From Minimum Cut to Submodular Minimization
Leveraging the Decomposable Structure

Alina Ene
Boston University

Joint work with
Huy Nguyen (Northeastern University)
Laszlo Vegh (London School of Economics)

BOSTON UNIVERSITY

Northeastern University
College of Computer and Information Science

LSE
THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Minimum s-t Cut

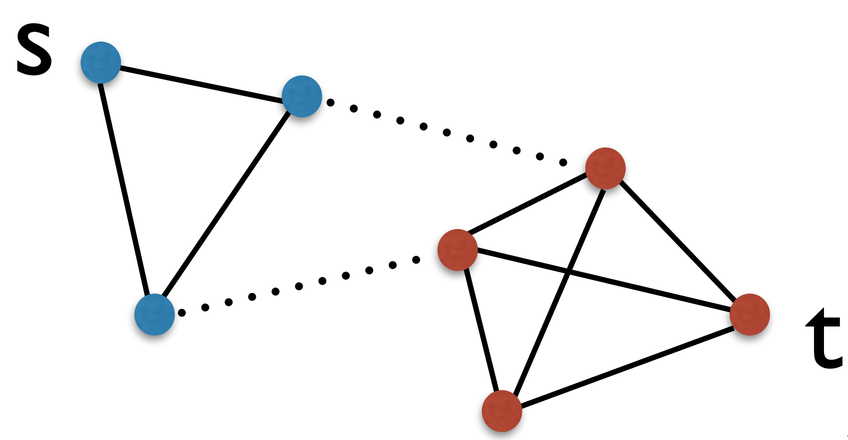
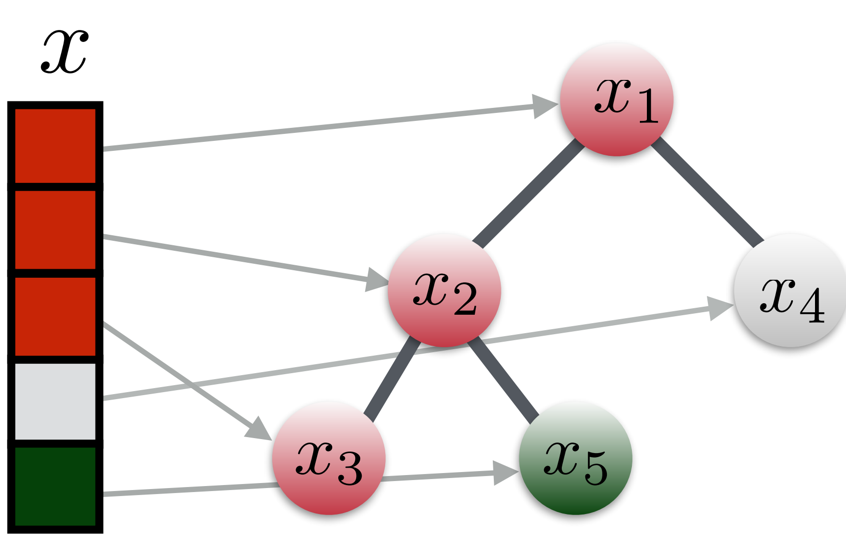


Image Segmentation

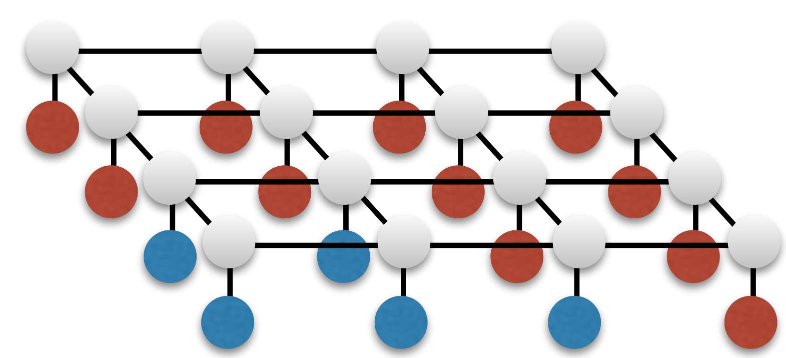
Submodular Minimization

$$\min_{S \subseteq V} f(S)$$



Structured Sparsity

[Bach '10]



MAP Inference

Minimum s-t Cut

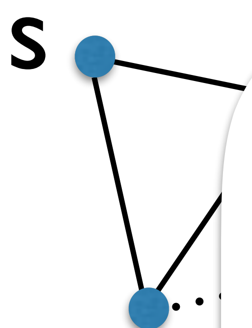
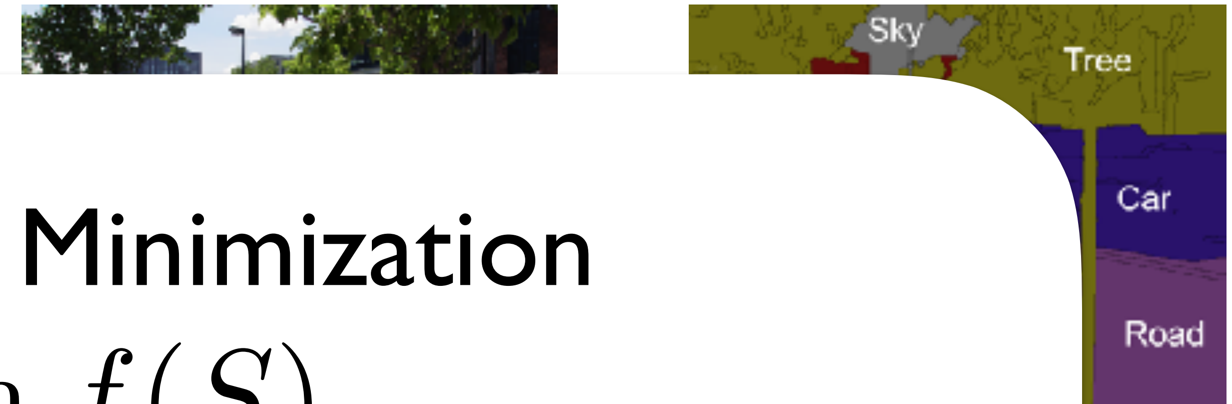
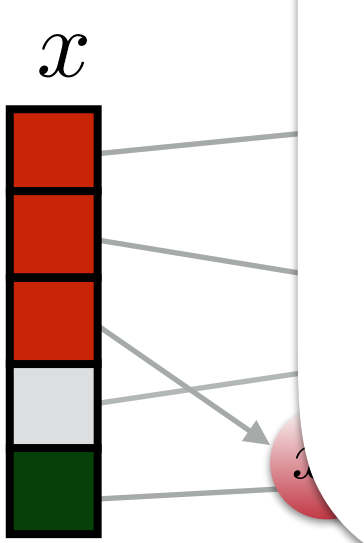


Image Segmentation

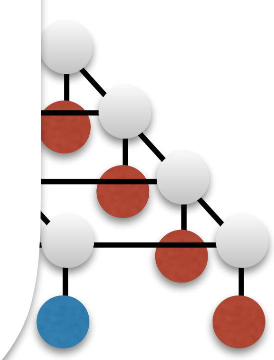


Submodular Minimization

$$\min_{S \subseteq V} f(S)$$

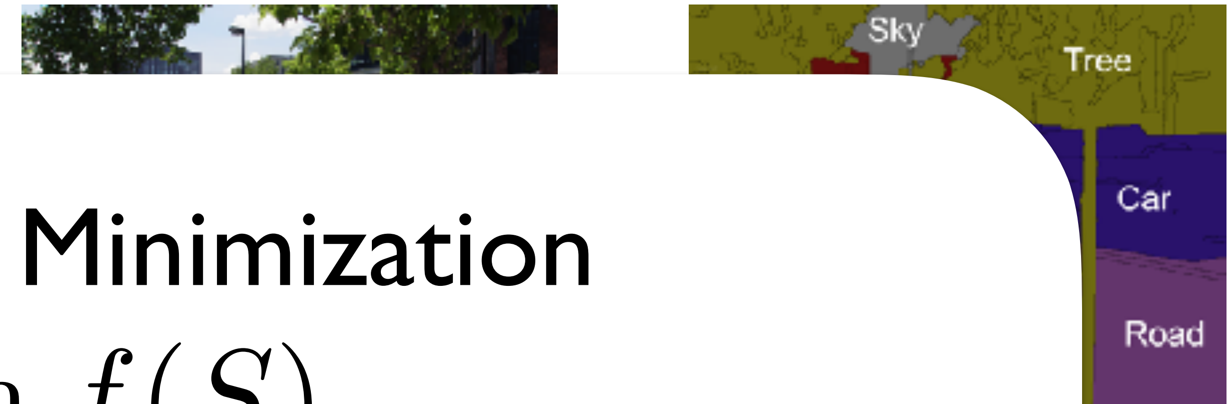


Structured Sparsity
[Bach '10]

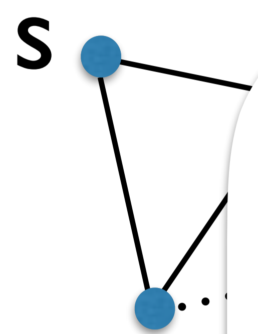


MAP Inference

Image Segmentation



Minimum s-t Cut

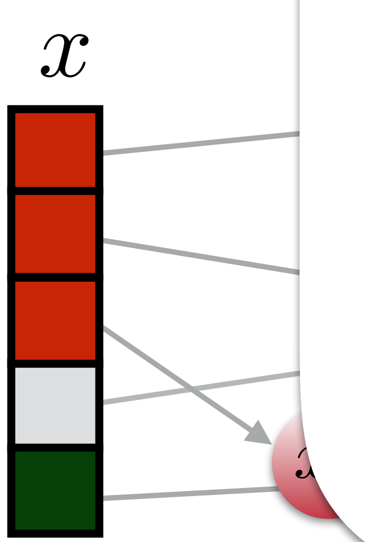


Submodular Minimization

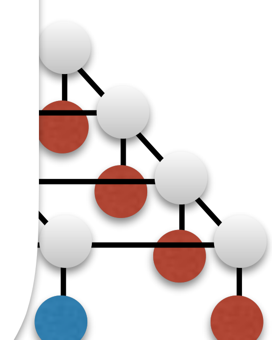
$$\min_{S \subseteq V} f(S)$$

Solvable in polynomial time

Combinatorial, ellipsoid, cutting plane,…



Structured Sparsity
[Bach '10]



