Computation and Incentives in Public Projects

Dave Buchfuhrer Michael Schapira Yaron Singer

Caltech, Yale, Berkeley

May 18, 2010

Outline

Outline

Public Projects

A combinatorial public project is a game in which the goal is to choose k items from a set to provide for shared use among n agents.

Public Projects

A combinatorial public project is a game in which the goal is to choose k items from a set to provide for shared use among n agents.

Example

Suppose you are the administrator for a small park. You have room for 3 pieces of equipment. Which pieces do you choose in order to make local families happiest?

Park Design

So many choices for equipment

Park Design

So many choices for equipment




And every parent has an opinion

- “There must be a swing set”
- “If there isn't tetherball, the terrorists have already won”
- “My kids need a merry go round and a playground equipment”
- “I'll picket if there's anything dangerous”
- “Kids need exercise. No equipment that can be used while sitting!”

What to do?

You can't make everyone happy, but you can try to make the park as good for the public as possible.

Definition (Social Welfare)

Suppose that each agent i gets value $v_i(S)$ for allocation S. Then the social welfare of S is

$$\sum_i v_i(S)$$

More than computation

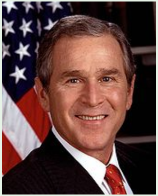
So now we have a computational problem. Given the valuations $v_1, \ldots, v_n$ of the n agents, find a set S of size k maximizing the social welfare. Unfortunately, we have the added difficulty that people will lie.

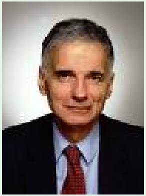
More than computation

So now we have a computational problem. Given the valuations $v_1, \ldots, v_n$ of the n agents, find a set S of size k maximizing the social welfare. Unfortunately, we have the added difficulty that people will lie.

Example (Elections)

In the US, people will lie on ballots that they desire one of the major party candidates, rather than "throw their votes away" on preferred third party candidates.





Truthfulness

We would like to design a mechanism whereby we are able to elicit the true valuations and approximate the social welfare well.

Truthfulness

We would like to design a mechanism whereby we are able to elicit the true valuations and approximate the social welfare well.

In order to do so, we allow money to change hands. We already know how to achieve truthful prices via the VCG mechanism. This allows for any allocation algorithm to be made truthful iff it is maximal in range.

Truthfulness

We would like to design a mechanism whereby we are able to elicit the true valuations and approximate the social welfare well.

In order to do so, we allow money to change hands. We already know how to achieve truthful prices via the VCG mechanism. This allows for any allocation algorithm to be made truthful iff it is maximal in range.

Definition (Maximal in Range)

An allocation algorithm is maximal in range if there exists a set R such that it always outputs a member of R maximizing the social welfare.

VCG-based mechanisms

VCG-based mechanisms are the only known general method for achieving truthfulness in games like this. Furthermore, they are sometimes the only way to get truthfulness.

Theorem (Roberts, 1979)

The only truthful mechanism for general games is the VCG mechanism.

Papadimitriou, Schapira and Singer showed this for public projects in 2008

Performance of MIR mechanisms

Theorem

A maximal in range allocation algorithm for any NP-hard combinatorial public project cannot approximate the welfare with a ratio better than $\sqrt{m}$ *unless* $NP \subseteq P/poly$.

Proof (Sketch).

- A mechanism that gets better than a $\sqrt{m}$ ratio requires an exponential range for sufficiently expressive valuation classes (PSS 08)

□

Performance of MIR mechanisms

Theorem

A maximal in range allocation algorithm for any NP-hard combinatorial public project cannot approximate the welfare with a ratio better than $\sqrt{m}$ unless $NP \subseteq P/poly$.

Proof (Sketch).

- A mechanism that gets better than a $\sqrt{m}$ ratio requires an exponential range for sufficiently expressive valuation classes (PSS 08)
- By Sauer's lemma, an exponential range must contain a polynomial-sized subset S^* of items allocated in every way

□

Performance of MIR mechanisms

Theorem

A maximal in range allocation algorithm for any NP-hard combinatorial public project cannot approximate the welfare with a ratio better than $\sqrt{m}$ unless $NP \subseteq P/poly$.

