Provable Submodular Minimization using Wolfe's Algorithm

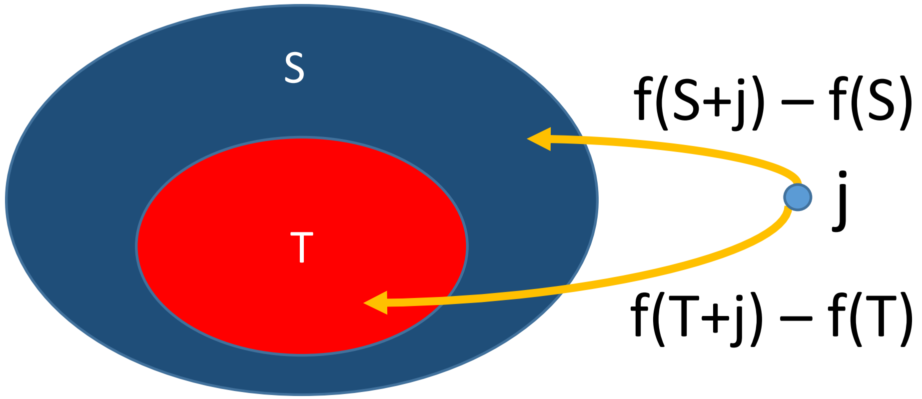
Deeparnab Chakrabarty (Microsoft Research)

Prateek Jain (Microsoft Research)

Pravesh Kothari (U. Texas)

Submodular Functions

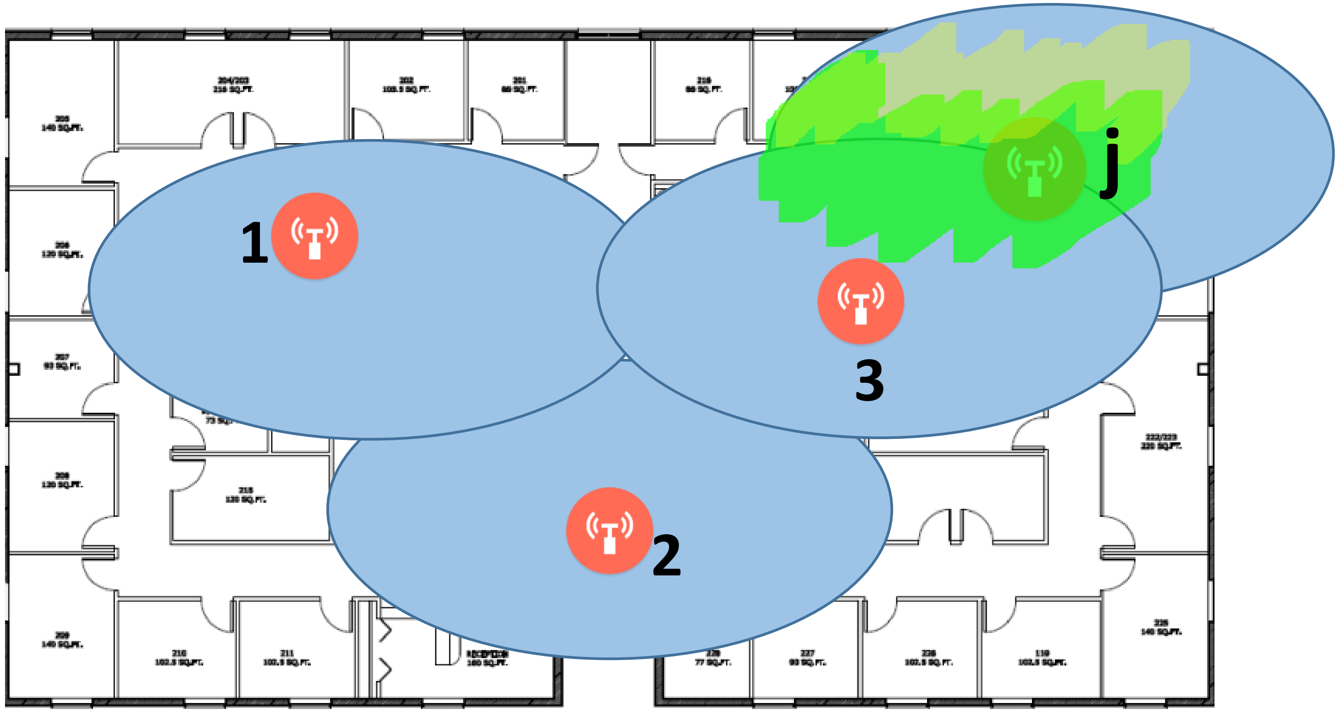
- f : Subsets of {1,2,..,n} → integers
- **Diminishing Returns Property.**



For all S, for all $T \subseteq S$,
for all j not in S:
$f(S + j) - f(S) \leq f(T + j) - f(T)$

- f may or may not be **monotone.**

Sensor Networks



Universe: Sensor Locations. f(A) = “Area covered by sensors”