MAP Inference

Minimum s-t Cut

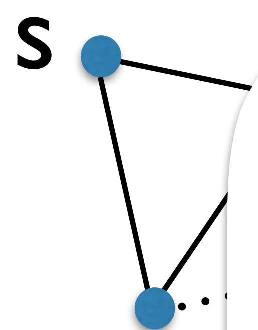
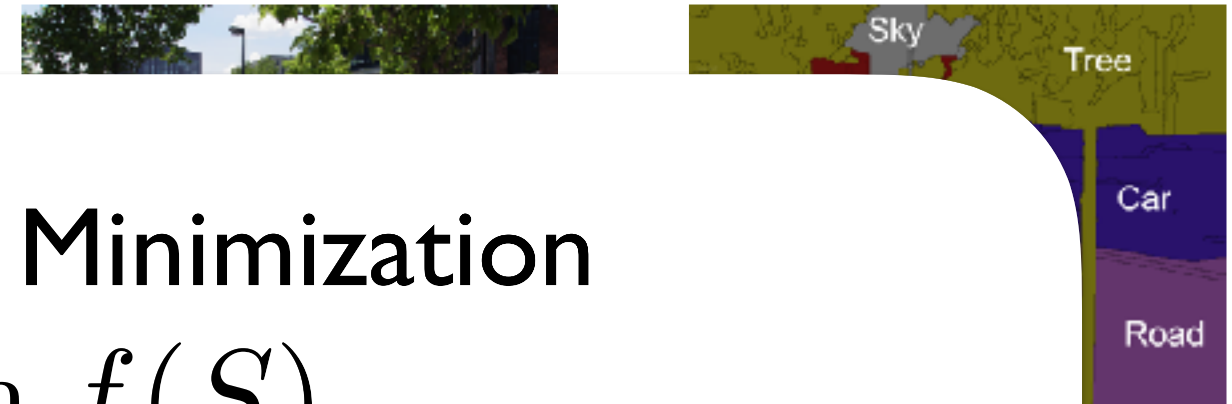


Image Segmentation



Submodular Minimization

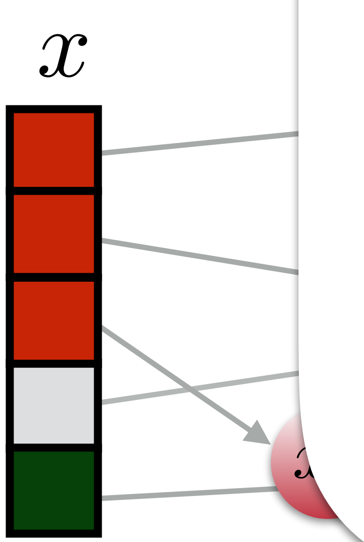
$$\min_{S \subseteq V} f(S)$$

Solvable in polynomial time

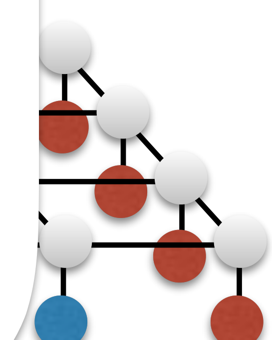
Combinatorial, ellipsoid, cutting plane,…

Very high running times $\Omega(|V|^3)$

Leverage structure to obtain faster algos



Structured Sparsity
[Bach '10]



MAP Inference

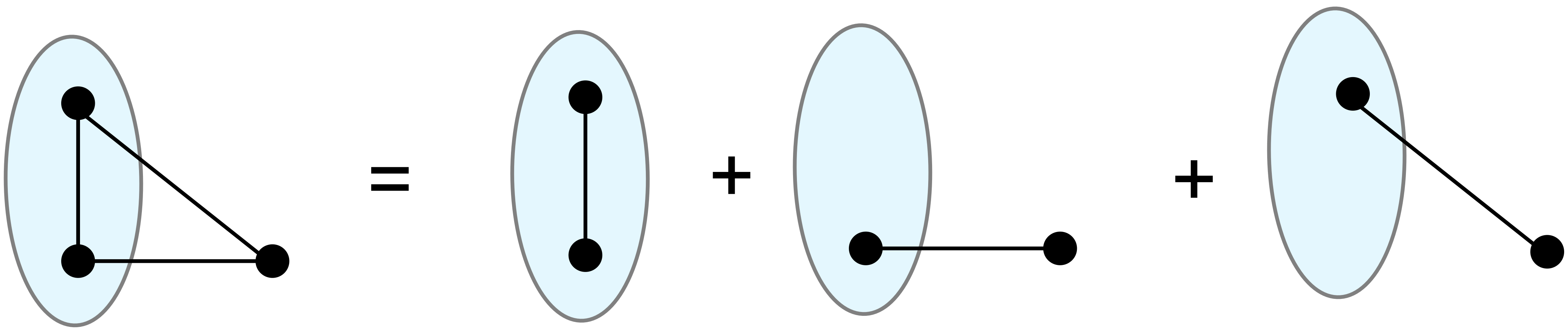
Decomposable Functions

$$\min_{S \subseteq V} \sum_{i=1}^{r} f_i(S)$$

Simple f_i : fast subroutine for minimizing $f_i(S) + w(S)$ for any linear function w

[Stobbe, Krause '10; Kolmogorov '12; Jegelka, Bach, Sra '13; Nishihara, Jegelka, Jordan '14]

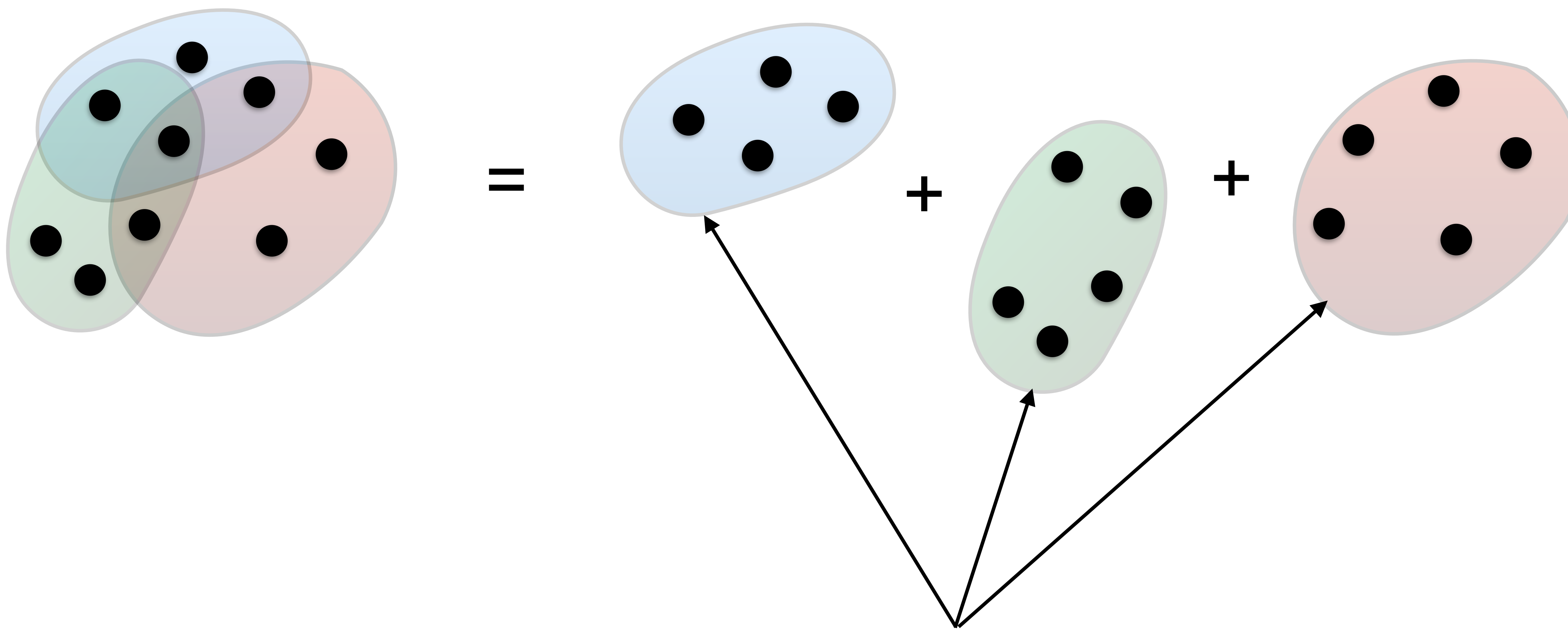
Decomposable Structure



$$f(S) = \sum_{e \in E} f_e(S)$$

Cut function of a single edge

Decomposable Structure



hyperedges, local functions, …

This Talk

Sum-of-Submodular / Decomposable SFM

$$\min_{S \subseteq V} \sum_{i=1}^{r} f_i(S)$$

Continuous algos based on gradient descent

Combinatorial algos based on max flow approaches

Interplay between continuous optimization
and combinatorial structure

Main Combinatorial Player

Key concept underpinning submodular algorithms

Base Polytope

$$B(f) = \{y \mid \langle y, \mathbf{1}_S \rangle \leq f(S) \;\; \forall S \subseteq V, \; \langle y, \mathbf{1}_V \rangle = f(V)\}$$

Think of y as a linear function on V

Continuous Algorithms

From Submodular to Convex Optimization

[Edmonds '70, Lovasz '83, Fujishige '84]
Submodular Minimization can be reduced to finding a point of minimum norm in the base polytope

Discrete Problem

$$\min_{S \subseteq V} f(S)$$

→

Exact Convex Formulation

$$\min_{y \in B(f)} \|y\|^2$$

Discrete Problem

$$\min_{S \subseteq V} f(S)$$

Exact Convex Formulation

$$\min_{y \in B(f)} \|y\|^2$$

Convex objective is very nice

Constraint is very complicated

Discrete Problem

$$\min_{S \subseteq V} f(S)$$

→

Exact Convex Formulation

$$\min_{y \in B(f)} \|y\|^2$$

Convex objective is very nice

Constraint is very complicated

Algorithms based on
Ellipsoid/Cutting Plane [GLS '84, LSW '15]

Provable polynomial running times

Running time prohibitively large in applications

Discrete Problem

$$\min_{S \subseteq V} f(S)$$

→

Exact Convex Formulation

$$\min_{y \in B(f)} \|y\|^2$$

Convex objective is very nice

Constraint is very complicated

Fujishige-Wolfe and Frank-Wolfe algorithms

Suitable for some applications

Pseudo-polynomial running times

Known to require large number of iterations

In the decomposable setting, can leverage more of the convex optimization toolkit

$$\textbf{Lemma:}\ \text{If } f = \sum_{i=1}^{r} f_i \text{ then } B(f) = \sum_{i=1}^{r} B(f_i)$$

Discrete Problem

$$\min_{S \subseteq V} \sum_{i=1}^{r} f_i(S)$$

→

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

f_i simple $\Rightarrow$ quadratic min over $B(f_i)$ easy

Discrete Problem

$$\min_{S \subseteq V} \sum_{i=1}^{r} f_i(S)$$

→

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

Decomposition allows for fast and practical algorithms based on gradient descent

[Stobbe, Krause '10; Jegelka, Bach, Sra '13; Nishihara, Jegelka, Jordan '14, E., Nguyen '15]

Discrete Problem

$$\min_{S \subseteq V} \sum_{i=1}^{r} f_i(S)$$

→

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

[E., Nguyen ICML '15]
Minimize the convex formulation using
Random Coordinate Gradient Descent