Proof (Sketch).

- A mechanism that gets better than a $\sqrt{m}$ ratio requires an exponential range for sufficiently expressive valuation classes (PSS 08)
- By Sauer's lemma, an exponential range must contain a polynomial-sized subset S^* of items allocated in every way
- We construct instances in which it is NP-hard to determine which members of S^* should be selected. These follow fairly directly from the proofs of NP-hardness. □

Using the proof

The proof framework described requires different proofs depending on what class of valuations the agents are allowed to have.

Outline

Motivating Example

A local food court has k empty storefronts. The shoppers each only care about their favorite restaurant in the food court.

Definition

Definition (Unit-Demand Valuation)

An agent with a unit-demand valuation has private values w_j for each item j, and has total value

$$v_i(S) = \max_{j \in S} w_j$$

for set S.

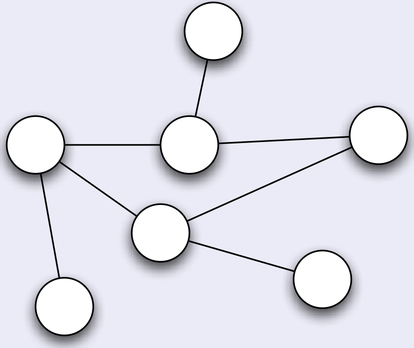
NP hardness

Theorem

The public project problem with unit-demand agents is NP-hard.

Proof by picture

Reduction from vertex cover:



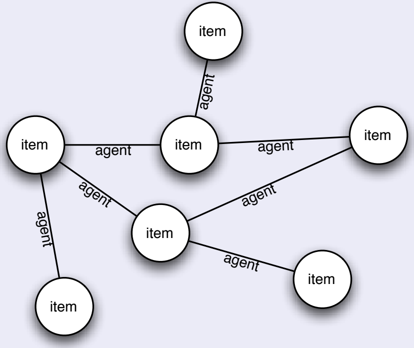
NP hardness

Theorem

The public project problem with unit-demand agents is NP-hard.

Proof by picture

Reduction from vertex cover:



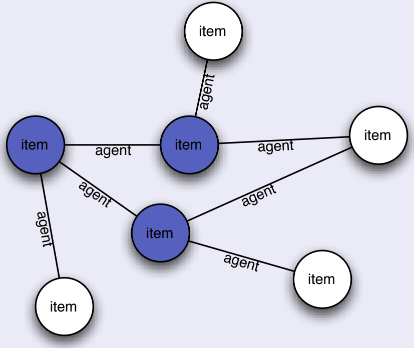
NP hardness

Theorem

The public project problem with unit-demand agents is NP-hard.

Proof by picture

Reduction from vertex cover:



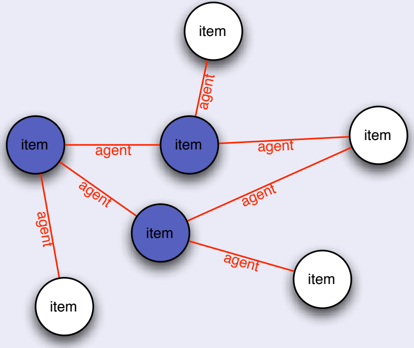
NP hardness

Theorem

The public project problem with unit-demand agents is NP-hard.

Proof by picture

Reduction from vertex cover:



Truthful Approximation

Recall

NP hardness means VCG mechanisms can't beat a $\sqrt{m}$ approximation

Truthful Approximation

Recall

NP hardness means VCG mechanisms can't beat a $\sqrt{m}$ approximation

Theorem

There exists a truthful 2-approximation for public projects produced by our vertex cover reduction

Truthful Approximation

Recall

NP hardness means VCG mechanisms can't beat a $\sqrt{m}$ approximation

Theorem

There exists a truthful 2-approximation for public projects produced by our vertex cover reduction

Mechanism

Choose the k items corresponding to the vertices of highest degree

Truthful Approximation

Recall

NP hardness means VCG mechanisms can't beat a $\sqrt{m}$ approximation

Theorem

There exists a truthful 2-approximation for public projects produced by our vertex cover reduction

Mechanism

Choose the k items corresponding to the vertices of highest degree

Proof (2-approximation).

