} = \max\{\mathrm{Greedy}, \{1\}, \{2\}, \ldots\}$$

- Any ‘non-bossy’ monotone algorithm is XBONE.
  - Non-bossy: if we change a bidder’s value without changing whether he is packed, then the allocation is not changed.
  - E.g. Greedy rejection algorithms (Milgrom and Segal 2020) used for the 2017 FCC reverse auction

Theorem: For any β, if x is XBONE and a β-approximation, then x is a β-approximation with investment.

Examples of XBONE algorithms

- Any exactly optimal algorithm is XBONE.
- Greedy is XBONE.
- The max of any set of XBONE algorithms is XBONE.
- SmartGreedy is XBONE and a $\frac{1}{2}$-approximation.

$$\text{SmartGreedy} = \max\{\text{Greedy}, \{1\}, \{2\}, \ldots\}$$

- Any ‘non-bossy’ monotone algorithm is XBONE.
    - Non-bossy: if we change a bidder’s value without changing whether he is packed, then the allocation is not changed.
    - E.g. Greedy rejection algorithms (Milgrom and Segal 2020) used for the 2017 FCC reverse auction

Theorem: For any β, if x is XBONE and a β-approximation, then x is a β-approximation with investment.

Intuition for theorem

- In a heuristic mechanism, threshold prices may not reflect social cost of packing a bidder.
- Consider an investment that moves value to just above or just below threshold, at a tiny cost.
  - There is an instance of the allocation problem with equally bad performance.
  - Efficiency loss already reflected in allocative guarantee.
- So investments reduce welfare by value changes that do not cross a packing threshold.
  - In a strategy-proof mechanism, these changes are not priced.
- XBONE forbids harm to other bidders from exactly those changes.

Theorem: For any β, if x is XBONE and a β-approximation, then x is a β-approximation with investment.

Sub-problems can have better guarantees

Allocation problem Ω' is a **sub-problem** of Ω if $\Omega' \subseteq \Omega$.

Approximation ratios can only improve on sub-problems.

e.g. if no bidder's size exceeds 10% of the knapsack's capacity, then Greedy is a 0.9-approximation.

Corollary of Theorem: If x is XBONE on Ω, then for any sub-problem Ω' and any β': If x is a β'-approximation on Ω', then x is a β'-approximation with investment on Ω'.

Greedy is a good heuristic when bidders are small, even with investment.

A characterization theorem

Can we weaken XBONE, and still have allocation guarantees be robust to investment?

Theorem: Let x be a β-approximation on Ω for some $\beta > 0$.
These statements are equivalent:

1. x is 'weakly XBONE' on Ω.
2. For any sub-problem Ω' and any β': If x is a β'-approximation on Ω', then x is a β'-approximation with investment on Ω'.

Corollary: If x is *not* 'weakly XBONE', then there exists a sub-problem such that x's approximation ratio gets worse when we add investment.

Weakly XBONE

XBONE can only be relaxed for certain kinds of ‘downward’ changes.

x is **weakly XBONE** if for any (v, A) and any v'_n , if

1. $n \in x(v, A)$ and $v_n < v'_n$,

2. or: $n \notin x(v, A)$ and n is in every efficient allocation at (v, A)

   and $v_n > v'_n$.

then we have

$$\sum_{m \in x(v,A)\setminus\{n\}} v_m \leq \sum_{m \in x(v'_n, v_{-n}, A)\setminus\{n\}} v_m$$

In problems of interest, we can’t compute efficient allocations.

Extensions

Results extend **to cost-minimization problems** (flip signs).

- Traveling salesman problem
- Steiner tree problem (building a network)

Results extend to **multi-dimensional types** (multiple ways to be packed).

- Efficient investments still require exact optimization.
- A generalization of XBONE suffices for guarantees to be robust to investment.

Results extend to **combinatorial auctions** with XOS preferences (includes gross substitutes), given a restriction on investment costs.

Conclusion

In a strategy-proof mechanism, efficient investment incentives require exact optimization. In hard problems, this is infeasible.

Algorithms can choose near-optimal allocations, and yet be far from optimal with investments. Why? Bossy negative externalities.

If the algorithm excludes bossy negative externalities (XBONE), then its guarantee on any sub-problem is the same when we add investments.

Weak XBONE is necessary and sufficient for the conclusion.