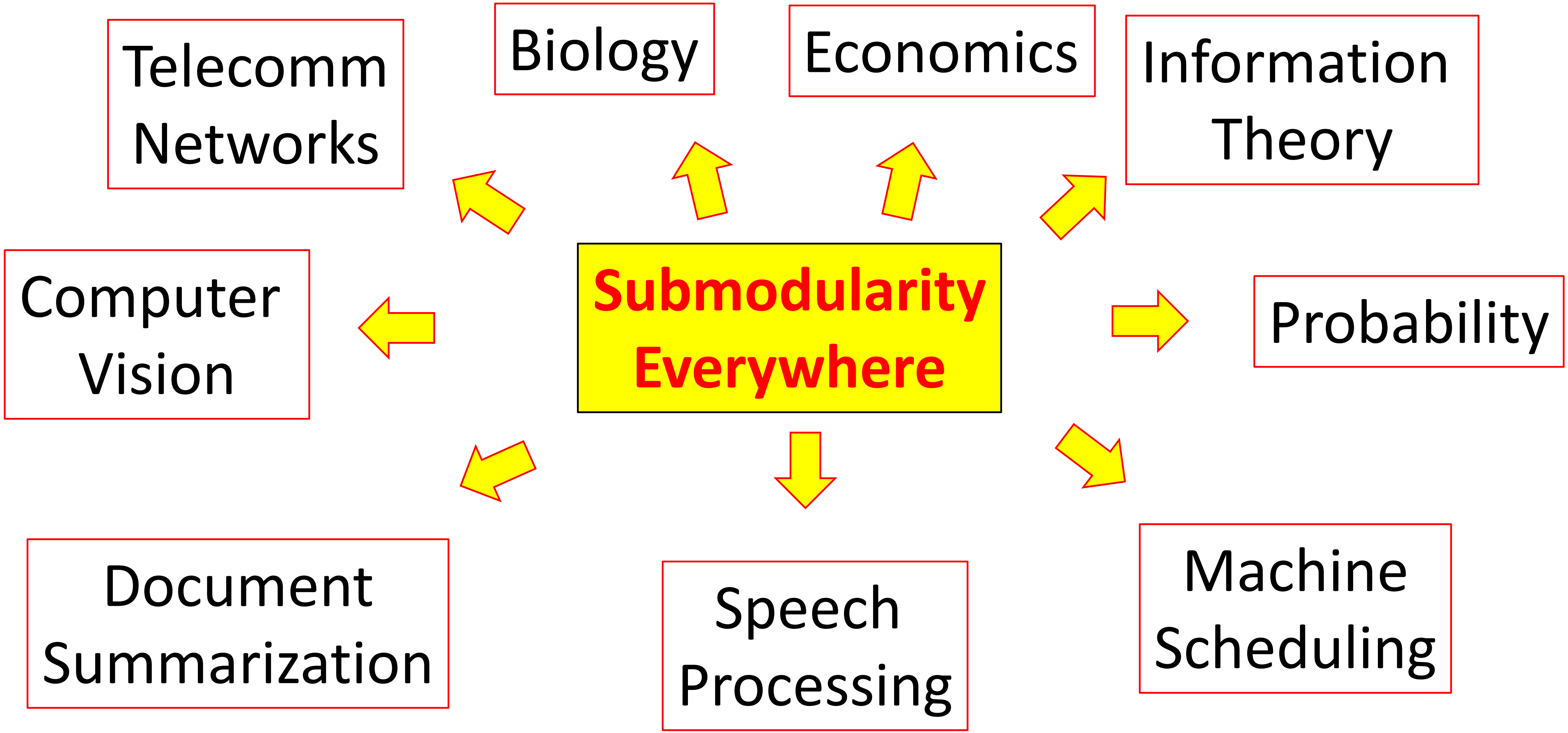
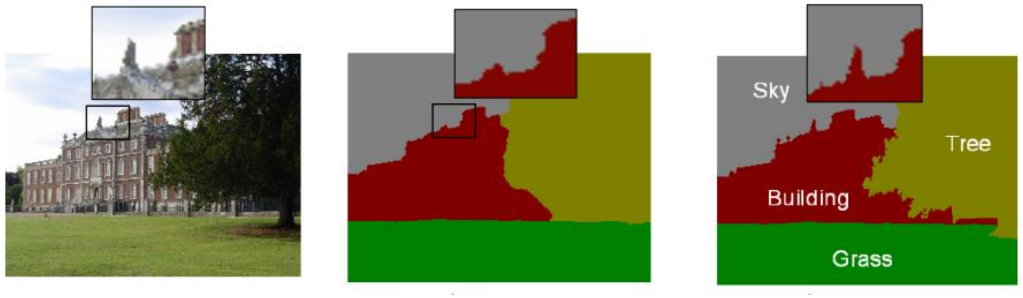

Submodularity Everywhere
Telecomm Networks
Biology
Economics
Information Theory
Computer Vision
Probability
Document Summarization
Speech Processing
Machine Scheduling