Also accelerated algorithm

Fastest running times currently known

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

$$g(y) := \left\| \sum_{i=1}^{r} y_i \right\|^2$$

Random Coordinate Descent Algorithm

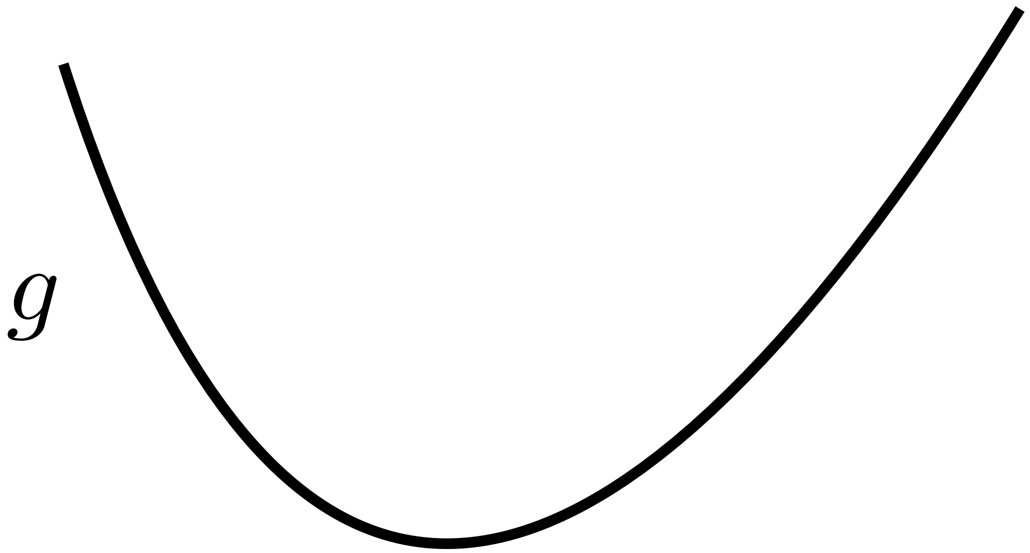
Start with $y = (y_1, \ldots, y_r)$, where $y_i \in B(f_i)$

In each iteration

Pick an index $i \in \{1, 2, ..., r\}$ uniformly at random

$\langle\!\langle$ *Update block i* $\rangle\!\rangle$

$$y_i' \leftarrow \operatorname*{argmin}_{z_i \in B(f_i)} \left(g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2 \right)$$



Smoothness

$$\|\nabla_i g(y) - \nabla_i g(z)\| \le \|y_i - z_i\|$$

y and z differ only in block i

<u>Random Coordinate Descent Algorithm</u>

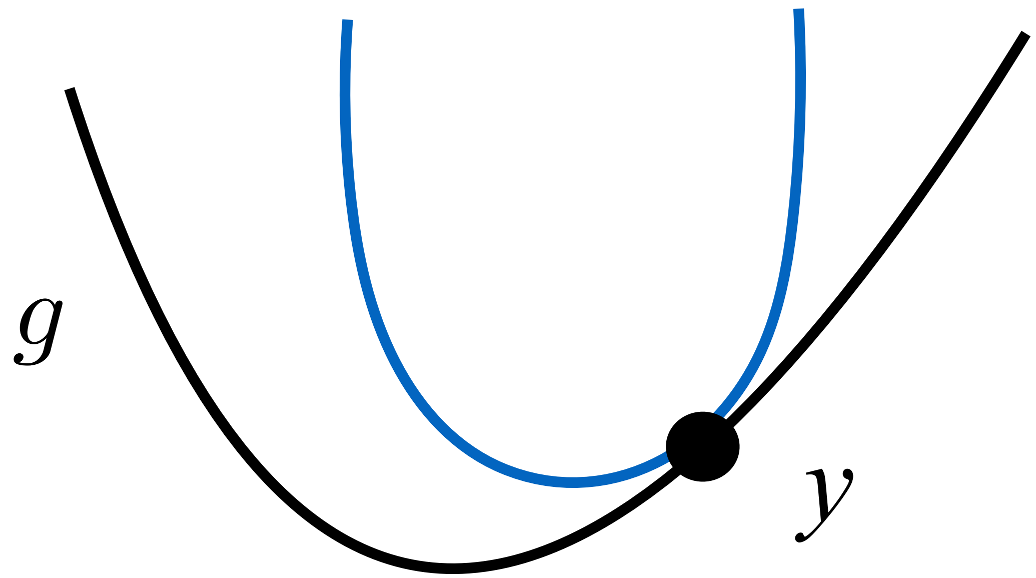
Start with $y = (y_1, \ldots, y_r)$, where $y_i \in B(f_i)$

In each iteration

Pick an index $i \in \{1, 2, \ldots, r\}$ uniformly at random

$\langle\langle$ *Update block i* $\rangle\rangle$

$$y'_i \leftarrow \operatorname*{argmin}_{z_i \in B(f_i)} \left(g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2 \right)$$



Smooth functions have quadratic upper bounds

Random Coordinate Descent Algorithm

Start with $y = (y_1, \ldots, y_r)$, where $y_i \in B(f_i)$

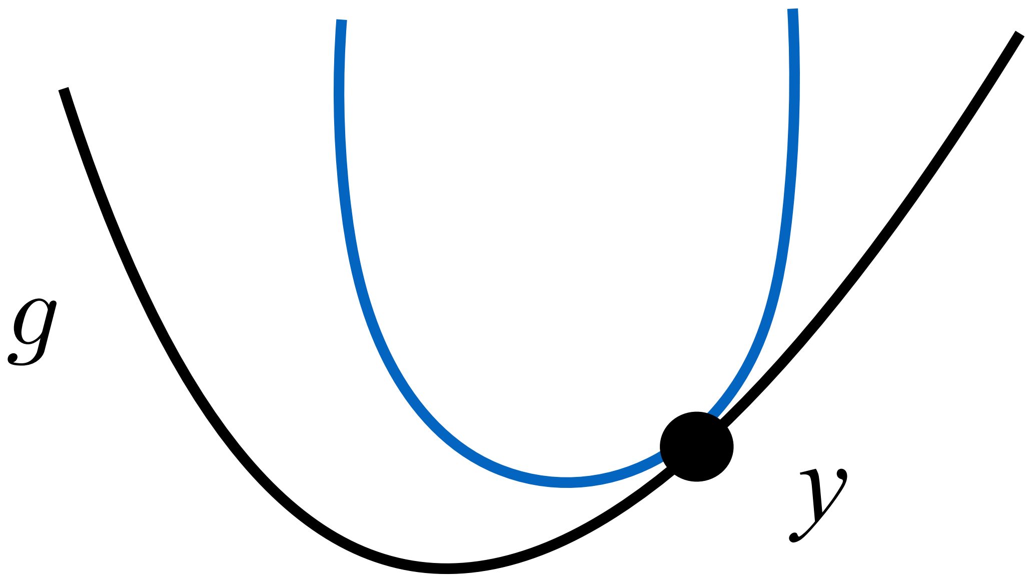
In each iteration

Pick an index $i \in \{1, 2, ..., r\}$ uniformly at random

$\langle\langle$*Update block* $i\rangle\rangle$

$$y_i' \leftarrow \underset{z_i \in B(f_i)}{\operatorname{argmin}} \left(g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2 \right)$$

$$\phi(z) = g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2$$



z and y differ only in block i

Random Coordinate Descent Algorithm

Start with $y = (y_1, \ldots, y_r)$, where $y_i \in B(f_i)$

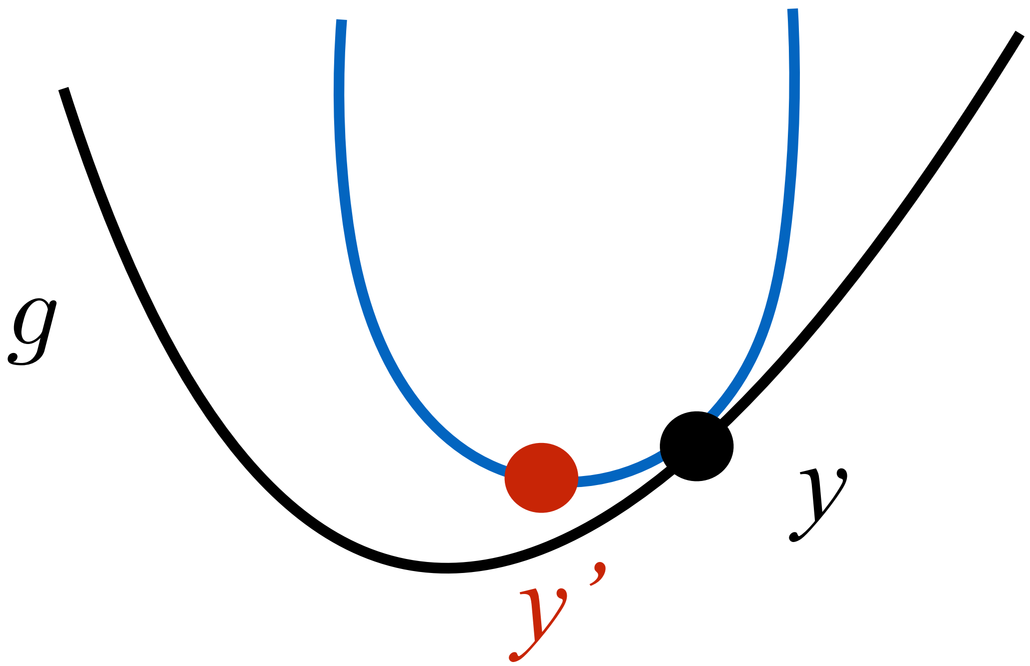
In each iteration

Pick an index $i \in \{1, 2, \ldots, r\}$ uniformly at random

$\langle\langle$ *Update block i* $\rangle\rangle$

$$y'_i \leftarrow \operatorname*{argmin}_{z_i \in B(f_i)} \left(g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2 \right)$$

$$\phi(z) = g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2$$



z and y differ only in block i

Minimize the upper bound

Random Coordinate Descent Algorithm

Start with $y = (y_1, \ldots, y_r)$, where $y_i \in B(f_i)$

In each iteration

Pick an index $i \in \{1, 2, \ldots, r\}$ uniformly at random

$\langle\langle$ *Update block i* $\rangle\rangle$

$$y'_i \leftarrow \operatorname*{argmin}_{z_i \in B(f_i)} \left(g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2 \right)$$

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

$$g(y) := \left\| \sum_{i=1}^{r} y_i \right\|^2$$

Random Coordinate Descent Algorithm

Start with $y = (y_1, \ldots, y_r)$, where $y_i \in B(f_i)$

In each iteration

Pick an index $i \in \{1, 2, ..., r\}$ uniformly at random

$\langle\langle$ *Update block i* $\rangle\rangle$

$$y_i' \leftarrow \operatorname*{argmin}_{z_i \in B(f_i)} \left(g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2 \right)$$

Efficient update for simple functions

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

$$g(y) := \left\| \sum_{i=1}^{r} y_i \right\|^2$$

Random Coordinate Descent Algorithm

Start with $y = (y_1, \ldots, y_r)$, where $y_i \in B(f_i)$

In each iteration

Pick an index $i \in \{1, 2, ..., r\}$ uniformly at random

$\langle\langle$ *Update block i* $\rangle\rangle$

$$y_i' \leftarrow \operatorname*{argmin}_{z_i \in B(f_i)} \left(g(y) + \langle \nabla_i g(y), z_i - y_i \rangle + \|z_i - y_i\|^2 \right)$$

Also accelerated version building on [Fercoq-Richtárik '13]

Discrete Problem

$$\min_{S \subseteq V} \sum_{i=1}^{r} f_i(S)$$

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

[E., Nguyen ICML '15]

Minimize the convex formulation using

Random Coordinate Gradient Descent

Discrete Problem

$$\min_{S \subseteq V} \sum_{i=1}^{r} f_i(S)$$

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

[E., Nguyen ICML '15]

Minimize the convex formulation using

Random Coordinate Gradient Descent

Running time (with acceleration) [ENV '17]

$$O\left(nr \log\left(\frac{F_{\max}}{\epsilon}\right) Q\right)$$

Q: time for one update via oracle for f_i

Discrete Problem

$$\min_{S \subseteq V} \sum_{i=1}^{r} f_i(S)$$

Exact Convex Formulation

$$\min_{y_i \in B(f_i)} \left\| \sum_{i=1}^{r} y_i \right\|^2$$

[E., Nguyen ICML '15]
Minimize the convex formulation using
Random Coordinate Gradient Descent

Running time (with acceleration) [ENV '17]

$$O(nr \log(nF_{\max})\, Q)$$

Q: time for one update via oracle for f_i

Discrete Algorithms

The quest for combinatorial algos for submodular min

[Cunningham '85]
It is an outstanding open problem to find a practical combinatorial algorithm to minimize a general submodular function, which also runs in polynomial time.

