Combinatorial allocation and submodular maximization over a matroid

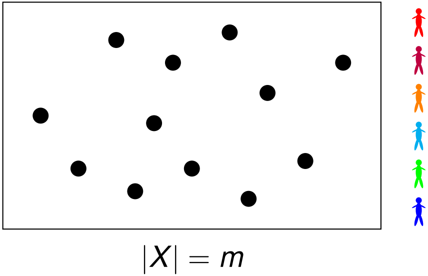
Jan Vondrák, Princeton University

RIMS 2008

with G. Calinescu, C. Chekuri and M. Pál

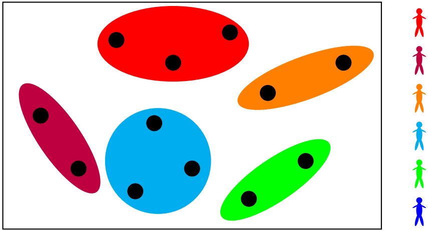
Combinatorial allocation problems

Given: $|X| = m$ items, n players with *utility functions* $w_i : 2^X \to \mathbb{R}_+$.
Goal: Find an allocation of disjoint sets $S_1, \ldots, S_n$ to the n players.



Combinatorial allocation problems

Given: $|X| = m$ items, n players with *utility functions* $w_i : 2^X \to \mathbb{R}_+$.
Goal: Find an allocation of disjoint sets $S_1, \ldots, S_n$ to the n players.



$|X| = m$

Allocation $(S_1, S_2, \ldots, S_n)$

has value $\sum_{i=1}^{n} w_i(S_i)$.

We want to maximize the *social welfare* $\sum_{i=1}^{n} w_i(S_i)$.

Motivation

Combinatorial auctions:
multiple items/resources are sold to agents with different preferences.

- *Google AdWords:* advertising space associated with web search.
- *FCC spectrum auctions:* frequencies for wireless communication.

Motivation

Combinatorial auctions:
multiple items/resources are sold to agents with different preferences.

- *Google AdWords:* advertising space associated with web search.
- *FCC spectrum auctions:* frequencies for wireless communication.

Here: we consider only the optimization problem: knowing the utility functions, how do we allocate optimally?

Computational issue: in general, the welfare maximization problem is at least as hard as Set Packing. I.e., $n^{1-\epsilon}$ or $m^{1/2-\epsilon}$ approximation is NP-hard.

Submodular utility functions

Positive results: have been obtained only under certain assumptions on the utility functions.

Submodular utility functions

Positive results: have been obtained only under certain assumptions on the utility functions. In particular:

utility functions which are ***monotone and submodular***.

Submodular utility functions

Positive results: have been obtained only under certain assumptions on the utility functions. In particular:

utility functions which are ***monotone and submodular***.

- Monotonicity: $w_i(S) \leq w_i(T)$ for $S \subset T$.
- Submodularity is the discrete analogue of concavity; in economics, known as *diminishing returns*.
- Monotone submodular functions appear naturally in many combinatorial settings: rank functions of matroids, coverage systems, etc.

Submodularity

Definition

A function $f : 2^X \to \mathbb{R}$ is submodular if for any S, T,

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T).$$

Submodularity

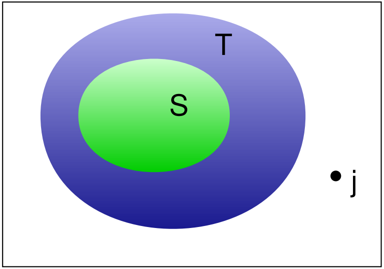
Definition

A function $f : 2^X \to \mathbb{R}$ is submodular if for any S, T,

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T).$$

Alternative definition: Define the *marginal value of element j*,

$$f_S(j) = f(S \cup \{j\}) - f(S).$$



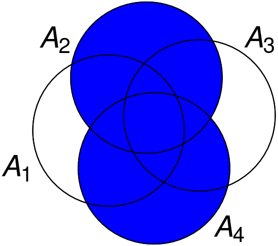
f is submodular, if $\forall S \subset T, j \notin T$:

$$f_S(j) \geq f_T(j).$$

Examples of Submodular Functions

Coverage functions:
Given $A_1, \ldots, A_n \subset U$,

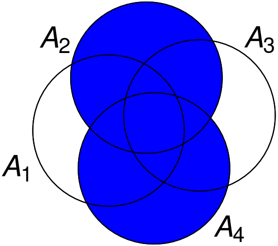
$$f(S) = \left| \bigcup_{j \in S} A_j \right|.$$

