Graph Sketching, Sampling, Streaming, and Space Efficient Optimization (Part II)

Sudipto Guha and Andrew McGregor

Space Efficient Optimization for Graphs

Impact of Dimensionality Reduction, Embeddings, $L_p \rightarrow L_q$, etc.

Thesis: Graph optimization problems are natural next candidates.

(Part I): Building blocks: sketching, sampling in graphs.

Why?
How to use them?
How do we think these problems?

Space Efficient Optimization for Graphs

Impact of Dimensionality Reduction, Embeddings, $L_p \to L_q$, etc.

Thesis: Graph optimization problems are natural next candidates.

(Part I): Building blocks: sketching, sampling in graphs.

Why?
How to use them?
How do we think these problems?

Space Efficient Optimization.

- Storage grows. Problem sizes grow larger.
- Streaming=Organizing accesses in an algorithm.
- Sketching =Organizing information.
- Partition of input, model, output and algorithm.
- Processing Space $\neq$ Storage Space.

Optimization?

Many frameworks to choose from.

Linear/Convex programming.

1. A lot of general purpose techniques.
2. A rich history in graphs.
3. The connection to streaming is less well studied.

Correlation Clustering and Max Matchings (part I) as examples.
Rephrasing papers in SODA 2014, ICML 2015, SPAA 2015.

Tutorial Plan

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.
Feasibility, LP version

Tutorial Plan

(a) Recap of Multiplicative Weights Method.
Feasibility, LP version
Multiple perspectives on the algorithm.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.
Feasibility, LP version
Multiple perspectives on the algorithm.
What is the basic idea behind the proof?

Tutorial Plan

(a) Recap of Multiplicative Weights Method.
Feasibility, LP version
Multiple perspectives on the algorithm.
What is the basic idea behind the proof?
Where/how do we start from?

Tutorial Plan

(a) Recap of Multiplicative Weights Method.
Feasibility, LP version
Multiple perspectives on the algorithm.
What is the basic idea behind the proof?
Where/how do we start from?

(b) Application to Min Correlation Clustering.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.
Feasibility, LP version
Multiple perspectives on the algorithm.
What is the basic idea behind the proof?
Where/how do we start from?

(b) Application to Min Correlation Clustering.
How to design an Oracle.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.
Feasibility, LP version
Multiple perspectives on the algorithm.
What is the basic idea behind the proof?
Where/how do we start from?

(b) Application to Min Correlation Clustering.
How to design an Oracle.
"Drag and Drop" sparsification.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering
What if relaxations do not fit?

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering
What if relaxations do not fit?
How to "round" the fractional solution?

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering
What if relaxations do not fit?
How to "round" the fractional solution?

(d) Multiple Passes I: Max Bipartite Matching

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering
What if relaxations do not fit?
How to "round" the fractional solution?

(d) Multiple Passes I: Max Bipartite Matching
Optimization over fixed constraint matrices.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering
What if relaxations do not fit?
How to "round" the fractional solution?

(d) Multiple Passes I: Max Bipartite Matching
Optimization over fixed constraint matrices.
Use of Approximation Algorithms for speedup.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering
What if relaxations do not fit?
How to "round" the fractional solution?

(d) Multiple Passes I: Max Bipartite Matching
Optimization over fixed constraint matrices.
Use of Approximation Algorithms for speedup.
"Primal-Dual meets Primal-Dual" .

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering.

(d) Multiple Passes I: Max Bipartite Matching.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.
(b) Application to Min Correlation Clustering.
(c) Semi-Definite Programming (SDPs): Max Correlation Clustering.
(d) Multiple Passes I: Max Bipartite Matching.
(e) Multiple Passes II: Max Non-Bipartite Matching.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering.

(d) Multiple Passes I: Max Bipartite Matching.

(e) Multiple Passes II: Max Non-Bipartite Matching.
Exponentially many constraints. Constraint Sparsification!
How to find your way in the dark?

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering.

(d) Multiple Passes I: Max Bipartite Matching.

(e) Multiple Passes II: Max Non-Bipartite Matching.
Exponentially many constraints. Constraint Sparsification!
How to find your way in the dark?
MWM with Sparsification? Solution of optimization versus the path.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering.

(d) Multiple Passes I: Max Bipartite Matching.

(e) Multiple Passes II: Max Non-Bipartite Matching.
Exponentially many constraints. Constraint Sparsification!
How to find your way in the dark?
MWM with Sparsification? Solution of optimization versus the path.