Image Segmentation



Labelling — "Energy" function — Observed Image

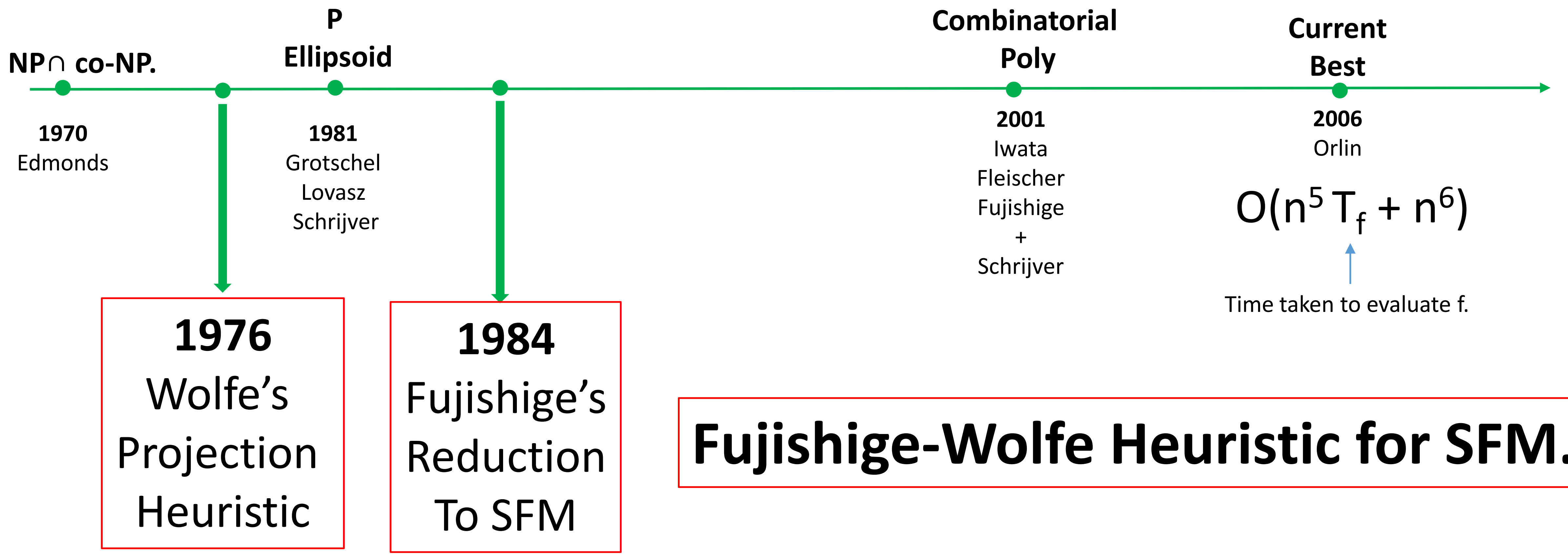
$$\mathbf{X} = \arg\min \mathbf{E}(\mathbf{X}|\mathbf{D})$$

Energy minimization done via reduction to submodular function minimization.

(Boykov, Veksler, Zabih 2001) (Kolmogorov Boykov 2004) (Kohli, Kumar, Torr 2007) (Kohli Ladicky Torr 2009)

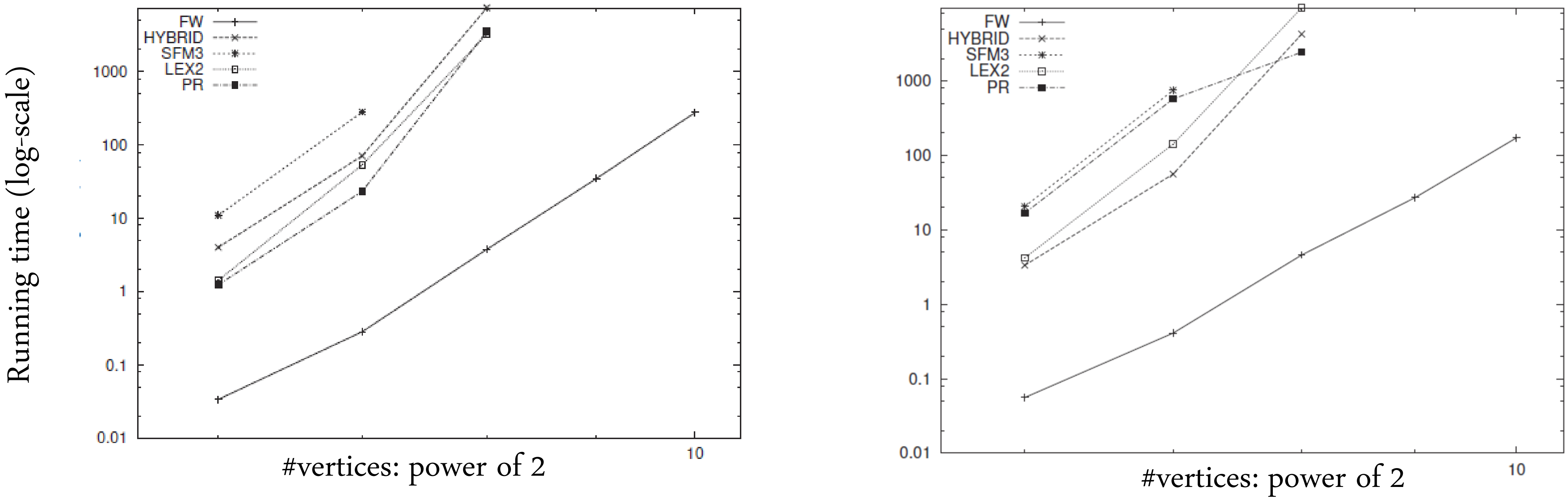
Submodular Function Minimization

Find set S which minimizes f(S)



Fujishige-Wolfe Heuristic for SFM.

Theory vs Practice



Cut functions from DIMACS Challenge

(Fujishige, Isotani 2009)

Is it good in theory?

Theoretical guarantee so far

$$2^{O(n^2)}$$

Today

Fujishige-Wolfe is (pseudo)**polynomial.**

$$O(n^7F^2),\ F = \max_i |f(\{i\})|$$

Fujishige-Wolfe Heuristic

- **Fujishige Reduction.** Submodular minimization reduced to finding nearest-to-**origin** point (i.e., a **projection**) of the **base** polytope.

- **Wolfe's Algorithm.** Finds the nearest-to-origin point of **any** polytope. Reduces to **linear optimization** over that polytope.