- You can't cover more edges than the maximum sum of degrees
- The number of edges covered is at least half that sum □

Lemma

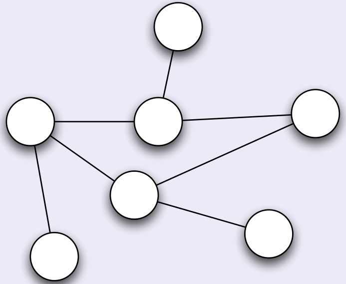
The below mechanism is truthful.

Mechanism

Choose the k items corresponding to the vertices of highest degree

Proof.

Consider an agent lying.



□

Lemma

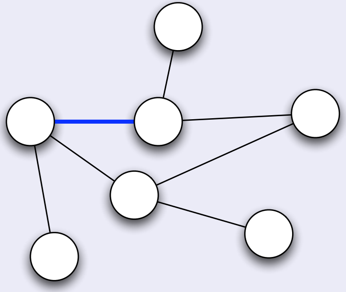
The below mechanism is truthful.

Mechanism

Choose the k items corresponding to the vertices of highest degree

Proof.

Consider an agent lying.



□

Lemma

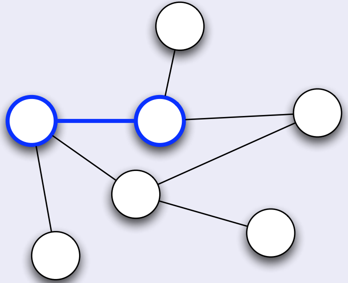
The below mechanism is truthful.

Mechanism

Choose the k items corresponding to the vertices of highest degree

Proof.

Consider an agent lying.



□

Lemma

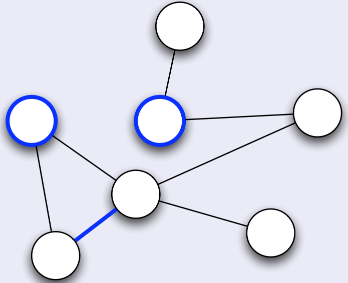
The below mechanism is truthful.

Mechanism

Choose the k items corresponding to the vertices of highest degree

Proof.

Consider an agent lying.



□

Lemma

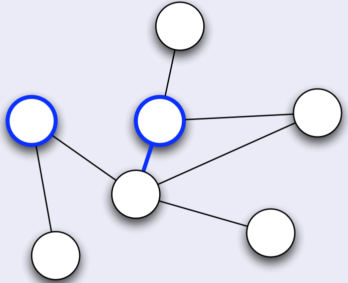
The below mechanism is truthful.

Mechanism

Choose the k items corresponding to the vertices of highest degree

Proof.

Consider an agent lying.



□

Summary

- Unit-demand is NP-hard, meaning that VCG can't beat a $\sqrt{m}$ approximation
- For one limited subclass, we can get around this limitation with a truthful 2-approximation

Outline

Motivating Example

Again, the food court has k empty storefronts. The neighbors have grown more sophisticated in their tastes and demand some variety. They want to eat at different restaurants for breakfast, lunch and dinner.

Definition

Definition (Multi-Unit-Demand Valuation)

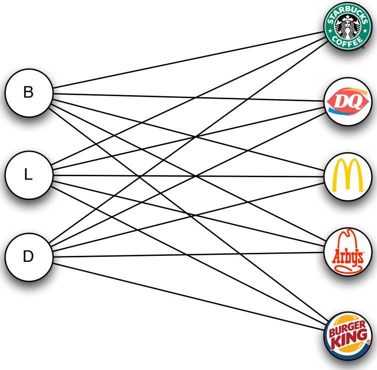
An agent with a multi-unit-demand valuation has private unit-demand valuation functions $v_i^{(1)}, \ldots, v_i^{(\ell)}$, and has value

$$v_i(S) = \max_{\substack{S_1,\ldots,S_\ell \subseteq S \\ S_j \cap S_{j'} = \emptyset \ \forall j \neq j'}} \sum_j v_i^{(j)}(S_j)$$

for set S.