The quest for combinatorial algos for submodular min

[Cunningham '85]
It is an outstanding open problem to find a practical combinatorial algorithm to minimize a general submodular function, which also runs in polynomial time.

[Bixby et al. '85, Cunningham '85]
Framework for designing combinatorial algorithms for submodular minimization

Basic framework for all combinatorial algorithms

Cunningham Framework

Maintain a point y in the base polytope

Iteratively update y using "flow-style" updates

How to represent a point in the base polytope?

Basic framework for all combinatorial algorithms

Cunningham Framework

Maintain a point y in the base polytope

Iteratively update y using "flow-style" updates

How to represent a point in the base polytope?

As a convex combination of vertices

Pro: Very general (works for any submodular fn)

Con: Difficult/expensive to maintain and update

For decomposable submodular functions simpler version of Cunningham's approach works!

Recall: For $f = \sum_{i=1}^{r} f_i$ we have $B(f) = \sum_{i=1}^{r} B(f_i)$

Can represent $y \in B(f)$ as

$y = y_1 + \cdots + y_r$ where $y_i \in B(f_i)$

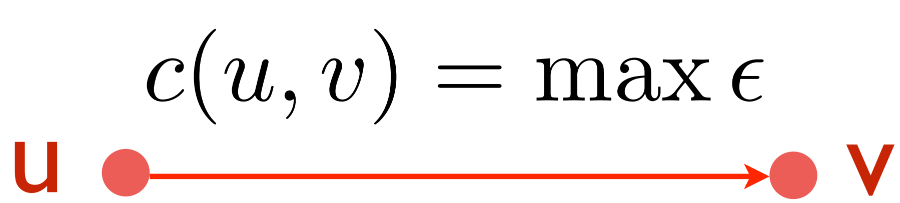
Verify $y_i \in B(f_i)$ using oracle for f_i

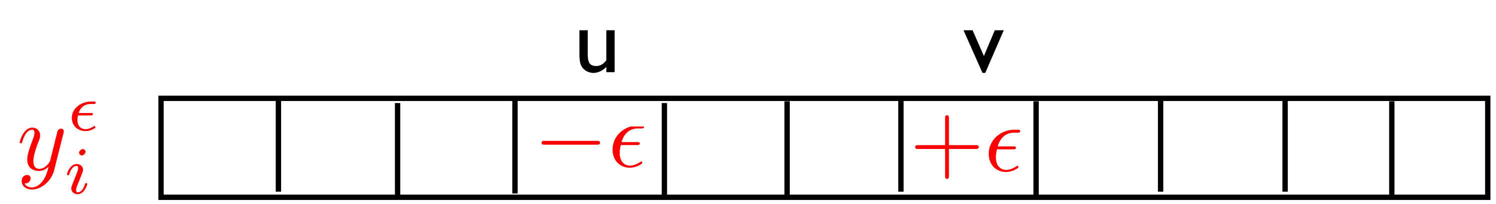
From decomposable functions to graphs

Maintain a point $y \in B(f)$ with a decomposition $y = y_1 + \cdots + y_r$ into points $y_i \in B(f_i)$

Update using auxiliary graph $G = (V, \cup_i E_i)$

$(u, v) \in E_i$ if $y_i^\epsilon \in B(f_i)$ for some $\epsilon > 0$

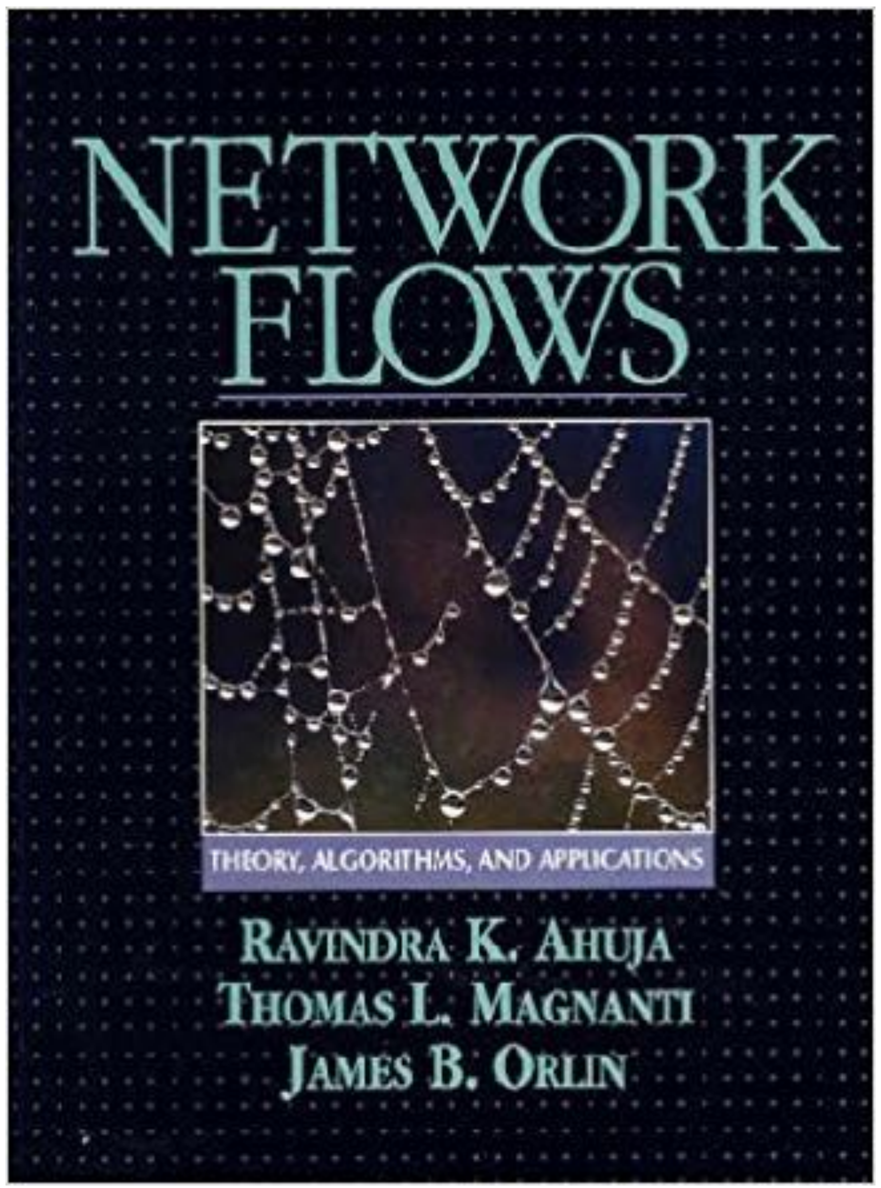


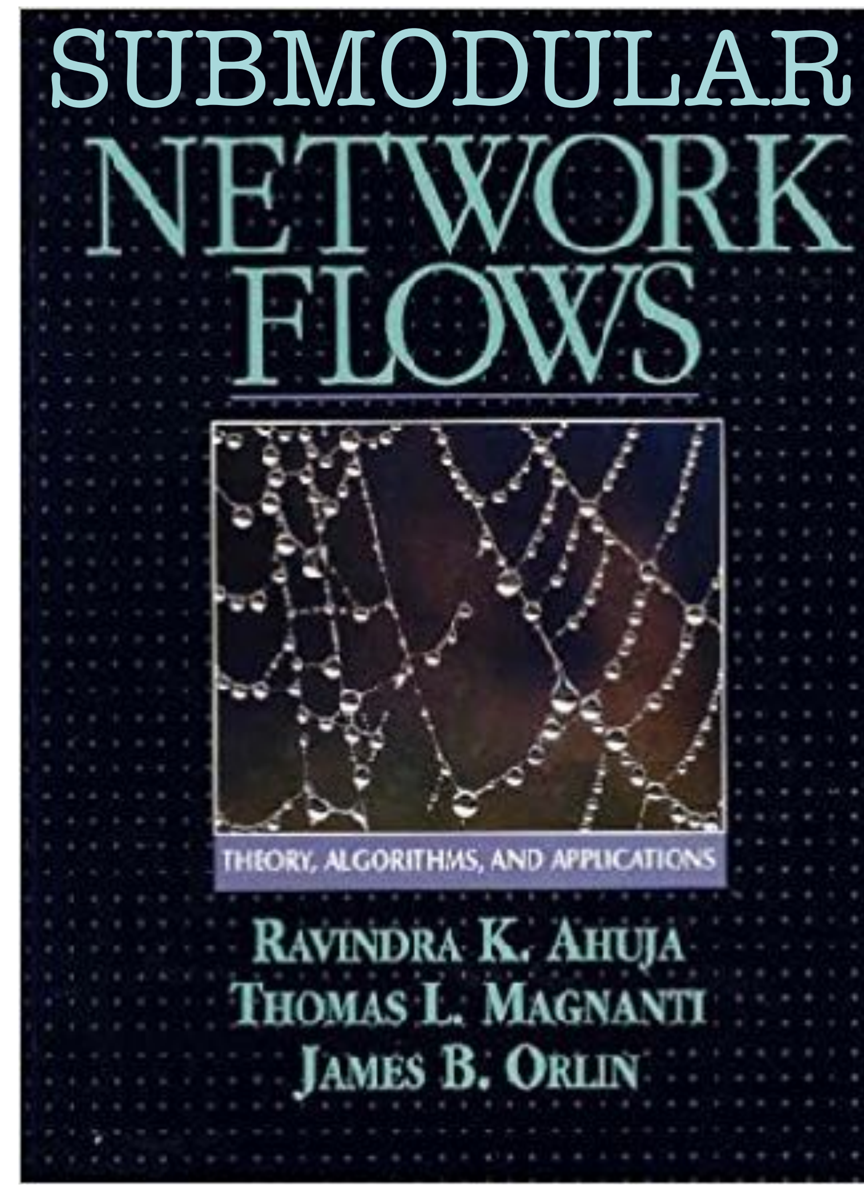


For graph cut instance, the auxiliary graph is the usual residual graph for network flows