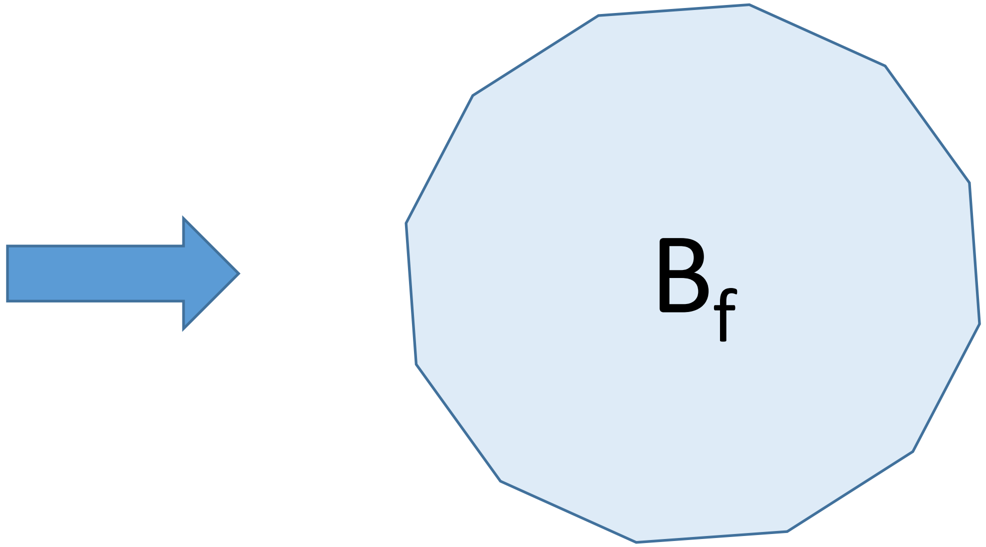
Our Results

- **First convergence analysis** of Wolfe's algorithm for projection on **any** polytope. *How quickly can we get within ε of optimum?* (THIS TALK)

- **Robust generalization of Fujishige Reduction.** When small enough, ε-close points can give exact submodular function minimization.

Base Polytope

Submodular function f



$$(x_1, \ldots, x_n) : \quad \sum_{i \in S} x_i \leq f(S) \quad \forall S \subseteq U$$

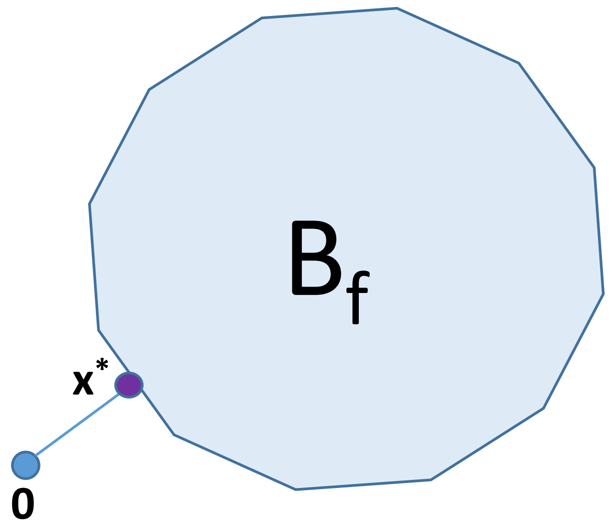
$$\sum_{i \in U} x_i = f(U)$$

Linear Optimization in almost **linear** time!

Fujishige's Theorem

$$(x_1, \ldots, x_n): \qquad \sum_{i \in S} x_i \leq f(S) \quad \forall S \subseteq U$$

$$\sum_{i \in U} x_i = f(U)$$

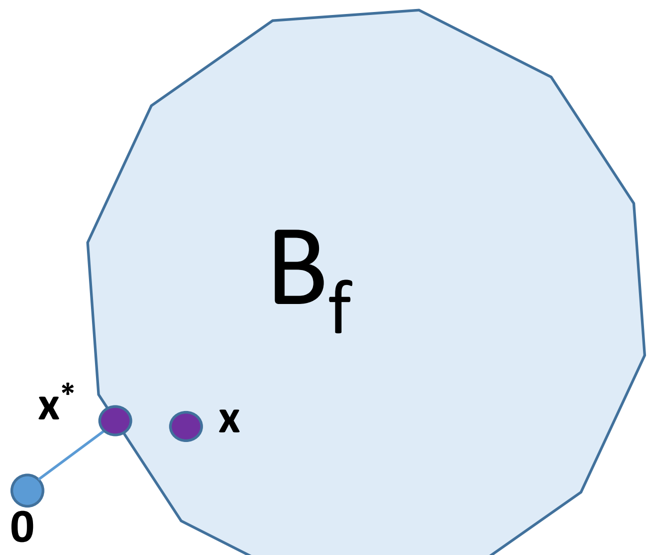


If x^* is the closest-to-origin point of B_f , then $A = \{j : x^*_j \leq 0\}$ is a minimizer of f.

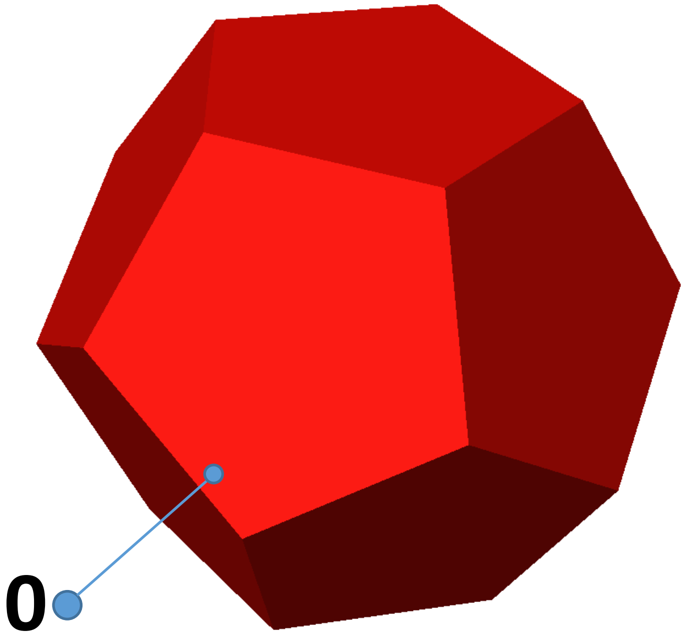
A Robust Version

Let x satisfy $||x-x^*|| \leq \varepsilon$.