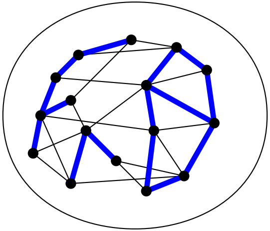


Examples of Submodular Functions

Coverage functions:
Given $A_1, \ldots, A_n \subset U$,

$$f(S) = \left| \bigcup_{j \in S} A_j \right|.$$





Connectivity in graphs:

$r(T) = n-$ number of connected components in T.

Accessing the utility functions

Issue of representation:
A submodular function $w_i : 2^X \rightarrow \mathbb{R}_+$ carries too much information - in general we cannot afford to write it down explicitly.

Accessing the utility functions

Issue of representation:
A submodular function $w_i : 2^X \to \mathbb{R}_+$ carries too much information - in general we cannot afford to write it down explicitly.

Oracle models:
An algorithm can query the players about their utility functions.

1. *Value query:* What is the value of $w_i(S)$?
2. *Demand query:* For given $p_1, \ldots, p_m \in \mathbb{R}$, which set maximizes $w_i(S) - \sum_{j \in S} p_j$?

Accessing the utility functions

Issue of representation:

A submodular function $w_i : 2^X \rightarrow \mathbb{R}_+$ carries too much information - in general we cannot afford to write it down explicitly.

Oracle models:

An algorithm can query the players about their utility functions.

1. *Value query:* What is the value of $w_i(S)$?
2. *Demand query:* For given $p_1, \ldots, p_m \in \mathbb{R}$, which set maximizes $w_i(S) - \sum_{j \in S} p_j$?

Note: in general, demand queries are NP-hard to answer, even for submodular functions.

History

Submodular Welfare Problem:
Given n players with submodular utility functions $w_i : 2^X \to \mathbb{R}_+$*.*
Allocate the items so as to maximize $\sum_{i=1}^{n} w_i(S_i)$*.*

History

Submodular Welfare Problem:
Given n players with submodular utility functions $w_i : 2^X \rightarrow \mathbb{R}_+$*.*
Allocate the items so as to maximize $\sum_{i=1}^n w_i(S_i)$*.*

Known results:

- Greedy $1/2$-approximation. [Lehmann,Lehmann,Nisan '01]
- No $(1 - 1/e + \epsilon)$-approximation in the *value oracle model* unless $P = NP$. [Khot, Lipton, Markakis, Mehta '05]
- $(1 - 1/e)$-approximation in the *demand oracle model*. [Dobzinski, Schapira '06]
- Can be improved to $1 - 1/e + \epsilon$. [Feige,V. '06]

History

Submodular Welfare Problem:
Given n players with submodular utility functions $w_i : 2^X \to \mathbb{R}_+$*.*
Allocate the items so as to maximize $\sum_{i=1}^{n} w_i(S_i)$*.*

Known results:

- Greedy $1/2$-approximation. [Lehmann,Lehmann,Nisan '01]
- No $(1 - 1/e + \epsilon)$-approximation in the *value oracle model* unless $P = NP$. [Khot, Lipton, Markakis, Mehta '05]
- $(1 - 1/e)$-approximation in the *demand oracle model*. [Dobzinski, Schapira '06]
- Can be improved to $1 - 1/e + \epsilon$. [Feige,V. '06]

A more general problem:
submodular maximization subject to a matroid constraint.