Translate classical maxflow toolkit to submodular

Edmonds-Karp-Dinitz, Preflow-push, …





Translate classical maxflow toolkit to submodular

Edmonds-Karp-Dinitz, Preflow-push, …

Sample Result: If all f_i have small support preflow-push runs in $O(n^2 r)$ oracle calls

First shown by [Fujishige-Zhang '94] for r = 2

[Fix et al. '13] adapt IBFS algo. of [Goldberg et al. '11]

Translate classical maxflow toolkit to submodular

Edmonds-Karp-Dinitz, Preflow-push, …

[E., Nguyen, Vegh NIPS '17]

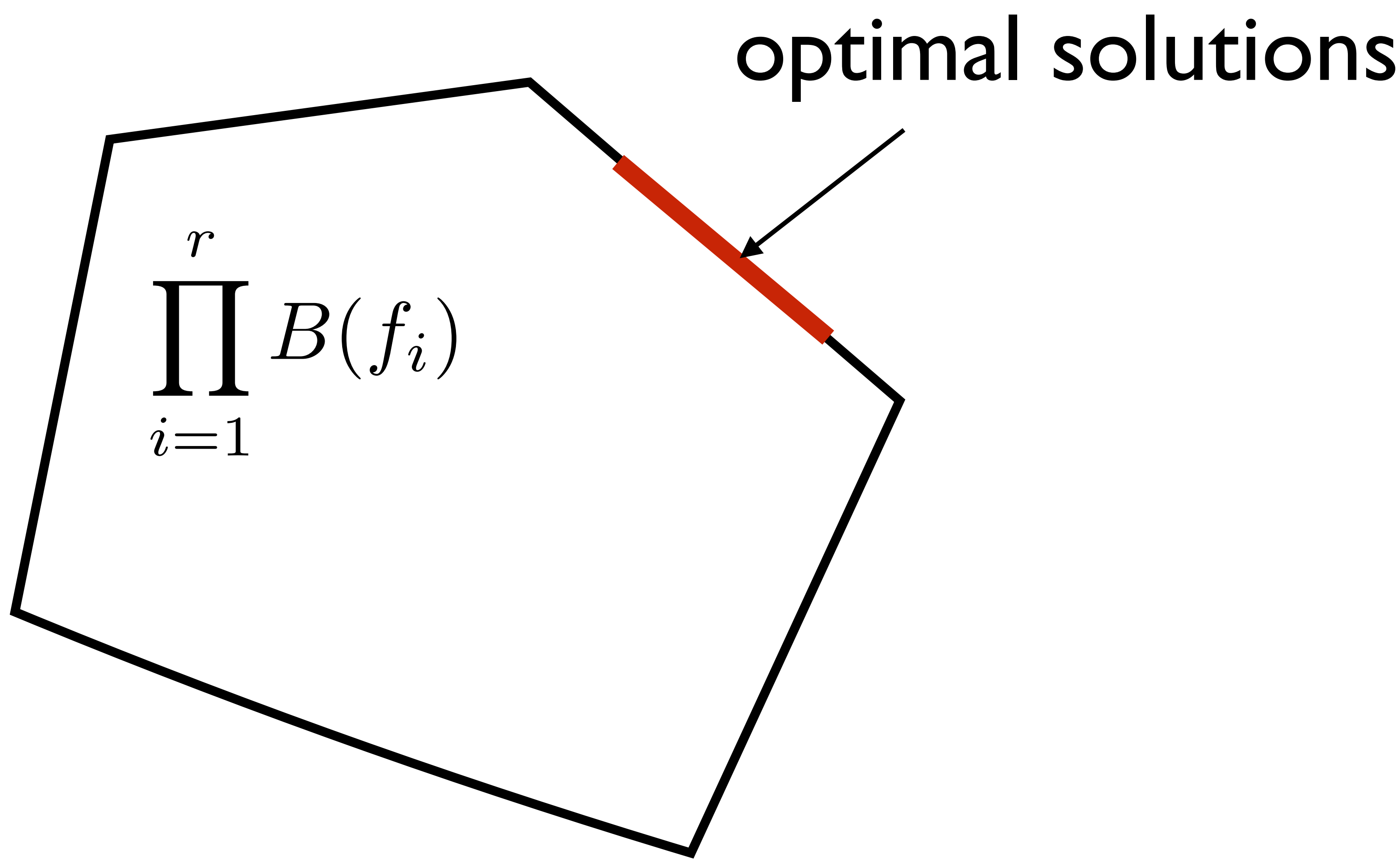
Discrete to Continuous

Tight running time analyses for the gradient descent algorithms

Discrete vs Continuous

Experimental comparison on computer vision tasks

Recall our problem: $\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$

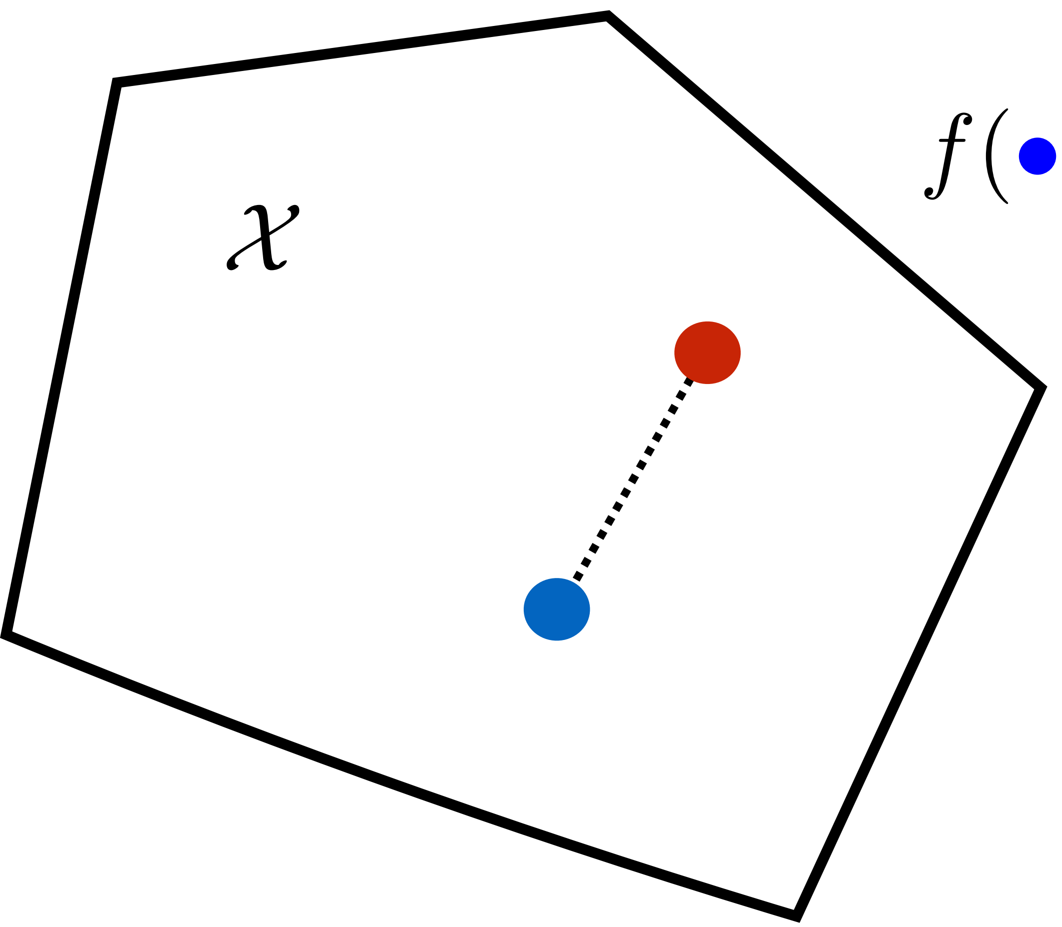


Minimize objective using random coordinate descent

Linear convergence rate despite lack of strong convexity

Use combinatorial structure instead of strong convexity

$$\min\left\{f(x) : x \in \mathcal{X}\right\}$$

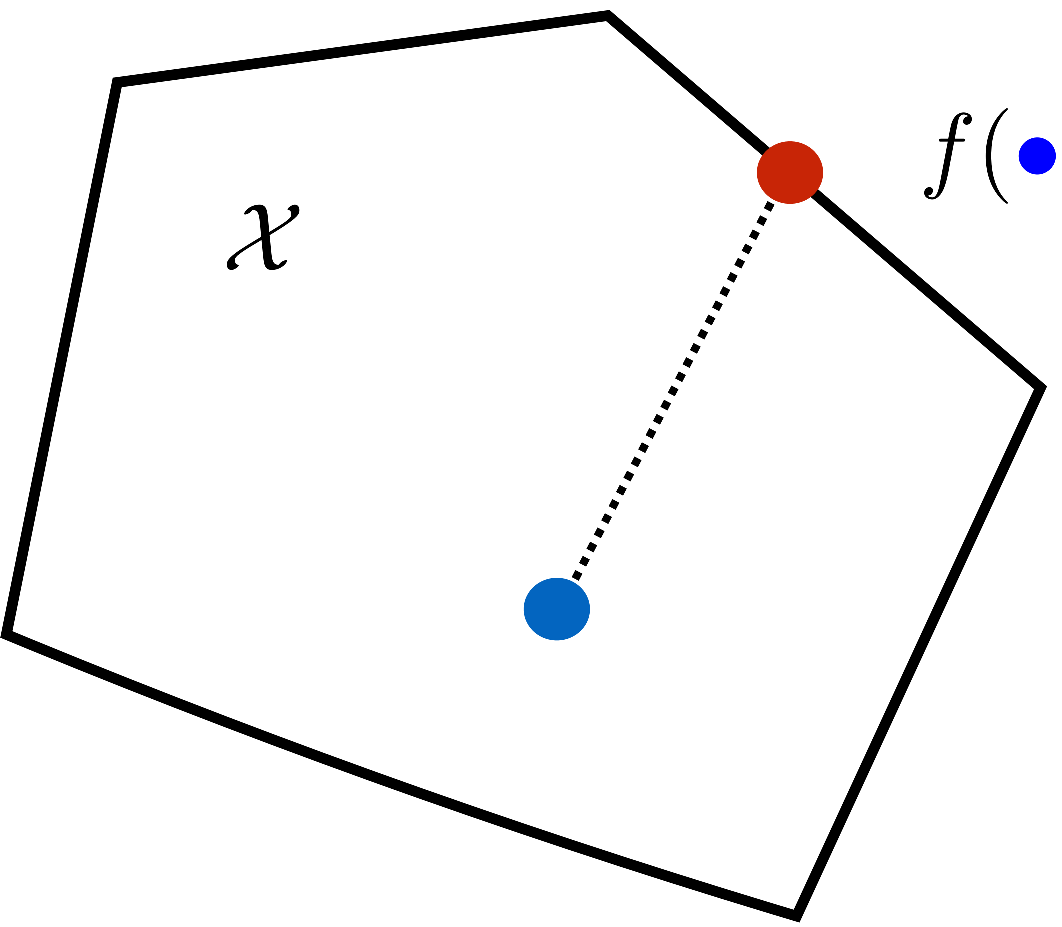


μ-Strong Convexity

$$f(\bullet) - f(\bullet) \geq \langle \nabla f(\bullet), \bullet - \bullet \rangle + \Omega_\mu(d(\bullet, \bullet)^2)$$

for any $\bullet, \bullet \in \mathcal{X}$

$$\min\left\{f(x) : x \in \mathcal{X}\right\}$$



μ-Strong Convexity

$$f({\color{blue}\bullet}) - f({\color{red}\bullet}) \geq \langle \nabla f({\color{red}\bullet}), {\color{blue}\bullet} - {\color{red}\bullet} \rangle + \Omega_\mu(d({\color{blue}\bullet}, {\color{red}\bullet})^2)$$