Computing Values

Computing $v_i(S)$ for an agent may seem difficult, but it can be accomplished easily via matching:



Flow

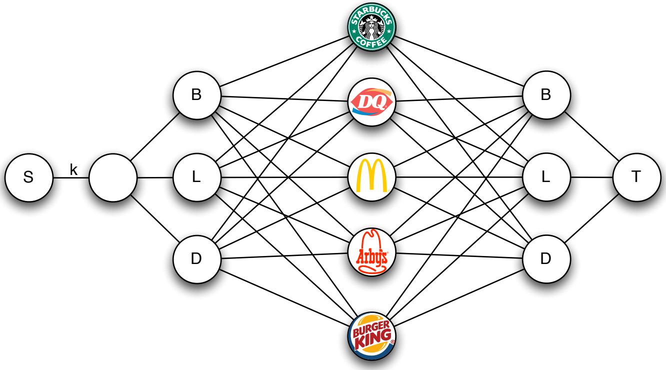
This can be extended to a solution for 2 agents via network flow.

Flow

This can be extended to a solution for 2 agents via network flow.

Theorem

The public projects problem with 2 multi-unit-demand agents is in P.

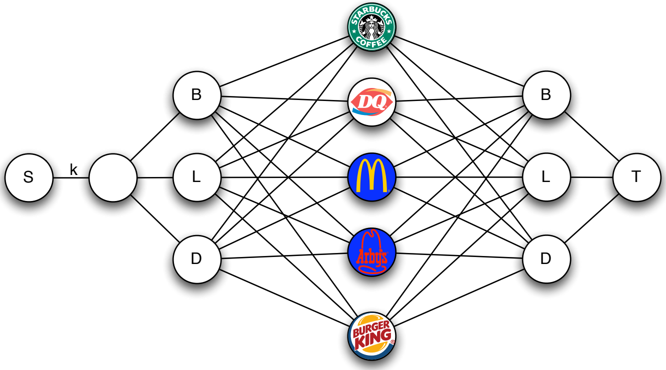


Flow

This can be extended to a solution for 2 agents via network flow.

Theorem

The public projects problem with 2 multi-unit-demand agents is in P.



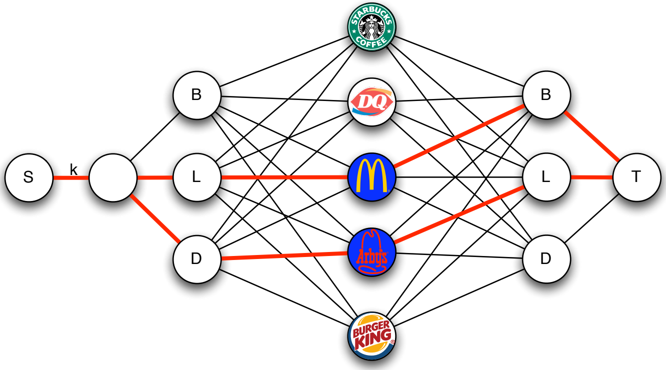
Each choice of k items corresponds to a flow.

Flow

This can be extended to a solution for 2 agents via network flow.

Theorem

The public projects problem with 2 multi-unit-demand agents is in P.



Each choice of k items corresponds to a flow.

NP Hardness

Theorem

The public projects problem with 3 multi-unit-demand agents is NP-hard.

We use a reduction from 3-dimensional matching.

NP Hardness

Theorem

The public projects problem with 3 multi-unit-demand agents is NP-hard.

We use a reduction from 3-dimensional matching.

Definition (3-Dimensional Matching)

Input: A set $M \subseteq [q] \times [q] \times [q]$.
Decision Problem: Does there exist some $M' \subseteq M$ such that $|M'| = q$ and no two members of M' agree on any coordinate?

NP Hardness

Theorem

The public projects problem with 3 multi-unit-demand agents is NP-hard.

We use a reduction from 3-dimensional matching.

Definition (3-Dimensional Matching)

Input: A set $M \subseteq [q] \times [q] \times [q]$.
Decision Problem: Does there exist some $M' \subseteq M$ such that $|M'| = q$ and no two members of M' agree on any coordinate?