Matroids

Definition

A matroid on N is a system of *independent sets* $\mathcal{M} \subset 2^N$, satisfying

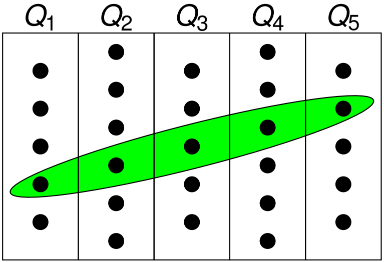
1. $\forall B \in \mathcal{M}, A \subset B \;\Rightarrow\; A \in \mathcal{M}$.
2. $\forall A, B \in \mathcal{M}, |A| < |B| \;\Rightarrow\; \exists x \in B \setminus A;\; A \cup \{x\} \in \mathcal{M}$.

Matroids

Definition

A matroid on N is a system of *independent sets* $\mathcal{M} \subset 2^N$, satisfying

1. $\forall B \in \mathcal{M}, A \subset B \;\Rightarrow\; A \in \mathcal{M}$.
2. $\forall A, B \in \mathcal{M}, |A| < |B| \;\Rightarrow\; \exists x \in B \setminus A;\; A \cup \{x\} \in \mathcal{M}$.



Example: *partition matroid*

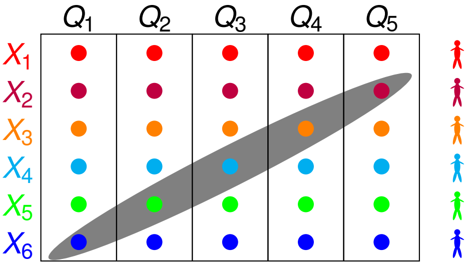
S is independent, if $|S \cap Q_i| \leq 1$ for each Q_i.

Submodular Welfare → matroid constraint

Submodular Welfare: *Given n players with submodular utility functions* $w_i : 2^X \rightarrow \mathbb{R}_+$*. Allocate the items so as to maximize* $\sum_{i=1}^{n} w_i(S_i)$*.*

Submodular Welfare → matroid constraint

Submodular Welfare: *Given n players with submodular utility functions* $w_i : 2^X \to \mathbb{R}_+$*. Allocate the items so as to maximize* $\sum_{i=1}^{n} w_i(S_i)$.



Reduction:

Create n clones of each item,
$f(S) = \sum w_i(S \cap X_i)$,
$\mathcal{M} = \{S : \forall i; |S \cap Q_i| \leq 1\}$
(a partition matroid).

The Submodular Welfare Problem is equivalent to $\max\{f(S) : S \in \mathcal{M}\}$.

Submodular Maximization under a Matroid Constraint

Problem. *Given a monotone submodular function f (via a* value oracle*) and a matroid $\mathcal{M} = (N, \mathcal{I})$ (via a* membership oracle*), find (or approximate)*

$$\max_{S \in \mathcal{I}} f(S).$$

Submodular Maximization under a Matroid Constraint

Problem. *Given a monotone submodular function f (via a* value oracle*) and a matroid $\mathcal{M} = (N, \mathcal{I})$ (via a* membership oracle*), find (or approximate)*

$$\max_{S \in \mathcal{I}} f(S).$$

Known:

- A greedy $1/2$-approximation [Fisher, Nemhauser, Wolsey '78].
- $(1 - 1/e)$-approximation when $\mathcal{M} = \{S : |S| \leq k\}$ and this is optimal with poly-many value queries [Nemhauser, Wolsey '78].
- The Max k-cover problem, $\max_{|S| \leq k} |\bigcup_{j \in S} A_j|$, is a special case, and $1 - 1/e$ is optimal unless $P = NP$ [Feige '98].