(f) Multiple Passes III: Max Non-Bipartite Matching.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering.

(d) Multiple Passes I: Max Bipartite Matching.

(e) Multiple Passes II: Max Non-Bipartite Matching.
Exponentially many constraints. Constraint Sparsification!
How to find your way in the dark?
MWM with Sparsification? Solution of optimization versus the path.

(f) Multiple Passes III: Max Non-Bipartite Matching.
Few passes and a good algorithm.
Compute in parallel; use sequentially.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering.

(d) Multiple Passes I: Max Bipartite Matching.

(e) Multiple Passes II: Max Non-Bipartite Matching.
Exponentially many constraints. Constraint Sparsification!
How to find your way in the dark?
MWM with Sparsification? Solution of optimization versus the path.

(f) Multiple Passes III: Max Non-Bipartite Matching.
Few passes and a good algorithm.
Compute in parallel; use sequentially.
Dual-primal versus primal-dual. New relaxations for matching.

Tutorial Plan

(a) Recap of Multiplicative Weights Method.

(b) Application to Min Correlation Clustering.

(c) Semi-Definite Programming (SDPs): Max Correlation Clustering.

(d) Multiple Passes I: Max Bipartite Matching.

(e) Multiple Passes II: Max Non-Bipartite Matching.
Exponentially many constraints. Constraint Sparsification!
How to find your way in the dark?
MWM with Sparsification? Solution of optimization versus the path.

(f) Multiple Passes III: Max Non-Bipartite Matching.
Few passes and a good algorithm.
Compute in parallel; use sequentially.
Dual-primal versus primal-dual. New relaxations for matching.

(g) Wrap Up.

(a) Recap of Multiplicative Weights Method

Basic version.

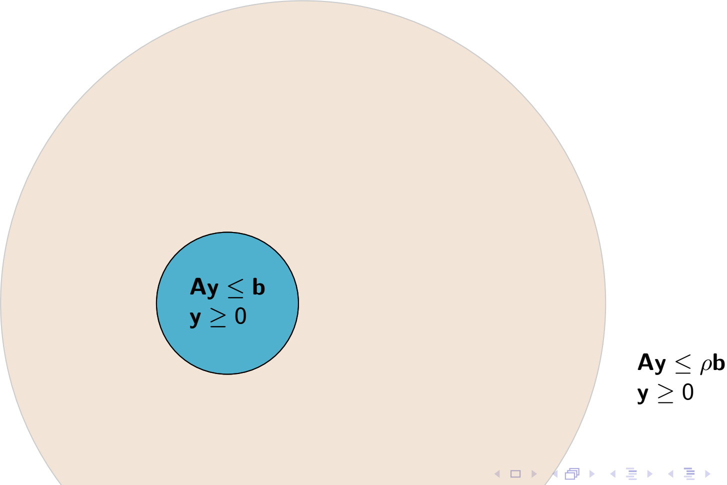
A proof sketch.

Alternate views.

Multiplicative Weights Method: Basic Version

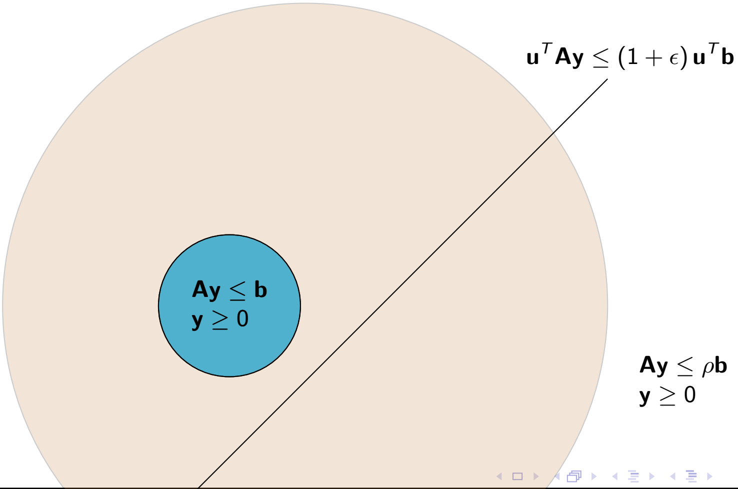
$$\mathbf{A}\mathbf{y} \leq \mathbf{b}$$
$$\mathbf{y} \geq 0$$