Can read out a set B from x such that: $f(B) \leq f(A) + 2n\varepsilon$



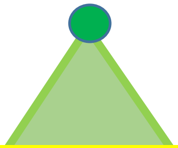
If f is **integral**, $\varepsilon < 1/2n$ implies **exact** SFM.



Wolfe's Algorithm: Projection onto a polytope

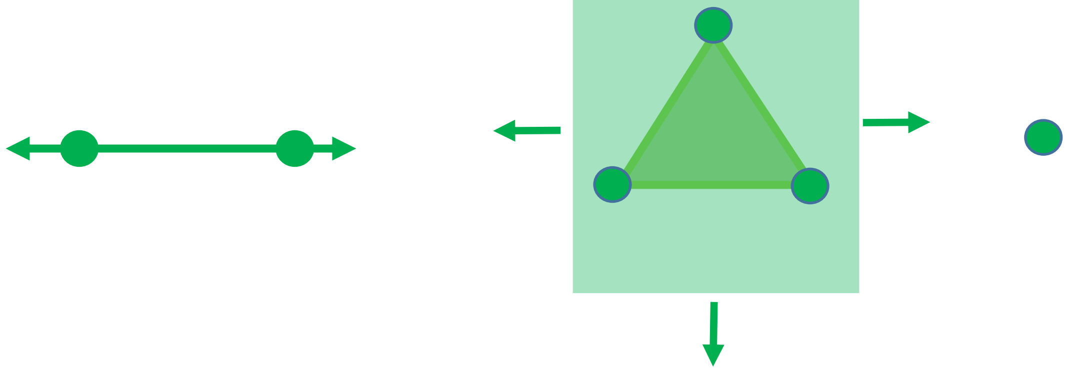
Geometrical preliminaries

Convex Hull: conv(S)



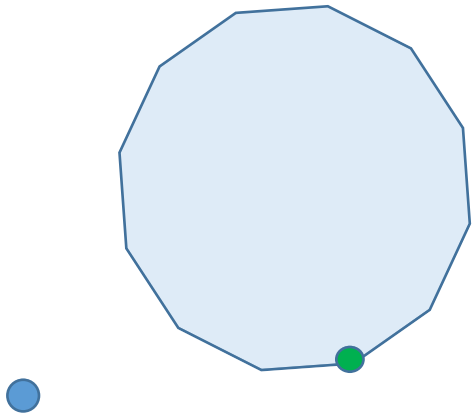
Finding closest-to-origin point on **aff**(S) is easy
Finding it on **conv**(S) is not.

Affine Hull: aff(S)

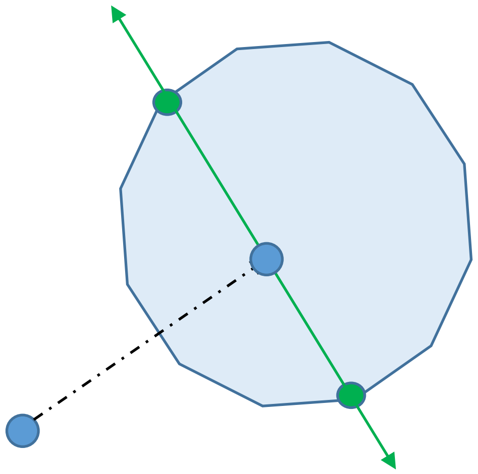


Corrals

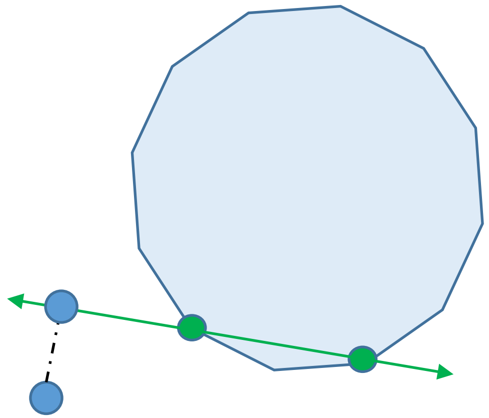
Set S of points s.t. the min-norm point in **aff**(S) lies in **conv**(S).