Our Results

Theorem (V. '07)

There is a randomized $(1 - 1/e)$*-approximation for the Submodular Welfare Problem, in the value oracle model.*

Our Results

Theorem (V. ’07)

There is a randomized $(1 - 1/e)$*-approximation for the Submodular Welfare Problem, in the value oracle model.*

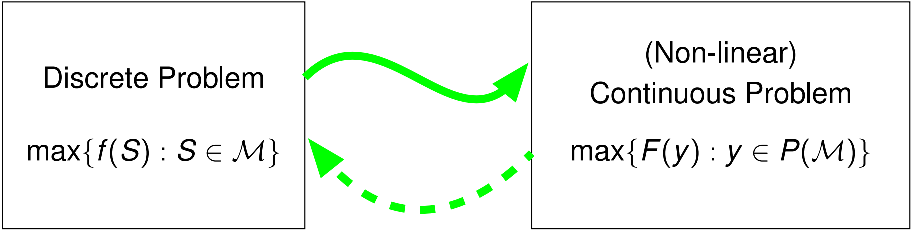
Theorem (Calinescu,Chekuri,Pál,V. ’07)

There is a randomized $(1 - 1/e)$*-approximation algorithm for*

$$\max\{f(S) : S \in \mathcal{M}\},$$

for any monotone submodular $f : 2^X \to \mathbb{R}_+$ *and a matroid* $\mathcal{M} \subset 2^X$*.*

General approach



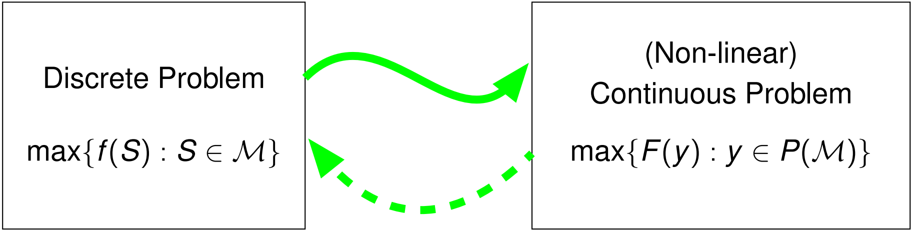
Discrete Problem

$\max\{f(S) : S \in \mathcal{M}\}$

(Non-linear)
Continuous Problem

$\max\{F(y) : y \in P(\mathcal{M})\}$

General approach



Discrete Problem

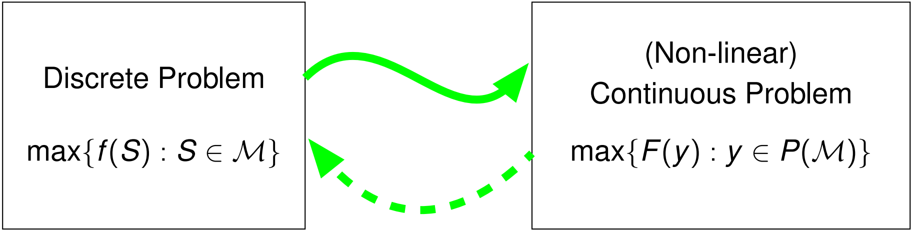
$\max\{f(S) : S \in \mathcal{M}\}$

(Non-linear)
Continuous Problem

$\max\{F(y) : y \in P(\mathcal{M})\}$

- $P(\mathcal{M}) = \mathrm{conv}\{\mathbf{1}_S : S \in \mathcal{M}\} \subset [0,1]^X$.

General approach



Discrete Problem

$\max\{f(S) : S \in \mathcal{M}\}$

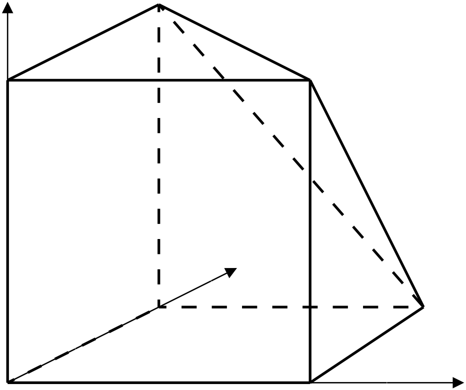
(Non-linear)
Continuous Problem

$\max\{F(y) : y \in P(\mathcal{M})\}$

- $P(\mathcal{M}) = \mathrm{conv}\{\mathbf{1}_S : S \in \mathcal{M}\} \subset [0,1]^X$.
- $F(y) = \mathbb{E}[f(\hat{y})]$, where $\hat{y}$ is obtained by rounding each y_j randomly to $0/1$ with probabilities $1 - y_j$ and y_j.