Multiplicative Weights Method: Basic Version



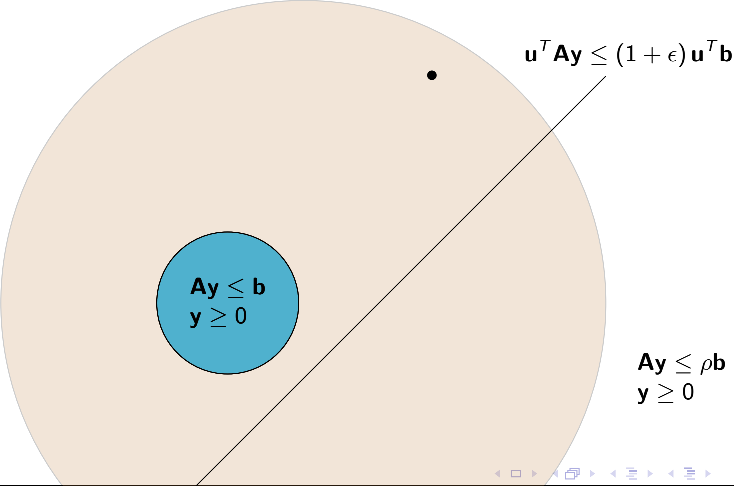
Multiplicative Weights Method: Basic Version

Initially $\mathbf{u} = \mathbf{1}$. Assume $\mathbf{A}, \mathbf{b} \geq \mathbf{0}$.

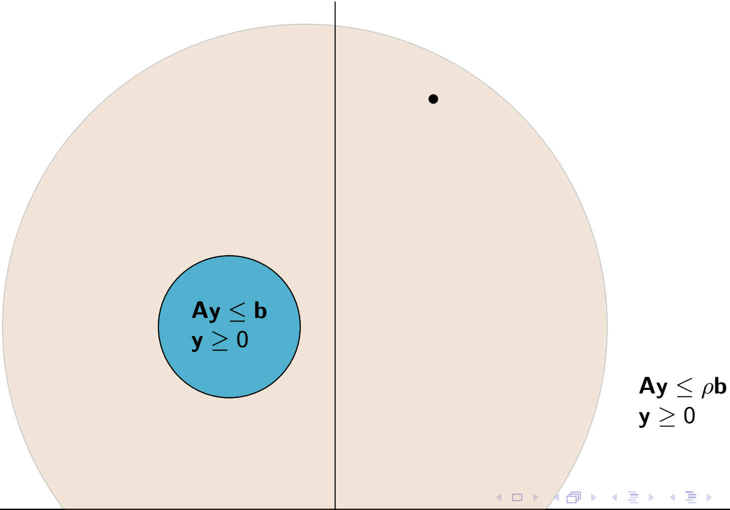


Multiplicative Weights Method: Basic Version

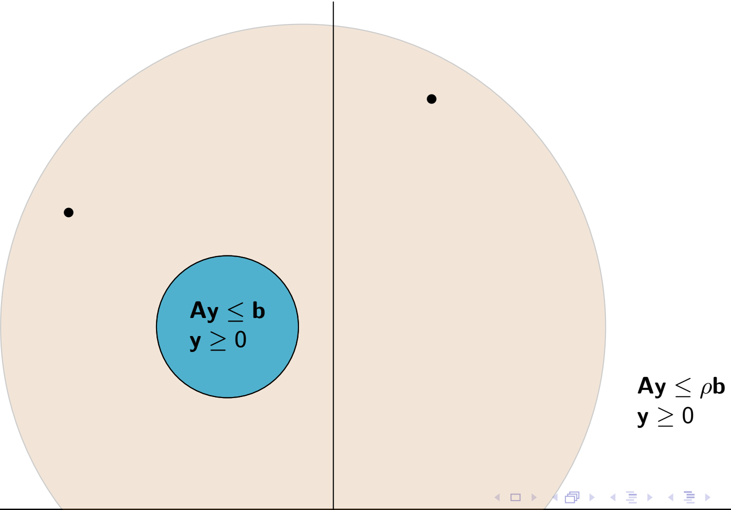
Initially $\mathbf{u} = \mathbf{1}$. Assume $\mathbf{A}, \mathbf{b} \geq \mathbf{0}$.
If $\mathbf{A}_i\mathbf{y} < \mathbf{b}_i$: lower $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1-\epsilon)^{(\mathbf{b}_i - \mathbf{A}_i\mathbf{y})/\mathbf{b}_i\rho}$.
If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y} - \mathbf{b}_i)/\mathbf{b}_i\rho}$.
$(\approx \mathbf{u}_i \leftarrow \mathbf{u}_i e^{\epsilon(\mathbf{A}_i\mathbf{y} - \mathbf{b}_i)/\rho})$