Trivial Corral



Corral



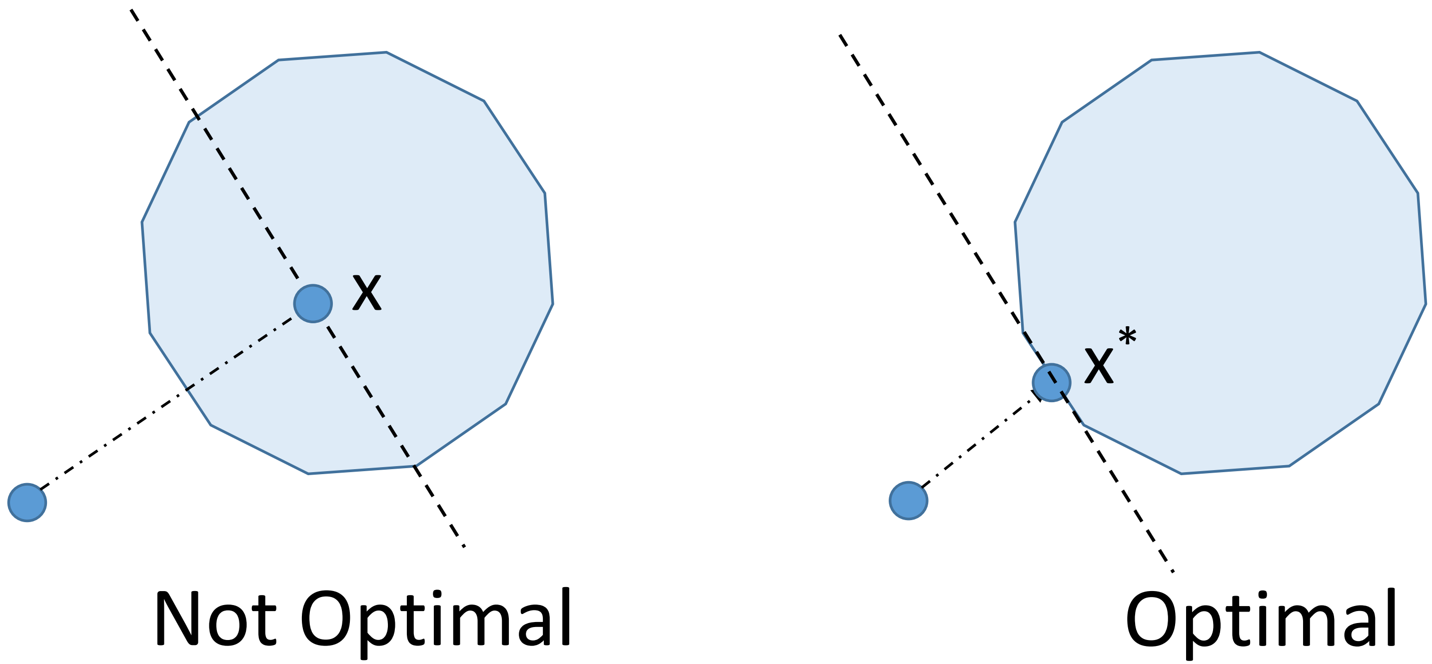
Not a Corral

Wolfe’s algorithm in a nutshell

- Moves from corral to corral till optimality.

- In the process it goes via “non-corrals”.

Checking Optimality



x is optimal iff $||x||^2 \leq x \cdot q$ for all $q \in P$

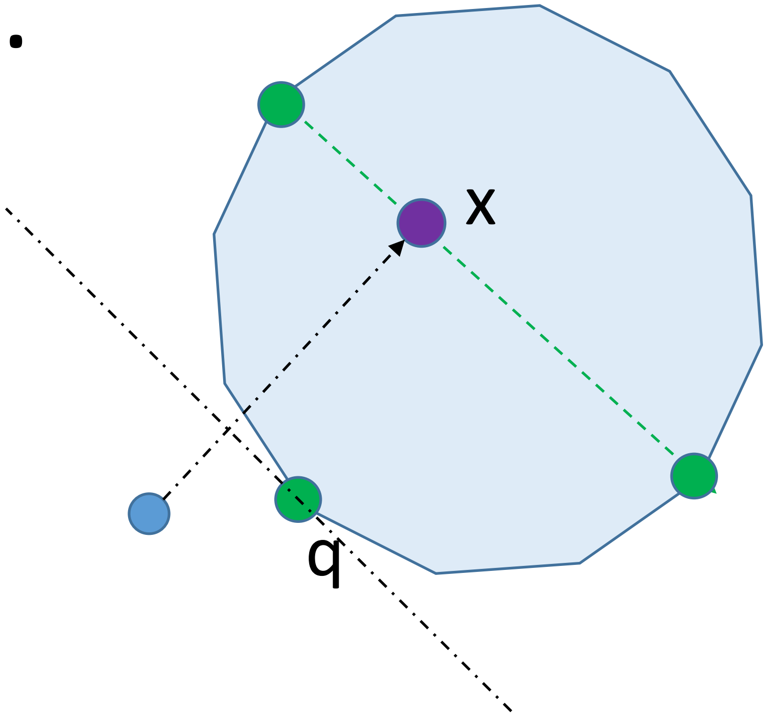
Wolfe's Algorithm: Details

- **State: (x,S).** S ⊆ vertices, x in **conv**(S)
- Start: S *arbitrary* vertex {q}, x = q.

- Till x is optimal, run **major** or **minor** cycle to update x and S.