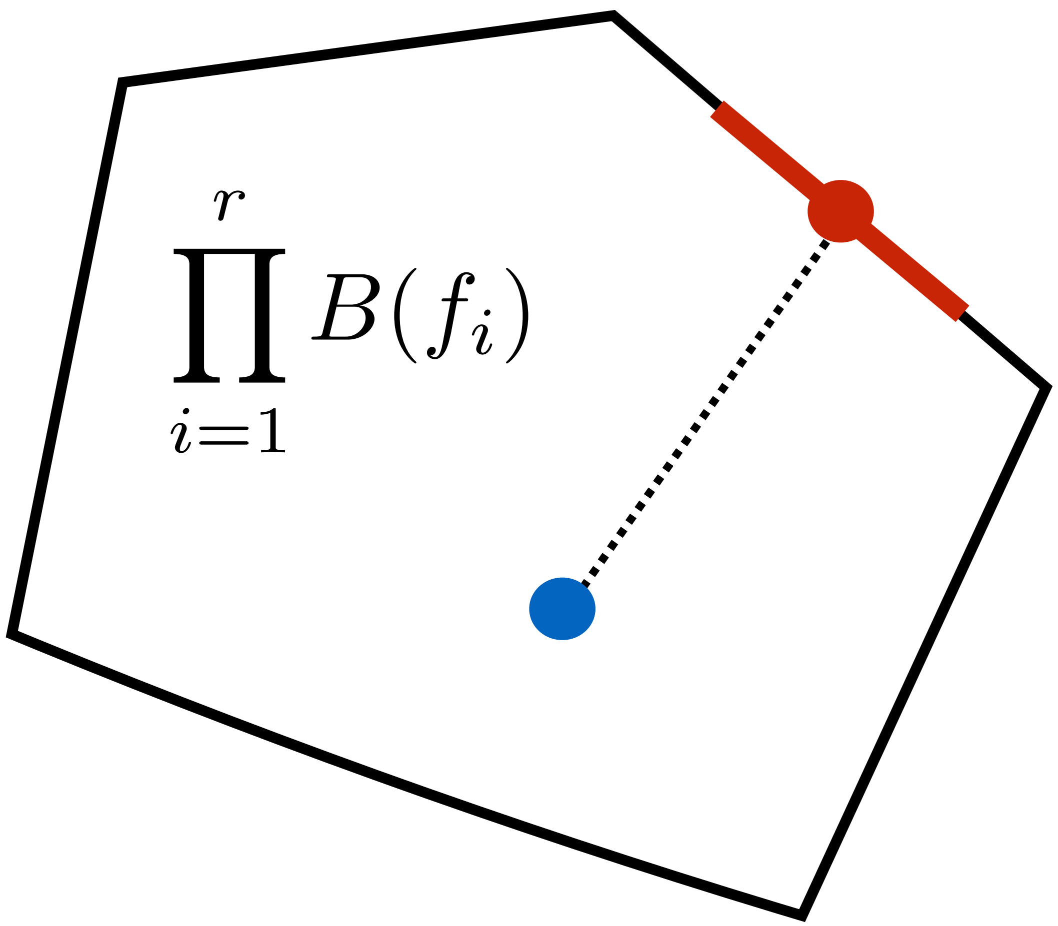
for any ${\color{blue}\bullet}, {\color{red}\bullet} \in \mathcal{X}$

What we really need

$$f({\color{blue}\bullet}) - f({\color{red}\bullet}) \geq \Omega_\mu(\min_{\color{red}\bullet} d({\color{blue}\bullet}, {\color{red}\bullet})^2)$$

where ${\color{red}\bullet}$ is an optimal point

Our problem: $\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$

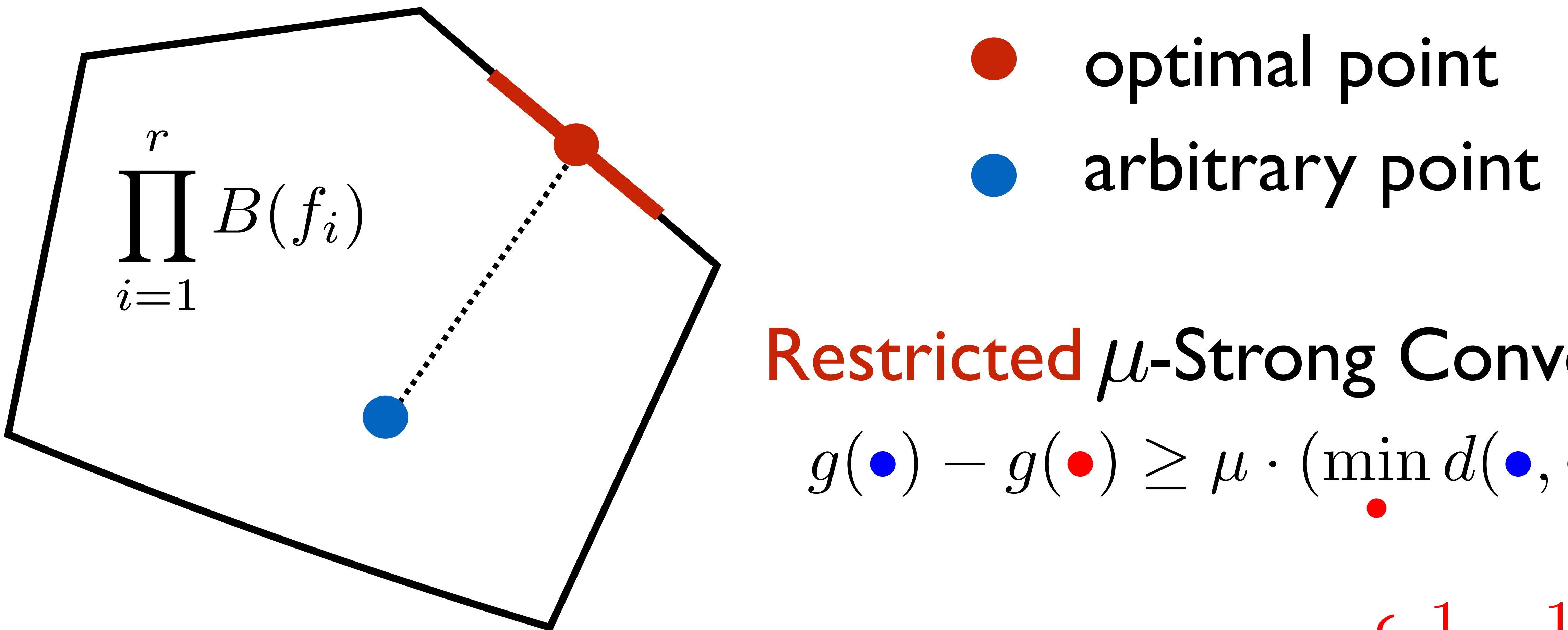


- optimal point
- arbitrary point

Restricted μ-Strong Convexity

$$g(\bullet) - g(\bullet) \geq \mu \cdot (\min_{\bullet} d(\bullet, \bullet)^2)$$

Our problem: $\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$

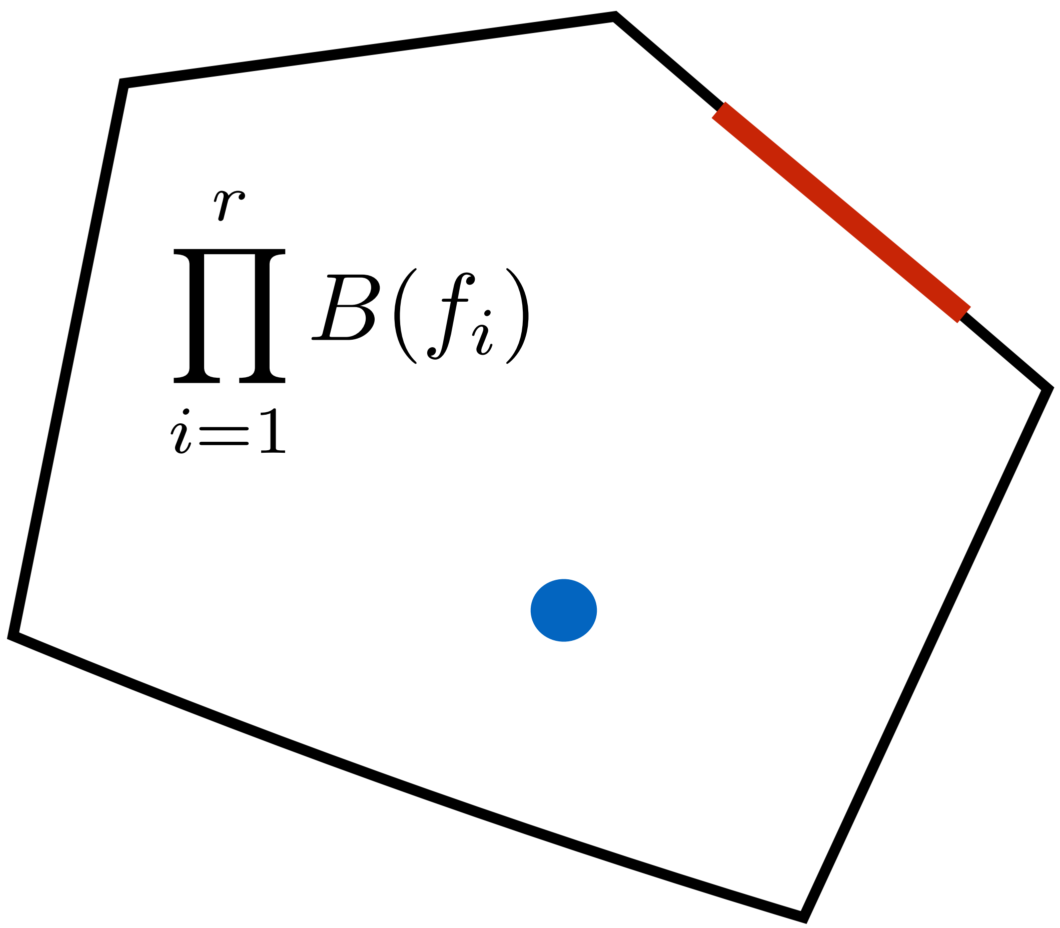


Restricted μ-Strong Convexity

$$g(\color{blue}{\bullet}) - g(\color{red}{\bullet}) \geq \mu \cdot (\min_{\color{red}{\bullet}} d(\color{blue}{\bullet}, \color{red}{\bullet})^2)$$