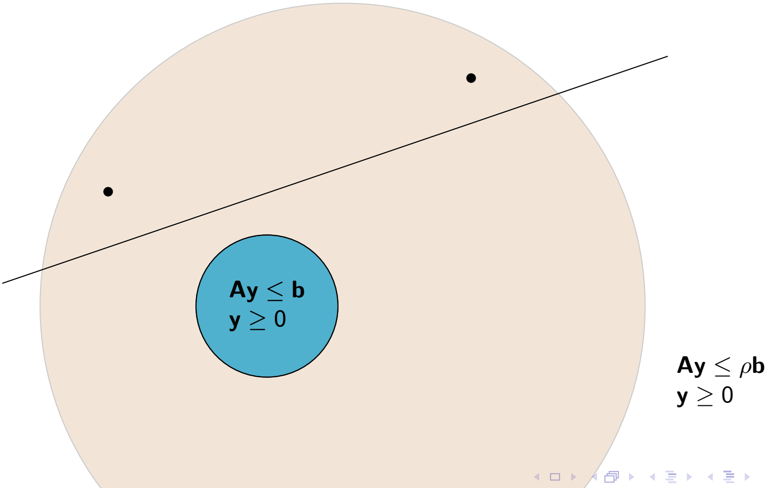
Multiplicative Weights Method: Basic Version



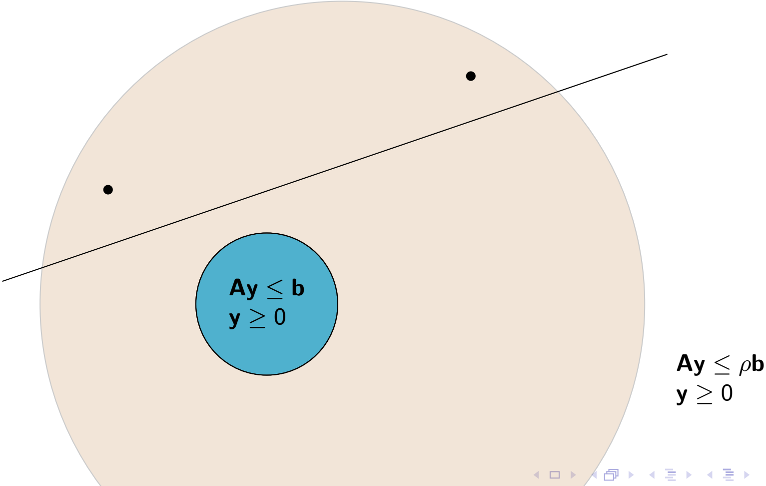
Multiplicative Weights Method: Basic Version



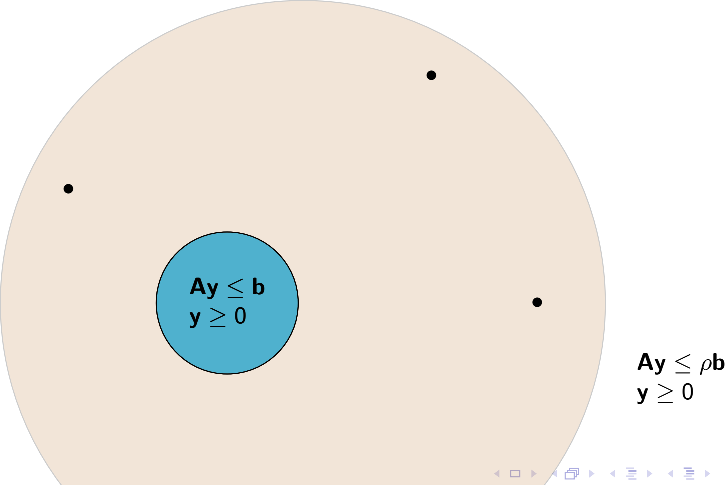
Multiplicative Weights Method: Basic Version