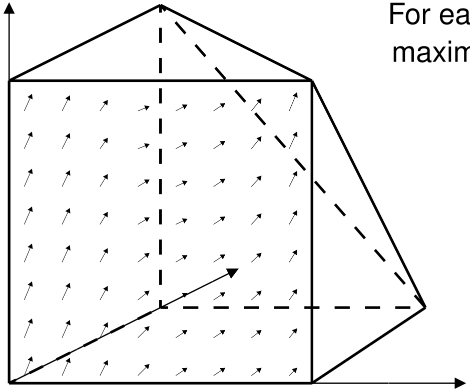
Solving the continous problem

Problem: $\max\{F(y) : y \in P(\mathcal{M})\}$.



Solving the continous problem

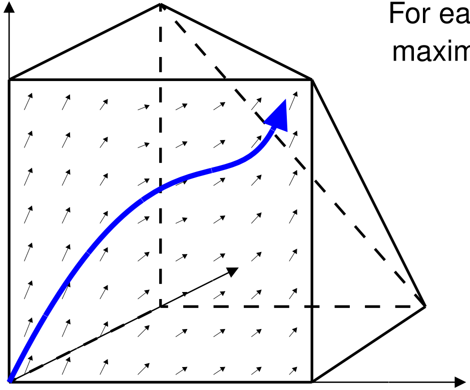
Problem: $\max\{F(y) : y \in P(\mathcal{M})\}$.



For each $y \in P(\mathcal{M})$, define $v(y)$ by maximizing $v \cdot \nabla F$ over $v \in P(\mathcal{M})$.

Solving the continous problem

Problem: $\max\{F(y) : y \in P(\mathcal{M})\}$.



For each $y \in P(\mathcal{M})$, define $v(y)$ by maximizing $v \cdot \nabla F$ over $v \in P(\mathcal{M})$.

Define a curve $y(t)$:

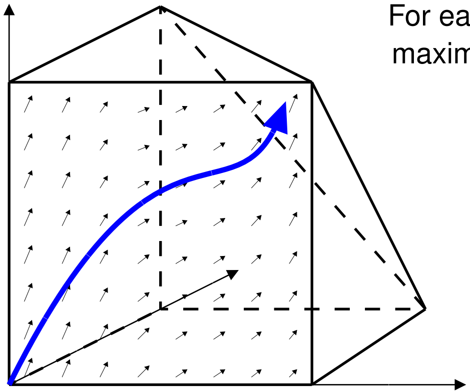
$$y(0) = 0$$

$$\frac{dy}{dt} = v(y)$$

Run this process for $t \in [0, 1]$ and return $y(1)$.

Solving the continous problem

Problem: $\max\{F(y) : y \in P(\mathcal{M})\}$.



For each $y \in P(\mathcal{M})$, define $v(y)$ by maximizing $v \cdot \nabla F$ over $v \in P(\mathcal{M})$.

Define a curve $y(t)$:

$$y(0) = 0$$

$$\frac{dy}{dt} = v(y)$$

Run this process for $t \in [0, 1]$ and return $y(1)$.

Claim: $y(1) \in P(\mathcal{M})$ and $F(y(1)) \geq (1 - 1/e)OPT$.

Analysis I

Lemma

For any point $y \in P$*, there is a direction* $v \in P(\mathcal{M})$ *such that*

$$v \cdot \nabla F(y) \geq OPT - F(y).$$

Analysis I

Lemma

For any point $y \in P$, there is a direction $v \in P(\mathcal{M})$ such that

$$v \cdot \nabla F(y) \geq OPT - F(y).$$

Proof sketch:

- $\frac{\partial F}{\partial y_j} \geq 0$ (monotonicity), $\frac{\partial^2 F}{\partial y_i \partial y_j} \leq 0$ (submodularity).
- Therefore, F is increasing and concave along any direction $v \geq 0$.
- For any y, take $v = (z - y)_+$ where $OPT = F(z)$:

$$v \cdot \nabla F(y) \geq F(y + v) - F(y) \geq OPT - F(y).$$

Analysis II

We have:

- $y(0) = 0, \ \frac{dy}{dt} = v(y)$.
- $v(y) \cdot \nabla F = \max_{v \in P(\mathcal{M})} v \cdot \nabla F \geq OPT - F(y)$ (by the last lemma).