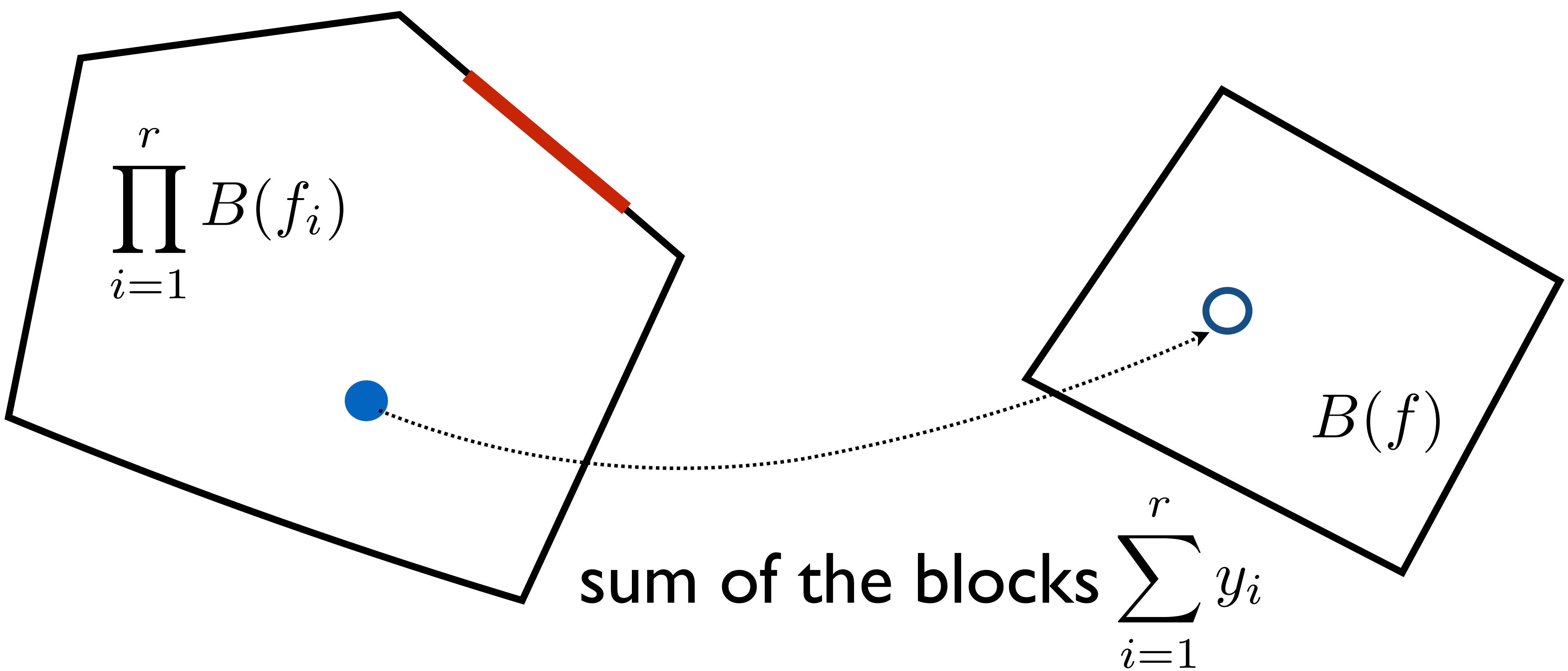
Can show $\mu \geq \max\left\{\frac{1}{nr}, \frac{1}{n^2}\right\}$

Our problem: $\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$



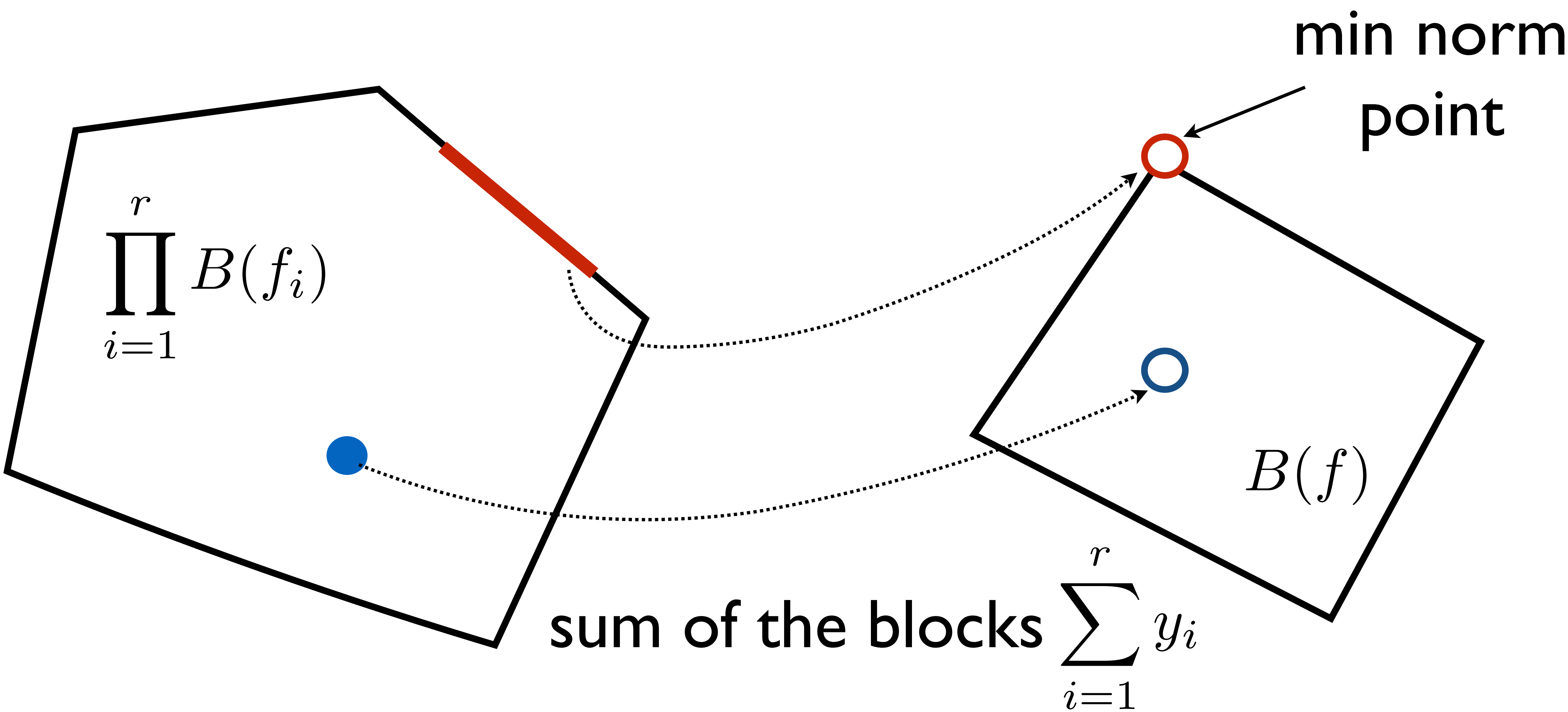
Want: g(●) - g(●) $\geq \frac{1}{n^2} \cdot$ d(● , ●)^2

Our problem: $\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$



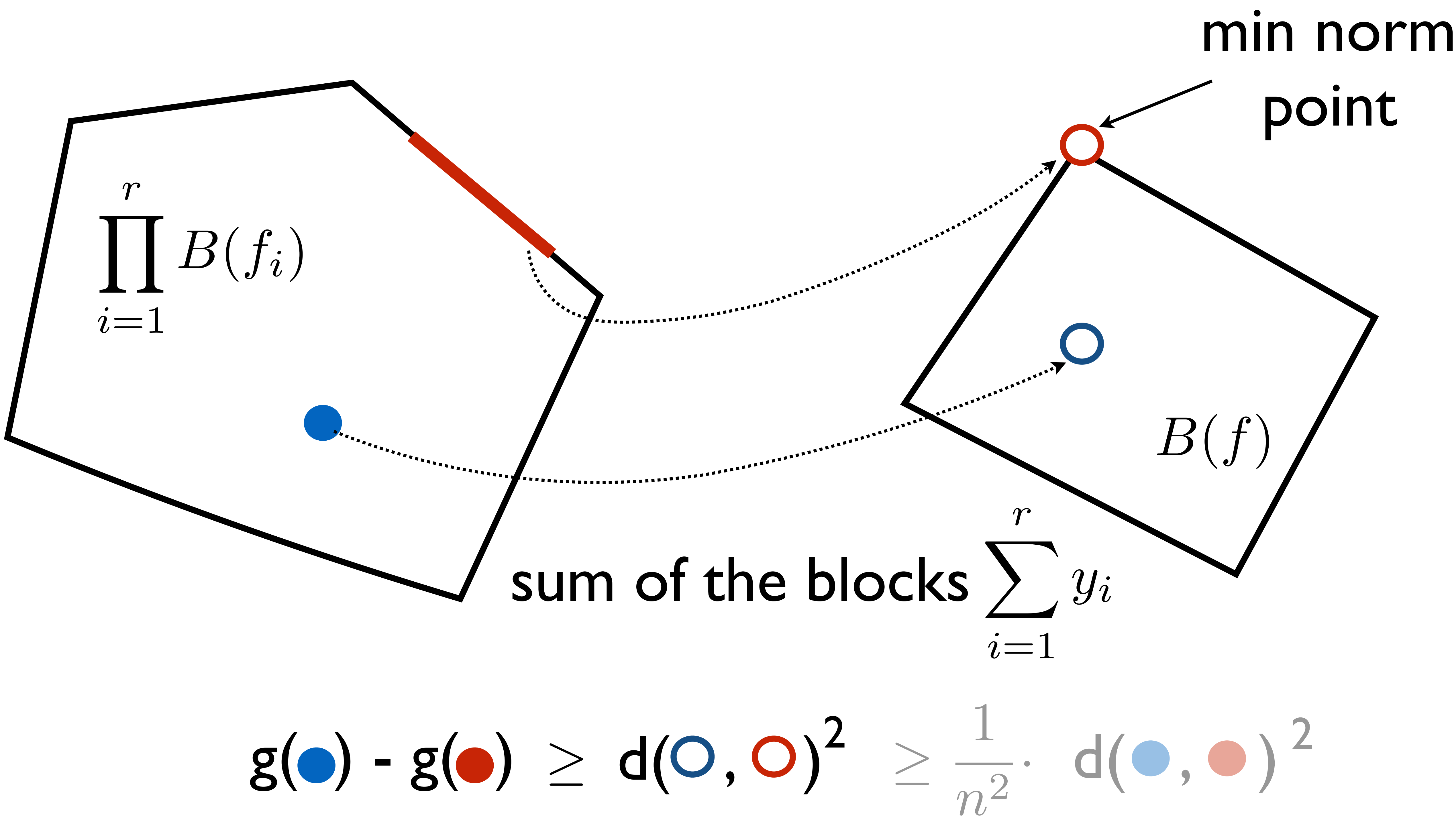
Want: g(●) - g(●) $\geq \frac{1}{n^2} \cdot$ d(●, ●)^2