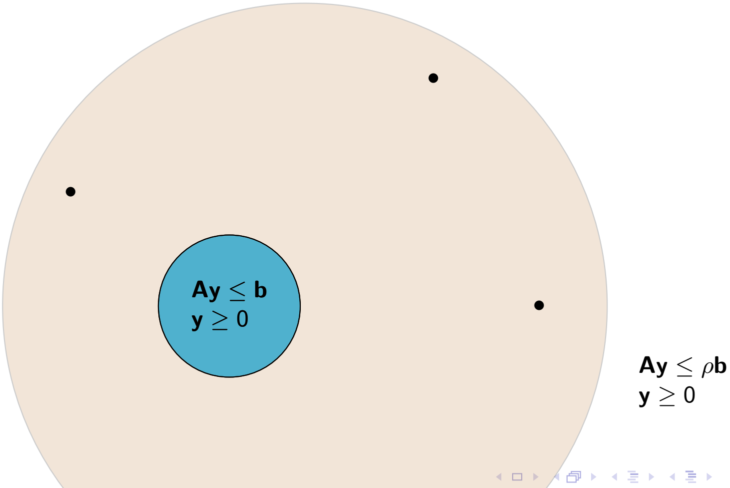
Multiplicative Weights Method: Basic Version



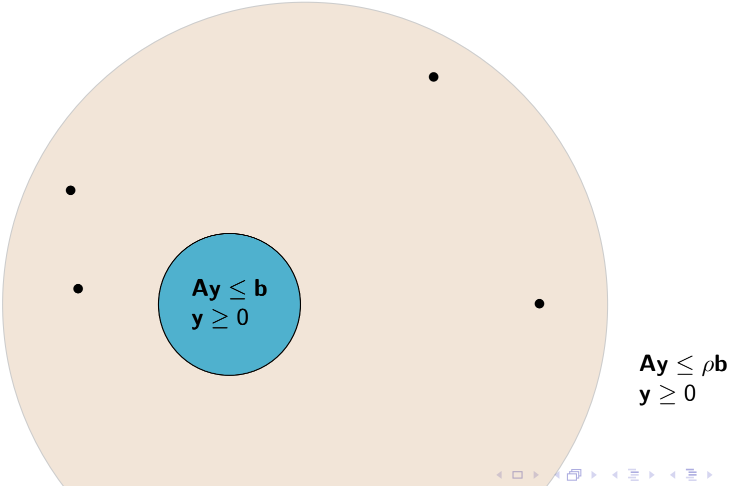
Multiplicative Weights Method: Basic Version



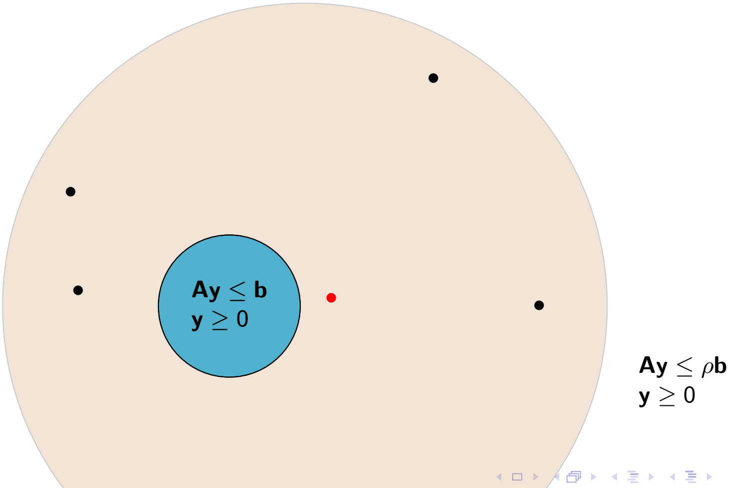
Multiplicative Weights Method: Basic Version



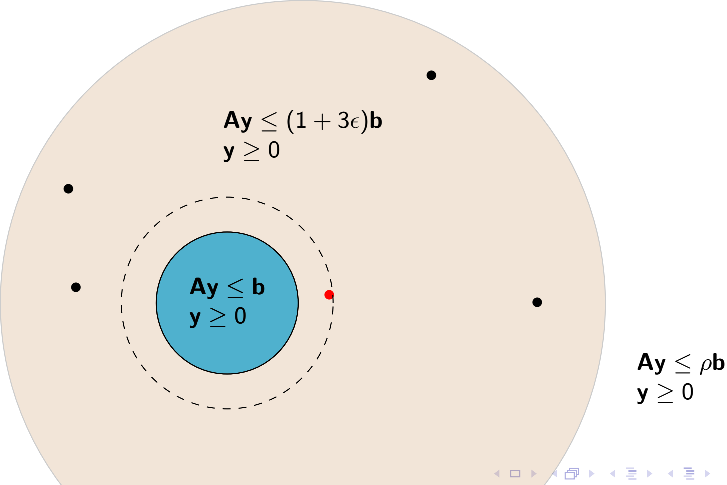
Multiplicative Weights Method: Basic Version