If S **is** a corral: Major Cycle

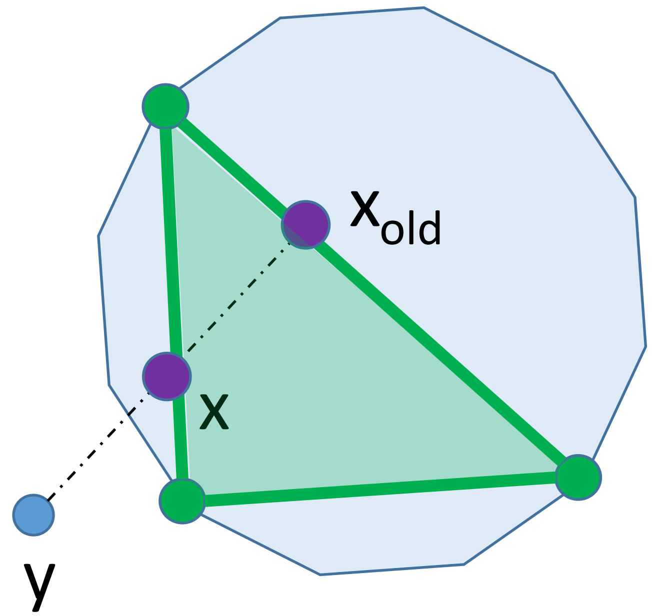
- x = min norm point in **aff**(S).
- $q = \arg\min_{p \text{ in } P} (p \cdot x)$
- S = S + q.



Major cycle **increments** |S|.

If S is not a corral: Minor Cycle

- y = min-norm point in **aff**(S)
- x = pt on $[y, x_{old}] \cap$ **conv**(S)
  closest to y
- **Remove** *irrelevant* points from S.



Minor cycle **decrements** |S|.

Summarizing Wolfe's Algorithm

- State: (x,S). x lies in **conv**(S).
- Each iteration is either a **major** or a **minor** cycle.
- Linear Programming and Matrix Inversion.
- **Major** cycles increment and **minor** cycles decrement |S|.
- In < **n** minor cycles, we get a major cycle, and vice versa.
- Norm strictly decreases. Corrals can't repeat.
  Finite termination.

Our Theorem

For any polytope P, for any $\varepsilon > 0$, in $O(nD^2/\varepsilon^2)$ iterations Wolfe's algorithm returns a point x such that $||x - x^*|| \leq \varepsilon$ where D is the diameter of P.

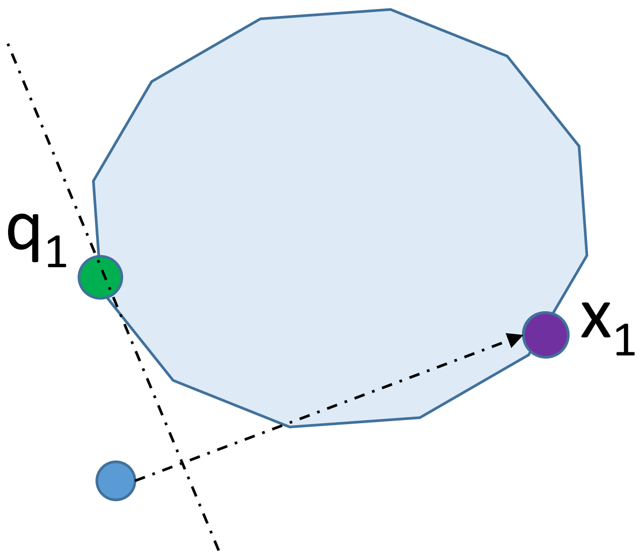
For SFM, the base polytope has diameter $D^2 < nF^2$.

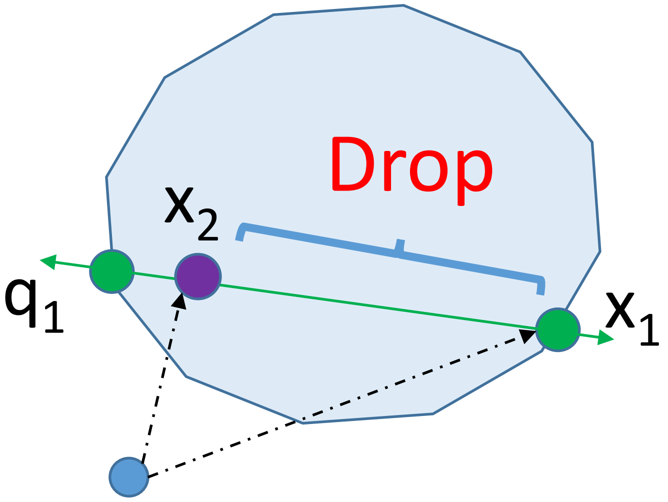
Outline of the Proof

- **Significant** norm decrease when far from optimum.

- Will argue this for two **major** cycles with **at most** one **minor** cycle in between.