Reduction

The ith agent has value equal to the number of distinct ith coordinates.

Each of the 3 agents has q unit-demand valuations. For each $i, j, k \in M$, create an item that is only valued by agent 1's ith valuation, agent 2's jth valuation and agent 3's kth valuation. Allow for q items to be chosen.

2/3 approximation

Although VCG can't beat a $\sqrt{m}$ approximation for 3 agents, we can use a randomized VCG strategy to get an expected 2/3 approximation.

Theorem

The below mechanism is universally truthful and achieves at least a 2/3 approximation in expectation.

Mechanism

- Choose 2 of the 3 agents uniformly at random
- Solve exactly for these 2 agents using the VCG mechanism

Summary

- Multi-unit-demand is easy for 2 agents, but hard for 3
- Despite the failure of the VCG mechanism for 3 agents, we can randomize for a constant approximation

Outline

1. Background
2. Unit-Demand
3. Multi-Unit-Demand
4. Capped Additive
5. Coverage
6. Summary and Conclusions

Motivating Example

Now that the mall visitors are well-fed, they head to the gym. The gym has room for k pieces of exercise equipment.

- Visitors want to maximize gym hours
- Visitors have differing abilities on each machine
- Visitors have time constraints on their workouts

Definition

Definition (Capped Additive Valuation)

An agent with a capped additive valuation has private values w_j for each item j, and a value limit b, and has value

$$v_i(S) = \min\left(\sum_{j \in S} w_j, b\right)$$

for set S.

NP Hardness

Theorem

The public projects problem with capped-additive agents is NP-hard.

We reduce from subset sum.

NP Hardness

Theorem

The public projects problem with capped-additive agents is NP-hard.

We reduce from subset sum.

Definition (Subset Sum)

Input: A set of positive integers $w_1, \ldots, w_\ell$, and a positive integer t
Decision Problem: Does there exist a set $S \subseteq [\ell]$ such that $\sum_{i \in S} w_i = t$?

NP Hardness

Theorem

The public projects problem with capped-additive agents is NP-hard.

We reduce from subset sum.

Definition (Subset Sum)

Input: A set of positive integers $w_1, \ldots, w_\ell$, and a positive integer t
Decision Problem: Does there exist a set $S \subseteq [\ell]$ such that $\sum_{i \in S} w_i = t$?

Proof.

- Add ℓ integers $w_{\ell+1}, \ldots, w_{2\ell} = 0$. There are $m = 2\ell$ items and $k = \ell$.
- Agent 1 has value $2w_j$ for item j and budget $2t$
- Let $W = \max_j w_j$. Agent 2 has value $W - w_j$ for item j and $b = \infty$
- Social welfare $\ell W + t$ is achievable iff $\exists S, \sum_{i \in S} w_i = t$ □

Pseudo-Poly Algorithm

We can use dynamic programming if the valuations are small.

Theorem

There is a pseudo-polynomial time algorithm for the public projects problem with a constant number of capped-additive agents.

Algorithm (Dynamic Programming)

- Create an $n+2$ dimensional table
- Entry $v_1, \ldots, v_n, i, j$ denotes whether there exists a set $S \subseteq [i], |S| = j$ such that agent ℓ has value v_ℓ.
- The table has poly size if no agent can have superpolynomial value
- If we fill in all the entries for $i-1, j$ and $i-1, j-1$, we can easily fill in all the entries for i, j

Pseudo-Poly Algorithm

We can use dynamic programming if the valuations are small.

Theorem

There is a pseudo-polynomial time algorithm for the public projects problem with a constant number of capped-additive agents.

Algorithm (Dynamic Programming)

- Create an $n+2$ dimensional table
- Entry $v_1, \ldots, v_n, i, j$ denotes whether there exists a set $S \subseteq [i], |S| = j$ such that agent ℓ has value v_ℓ.
- The table has poly size if no agent can have superpolynomial value
- If we fill in all the entries for $i-1, j$ and $i-1, j-1$, we can easily fill in all the entries for i, j

This can be turned into an FPTAS by ignoring the low order bits

Summary

Despite the existence of an FPTAS, no known truthful mechanism beats a $\sqrt{m}$ approximation for 2 capped additive agents.