Analysis II

We have:

- $y(0) = 0, \ \frac{dy}{dt} = v(y)$.
- $v(y) \cdot \nabla F = \max_{v \in P(\mathcal{M})} v \cdot \nabla F \geq OPT - F(y)$ (by the last lemma).

The chain rule:

$$\frac{dF}{dt} = \sum_j \frac{\partial F}{\partial y_j} \frac{dy_j}{dt} = v(y(t)) \cdot \nabla F \geq OPT - F(y(t)).$$

Analysis II

We have:

- $y(0) = 0,\ \frac{dy}{dt} = v(y)$.
- $v(y) \cdot \nabla F = \max_{v \in P(\mathcal{M})} v \cdot \nabla F \geq OPT - F(y)$ (by the last lemma).

The chain rule:

$$\frac{dF}{dt} = \sum_j \frac{\partial F}{\partial y_j} \frac{dy_j}{dt} = v(y(t)) \cdot \nabla F \geq OPT - F(y(t)).$$

Solve the differential equation:

$$F(y(t)) \geq (1 - e^{-t}) \cdot OPT.$$

Technical points

1. To compute $F(y(t))$ for $t \in [0,1]$, we discretize the process and run a loop for $t = 0, \frac{1}{n^2}, \frac{2}{n^2}, \ldots, 1$.
2. We need to estimate

$$(\nabla F)_j = \frac{\partial F}{\partial y_j} = \mathbb{E}[f(\hat{y}) \mid y_j = 1] - \mathbb{E}[f(\hat{y}) \mid y_j = 0];$$

   this can be done within $\pm OPT/poly(n)$ by repeated sampling.
3. We need to optimize $\max\{v \cdot \nabla F : v \in P(\mathcal{M})\}$; this is easy (greedy algorithm for max-weight independent set in a matroid).

Going back to a discrete solution

Submodular Welfare Problem:

- Interpretation of the fractional solution:
  randomized assignment $(R_1, R_2, \ldots, R_n)$, $\Pr[j \in R_i] = y_{ij}$,

$$F(y) = \sum_{i=1}^{n} \mathbb{E}[w_i(R_i)].$$

- Polytope constraints: for each item j,

$$\sum_{i=1}^{n} y_{ij} \leq 1.$$

Going back to a discrete solution

Submodular Welfare Problem:

- Interpretation of the fractional solution: *randomized assignment* $(R_1, R_2, \dots, R_n)$, $\Pr[j \in R_i] = y_{ij}$,

$$F(y) = \sum_{i=1}^{n} \mathbb{E}[w_i(R_i)].$$

- Polytope constraints: for each item j,

$$\sum_{i=1}^{n} y_{ij} \le 1.$$

Solution:

- Allocate each item j independently, with probability y_{ij} to player i.
- The expected value of our solution is $F(y) \geq (1 - 1/e) OPT$.

Pipage rounding for a matroid polytope

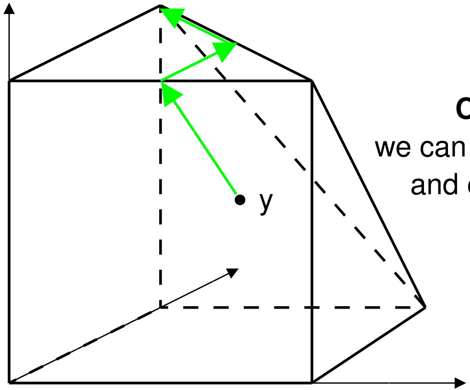
Lemma (Calinescu,Chekuri,Pál,V. '07)

There is a randomized procedure which, given $\mathcal{M}$ *and* $y \in P(\mathcal{M})$, *finds a solution* $S \in \mathcal{M}$ *of value* $\mathbb{E}[f(S)] \geq F(y)$.