Two Major Cycles in a Row





$$q_1 = \arg\min_{p \in P} \; p \cdot x_1$$

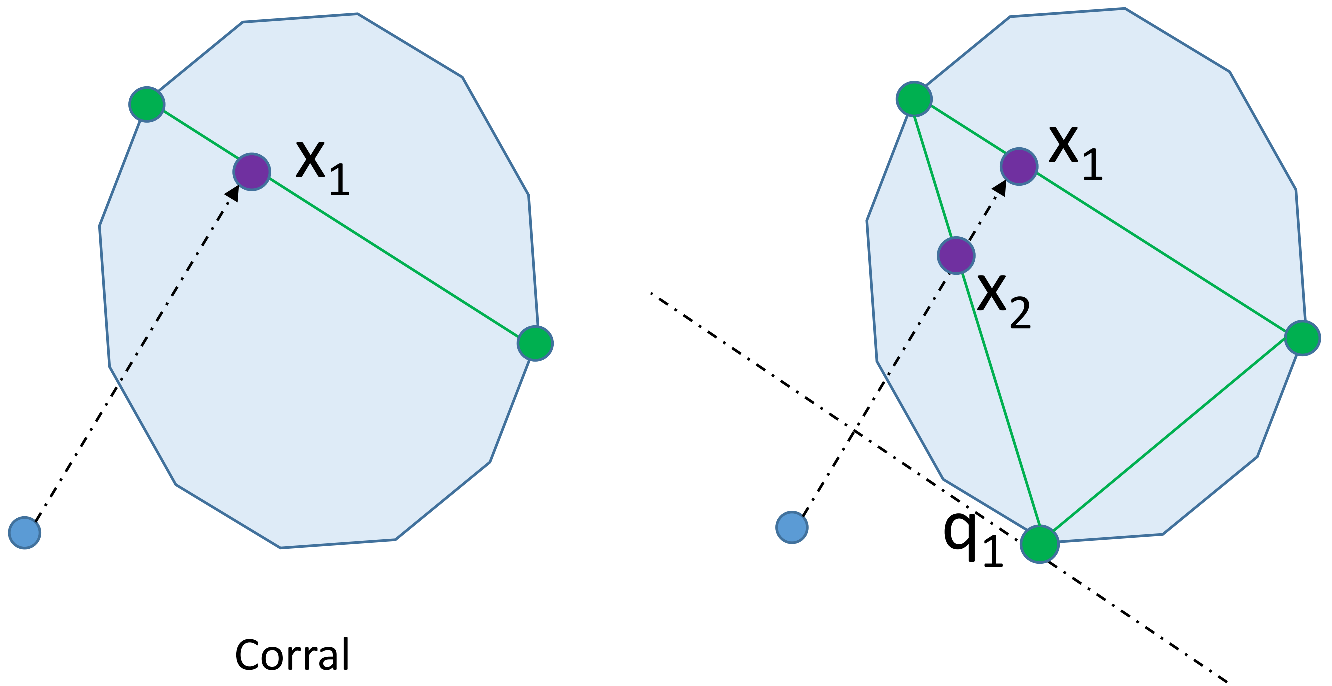
If x_1 is “far away”, then

$||x_1||^2 - x_1 \cdot q_1$ is “large”

$$\text{Drop} = \; x_1 \cdot \frac{(x_1 - q_1)}{D}$$

$$= \frac{||x_1||^2 - x_1 \cdot q_1}{D}$$

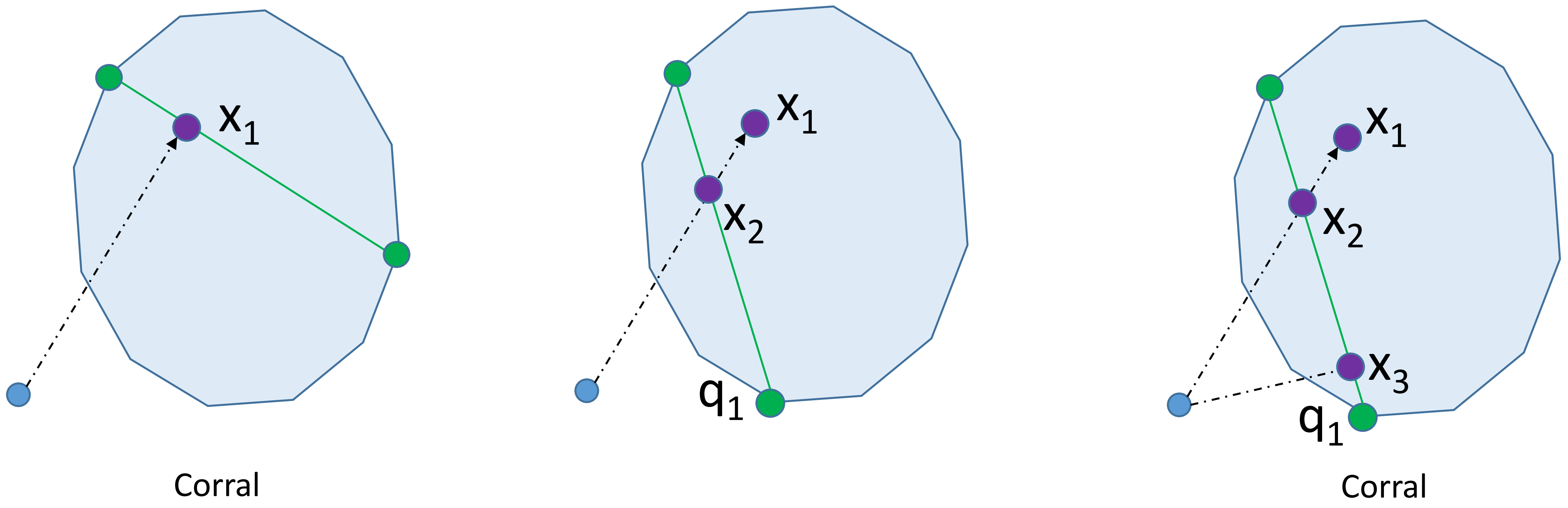
Major-minor-Major



$$q_1 = \arg\min_{p \in P} \; p \cdot x_1$$

aff(S + q_1) is the whole 2D plane. Origin is itself closest-to-origin

Major-minor-Major



Either x_2 "far away" from x_1 implying $||x_1||^2 - ||x_2||^2$ is large.

Or, x_2 "behaves like" x_1, and $||x_2||^2 - ||x_3||^2$ is large.

Outline of the Proof

- **Significant** norm decrease when far from optimum.

- Will argue this for two **major** cycles with **at most** one **minor** cycle in between.

- Simple combinatorial fact: in 3n iterations there must be one such “good pair”.

Take away points.

- Analysis of Wolfe's algorithm, a practical algorithm.
- Can one remove dependence on F?
- Can one **change** the Fujishige-Wolfe algorithm to get a better one, both in theory and in practice?

Thank you.