Outline

Motivating Example

Now that everyone is well fed and in shape, they can get to the serious business of shopping. But with all the food court and gym construction, there's only room for k more shops! Each shopper has a list.

Shopper 1:

- "Juicy" shorts
- Ear piercing
- "Hello Kitty" compact
- Cell phone accessories

Shopper 2:

- Pet toys
- Baby clothes
- Diapers
- Lingerie

Shopper 3:

- Sofa
- Chairs
- Bed
- Curtains
- Television

Shopper 4:

- Cough syrup
- Matchbooks
- Iodine

Definition

Definition (Coverage Valuation)

An agent with a coverage valuation associates a set T_j with each item j, and has value

$$v_i(S) = \left| \bigcup_{j \in S} T_j \right|$$

for set S.

NP hardness

Theorem

The public projects problem with a single coverage valuation agent is NP-hard.

Definition (max-k-cover)

Input: Several sets $T_1, \ldots, T_m$

Goal: Find a set $S \subseteq [m], |S| = k$ maximizing $|\bigcup_{j \in S} T_j|$

NP hardness

Theorem

The public projects problem with a single coverage valuation agent is NP-hard.

Definition (max-k-cover)

Input: Several sets $T_1, \ldots, T_m$
Goal: Find a set $S \subseteq [m], |S| = k$ maximizing $|\bigcup_{j \in S} T_j|$

Definition (Public Project with 1 Coverage Valuation Agent)

Input:
Goal:

NP hardness

Theorem

The public projects problem with a single coverage valuation agent is NP-hard.

Definition (max-k-cover)

Input: Several sets $T_1, \ldots, T_m$
Goal: Find a set $S \subseteq [m], |S| = k$ maximizing $|\bigcup_{j \in S} T_j|$

Definition (Public Project with 1 Coverage Valuation Agent)

Input: Several sets $T_1, \ldots, T_m$
Goal:

NP hardness

Theorem

The public projects problem with a single coverage valuation agent is NP-hard.

Definition (max-k-cover)

Input: Several sets $T_1, \ldots, T_m$
Goal: Find a set $S \subseteq [m], |S| = k$ maximizing $|\bigcup_{j \in S} T_j|$

Definition (Public Project with 1 Coverage Valuation Agent)

Input: Several sets $T_1, \ldots, T_m$
Goal: Find a set $S \subseteq [m], |S| = k$ maximizing $|\bigcup_{j \in S} T_j|$

Wait a second...

Theorem

No truthful poly-time mechanism for public projects can achieve better than a $\sqrt{m}$ approximation unless $NP \subseteq P/poly$.

Proof.

- Our results show hardness for VCG to do better than $\sqrt{m}$
- For a single agent, any mechanism must be maximal-in-range to be truthful, so VCG is all there is

□

Wait a second...

Theorem

No truthful poly-time mechanism for public projects can achieve better than a $\sqrt{m}$ approximation unless $NP \subseteq P/poly$.

Proof.

- Our results show hardness for VCG to do better than $\sqrt{m}$
- For a single agent, any mechanism must be maximal-in-range to be truthful, so VCG is all there is

□

As this is submodular, we can get a $1 - 1/e$ greedy approximation.

Both the agent and the mechanism want the social welfare maximized, but if we want truthfulness, they can't even do as well as the greedy algorithm.

Summary

No truthful mechanism gets a good approximation despite the lack of conflicting goals.

Outline

Summary of Results

VCG can't get any constant approximation for

- n unit-demand agents
- 3 multi-unit-demand agents
- 2 capped additive agents
- 1 coverage agent

But

- There is a constant approximation for a special case of unit-demand
- A solution for 2 multi-unit-demand agents can be used to get a 2/3 approximation for 3
- There is an FPTAS for capped additive agents
- All truthful mechanisms are VCG-based for coverage valuation agents

Conclusions and Open Problems

- For public projects, we have to look beyond the VCG mechanism
  Can we develop better truthful mechanisms than VCG?
- In some situations, truthfulness is inherently flawed
  What should we use when truthfulness doesn't fit?