Our problem: $\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$

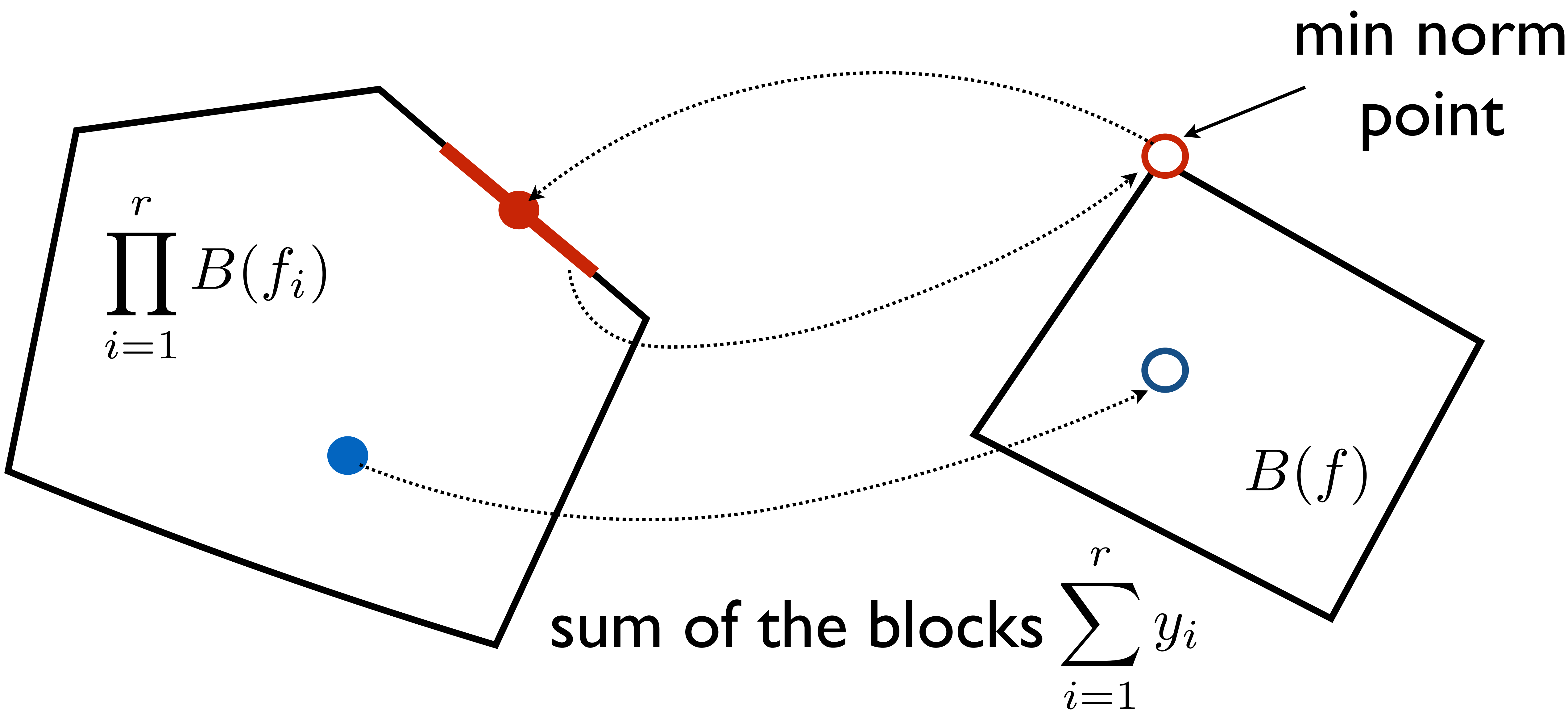


Want: g(●) - g(●) $\geq \frac{1}{n^2} \cdot$ d(●, ●)^2

Our problem: $$\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$$

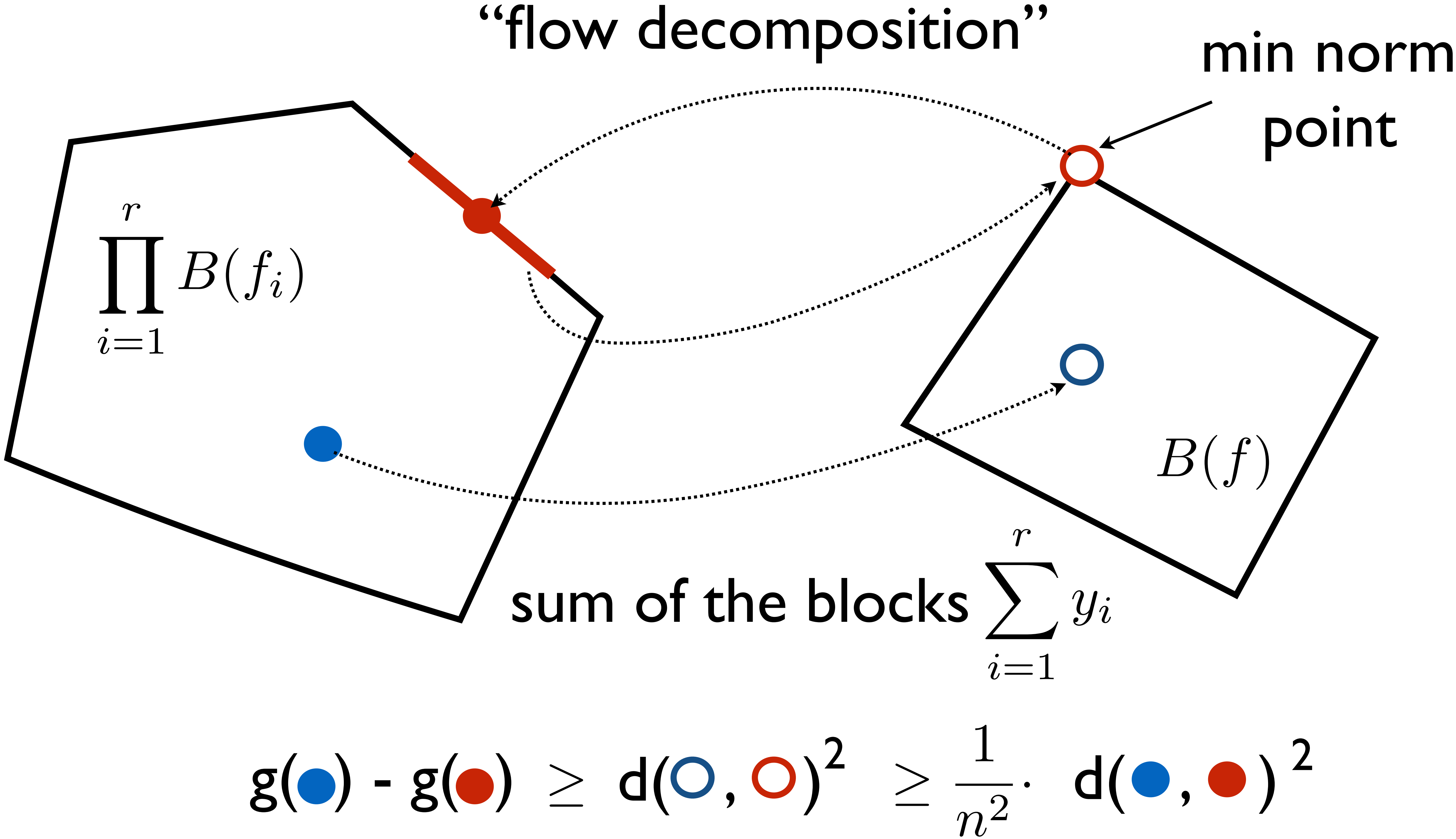


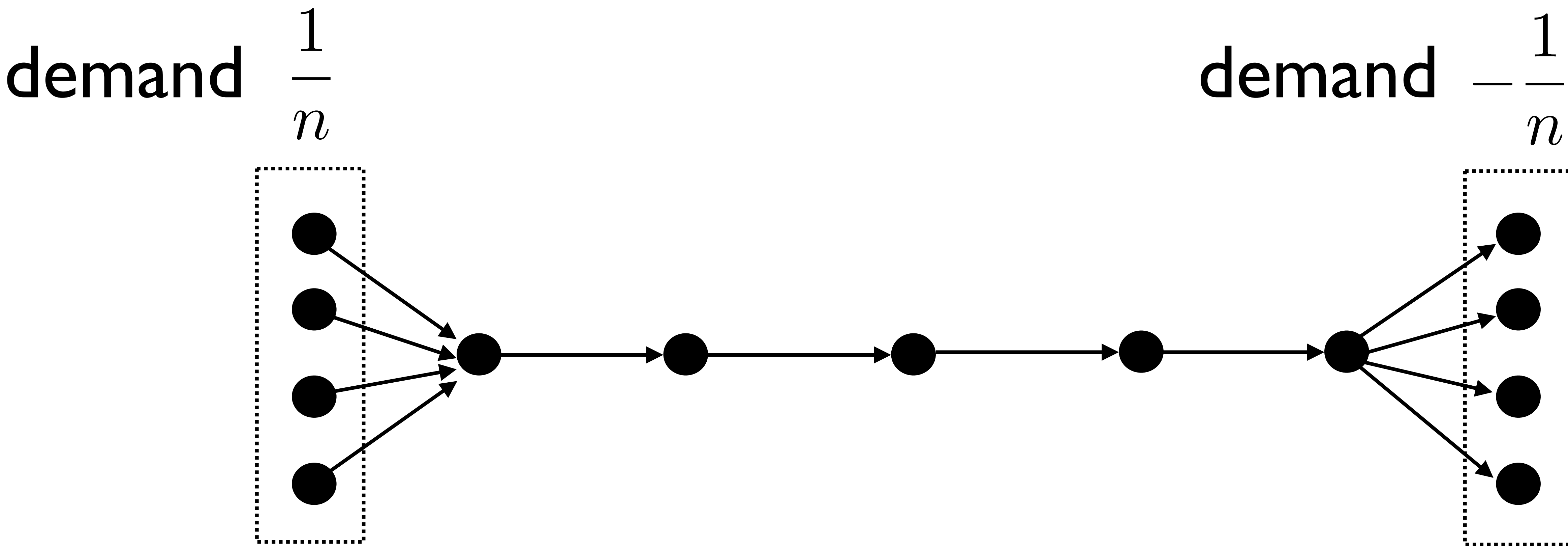
Our problem: $\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$



g(●) - g(●) $\geq$ d(○, ○)^2 $\geq \frac{1}{n^2} \cdot$ d(●, ●)^2

Our problem: $\min \left\{ g(y) = \left\| \sum_{i=1}^{r} y_i \right\|^2 : y \in \prod_{i=1}^{r} B(f_i) \right\}$





$$\ell_2^2 \text{ of demand} = \frac{1}{n^2} \cdot n$$

$$\ell_2^2 \text{ of flow} = n$$

$$\ell_2^2 \text{ of demand} = \frac{1}{n^2} \cdot (\ell_2^2 \text{ of flow})$$

Summary

Algorithms that leverage both convex optimization and the combinatorial structure (via graph max flow)