Pipage rounding for a matroid polytope

Lemma (Calinescu,Chekuri,Pál,V. '07)

There is a randomized procedure which, given $\mathcal{M}$ and $y \in P(\mathcal{M})$, finds a solution $S \in \mathcal{M}$ of value $\mathbb{E}[f(S)] \geq F(y)$.



Claim: In each step, $\exists i, j \in X$, we can move along $\mathbf{e}_i - \mathbf{e}_j$ or $\mathbf{e}_j - \mathbf{e}_i$ and eventually we reach a vertex.

The choice between $\mathbf{e}_i - \mathbf{e}_j$ and $\mathbf{e}_j - \mathbf{e}_i$ is random.

Convexity property

Lemma

Let $f : 2^X \to \mathbb{R}_+$ *be submodular,* $F(y) = \mathbb{E}[f(\hat{y})]$ *and* $i, j \in X$*. Then*

$$\phi(t) = F(y + t(\mathbf{e}_i - \mathbf{e}_j))$$

is a convex function.

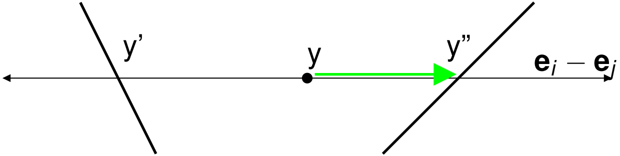
Convexity property

Lemma

Let $f : 2^X \to \mathbb{R}_+$ *be submodular,* $F(y) = \mathbb{E}[f(\hat{y})]$ *and* $i, j \in X$*. Then*

$$\phi(t) = F(y + t(\mathbf{e}_i - \mathbf{e}_j))$$

is a convex function.



A suitable convex combination of $F(y')$ and $F(y'')$ is at least $F(y)$
... we go randomly to y' or y'' & iterate.

Other applications

Algorithms for other allocation problems:

- *Generalized Assignment Problem:* simple $(1 - 1/e - o(1))$ approximation
  (GAP admits a $(1 - 1/e + \epsilon)$-approximation [Feige,V.'06])
- *Adwords Rectangle Packing Problem:* $(1 - 1/\sqrt{e} - o(1))$ approximation [previously known: $1/3$-approximation, GS'07].

Other applications

Algorithms for other allocation problems:

- *Generalized Assignment Problem:* simple $(1 - 1/e - o(1))$ approximation
  (GAP admits a $(1 - 1/e + \epsilon)$-approximation [Feige,V.'06])
- *Adwords Rectangle Packing Problem:* $(1 - 1/\sqrt{e} - o(1))$ approximation [previously known: $1/3$-approximation, GS'07].

Technique: assume we can optimize linear functions over $P(\mathcal{M})$ approximately. For each y, find $v(y) \in P(\mathcal{M})$, so that

$$v(y) \cdot \nabla F \geq \alpha \max_{v \in P(\mathcal{M})} v \cdot \nabla F \geq \alpha(OPT - F(y)).$$

$$\frac{dF}{dt} = \sum_j \frac{\partial F}{\partial y_j} \frac{dy_j}{dt} = v(y(t)) \cdot \nabla F \geq \alpha(OPT - F(y(t))).$$

$$F(y(1)) \geq (1 - e^{-\alpha}) \cdot OPT.$$

Unconditional hardness in the value oracle model

Theorem (Mirrokni,Schapira,V. '07)

An approximation better than $1 - 1/e$ *for the Submodular Welfare Problem would require exponentially many value queries.*

Unconditional hardness in the value oracle model

Theorem (Mirrokni,Schapira,V. '07)

An approximation better than $1 - 1/e$ *for the Submodular Welfare Problem would require exponentially many value queries.*

Theorem (V. '07)

For players with equal *submodular utility functions, a* random allocation *gives a* $(1 - 1/e)$*-approximation. (Even this is optimal, unless we use exponentially many value queries.)*

Unconditional hardness in the value oracle model

Theorem (Mirrokni,Schapira,V. '07)

An approximation better than $1 - 1/e$ *for the Submodular Welfare Problem would require exponentially many value queries.*

Theorem (V. '07)

For players with equal *submodular utility functions, a* random allocation *gives a* $(1 - 1/e)$*-approximation. (Even this is optimal, unless we use exponentially many value queries.)*

Theorem (V. '08)

An approximation better than $1 - e^{-\alpha}$*, using an* α*-approximate marginal value oracle, would require exponentially many queries.*

Note: These results are *independent of* $P = NP$.

Value query hardness: proof

Idea: use continuous functions ϕ, $\frac{\partial \phi}{\partial y_i} \geq 0$, $\frac{\partial^2 \phi}{\partial y_i \partial y_j} \leq 0$, to construct discrete submodular functions on a “hidden partition" $(X_1, \dots, X_n)$:

$$f(S) = \phi\left(\frac{|S \cap X_1|}{|X_1|}, \dots, \frac{|S \cap X_n|}{|X_n|}\right).$$

Analytic properties of ϕ ensure that instances differing by a certain gap are indistinguishable.

Value query hardness: proof

Idea: use continuous functions ϕ, $\frac{\partial \phi}{\partial y_i} \geq 0$, $\frac{\partial^2 \phi}{\partial y_i \partial y_j} \leq 0$, to construct discrete submodular functions on a "hidden partition" $(X_1, \ldots, X_n)$:

$$f(S) = \phi\left(\frac{|S \cap X_1|}{|X_1|}, \ldots, \frac{|S \cap X_n|}{|X_n|}\right).$$

Analytic properties of ϕ ensure that instances differing by a certain gap are indistinguishable. We choose:

$\phi(x_1, \ldots, x_n) = 1 - (1 - x_1)(1 - x_2) \cdots (1 - x_n)$,
$\psi(x_1, \ldots, x_n) = 1 - (1 - \frac{1}{n} \sum_{i=1}^{n} x_i)^n$.

- For a typical query, $x_1 = \ldots = x_n$.
- $\phi = \psi$ and $\nabla \phi = \nabla \psi$ for $x_1 = x_2 = \ldots = x_n$.
- The optima differ by a factor of $1 - 1/e$.

Future directions

Submodular optimization:

- What is the optimal approximation for Submodular Welfare in the demand oracle model?
- On-line Submodular Welfare: $(1/2 + o(1))$-approximation is still the best known [Dobzinski, Schapira '06].
- Submodular maximization subject to more general constraints:

  $$\max\{f(S) : S \in \mathcal{F}\}.$$

  E.g., $1/(k+1)$-approximation is known when $\mathcal{F}$ is the intersection of k matroids - what is the optimal factor?
- Hardness results seem to be cleaner in the value oracle model. Optimal approximation results for submodular maximization subject to more general constraints?

Combinatorial auctions

Incentive-compatible mechanism design:
Assuming that players can lie about their utility functions, we want to design mechanisms that motivate players to answer *truthfully*.

- The best known mechanisms in this setting achieve logarithmic approximation.
- Truthful $O(\log m \log\log m)$-approximation for subadditive utilities [Dobzinski '07].

Combinatorial auctions

Incentive-compatible mechanism design:
Assuming that players can lie about their utility functions, we want to design mechanisms that motivate players to answer *truthfully*.

- The best known mechanisms in this setting achieve logarithmic approximation.
- Truthful $O(\log m \log\log m)$-approximation for subadditive utilities [Dobzinski '07].

Major open question: Is there a truthful constant-factor approximation mechanism for submodular utilities?

Summary of results

Problem	Approximation	Hardness
Submodular Welfare (value queries)	$1 - 1/e$	$1 - 1/e + o(1)$
Submodular Welfare (demand queries)	$1 - 1/e + 10^{-5}$	$19/20$
Submodular Max over a matroid (value queries)	$1 - 1/e$	$1 - 1/e + o(1)$
Submodular Max over a matroid (α-approx. oracle)	$1 - e^{-\alpha}$	$1 - e^{-\alpha} + o(1)$