Multiplicative Weights Method: Basic Version

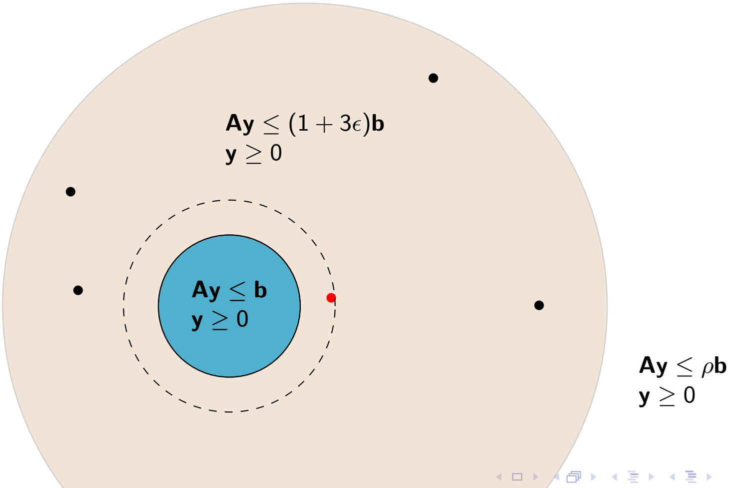


Multiplicative Weights Method: Basic Version



Multiplicative Weights Method: Basic Version

Number of rounds depends on ρ, ϵ and other specifics of updating **u**.
$\rho =$**width**.



How does the proof work?

Scale RHS to get $\mathbf{A}\mathbf{y} \leq 1$.

Let solution for iteration t be $\mathbf{y}(t)$, **assume** $-\rho \leq -\ell \leq \mathbf{A}_i\mathbf{y}(t) \leq \rho$.

"Violation" of constraint i as $V_i(\mathbf{y}(t)) = \mathbf{A}_i\mathbf{y}(t) - 1$; recall $\mathbf{u}_i(t+1) \approx \mathbf{u}_i(t)e^{\epsilon V_i(\mathbf{y}(t))/\rho}$.

"Average Violation" as $av(t) = \sum_i \frac{\mathbf{u}_i}{\sum_j \mathbf{u}_j} V_i(\mathbf{y}(t))$.
On the same side: ≤ 0 (easier case). For approximation $\leq \delta$.

"Potential" at iteration $t = \sum_i \mathbf{u}_i(t)$.
Now $\sum_i \mathbf{u}_i(t+1) \leq \left(\sum_i \mathbf{u}_i(t)\right) e^{\epsilon \ av(t)/\rho}$. Telescopes.

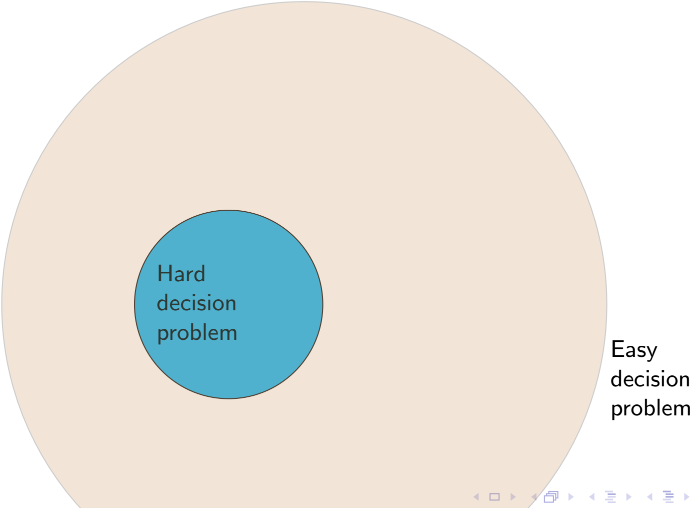
$$\ln \mathbf{u}_i(t) \leq \ln \frac{\text{Upper Bound}}{\text{Final Fractional wt of i}} + \frac{\epsilon}{\rho} \sum_t av(t)$$

$$\epsilon \sum_t V_i(t)/\rho - 2\epsilon^2 \ell T/\rho \leq \ln \frac{\text{Upper Bound}}{\text{Final Fractional wt of i}} + \frac{\epsilon}{\rho} \sum_t aV(t)$$

$$\sum_t V_i(t) \leq \cdots \leq \delta$$

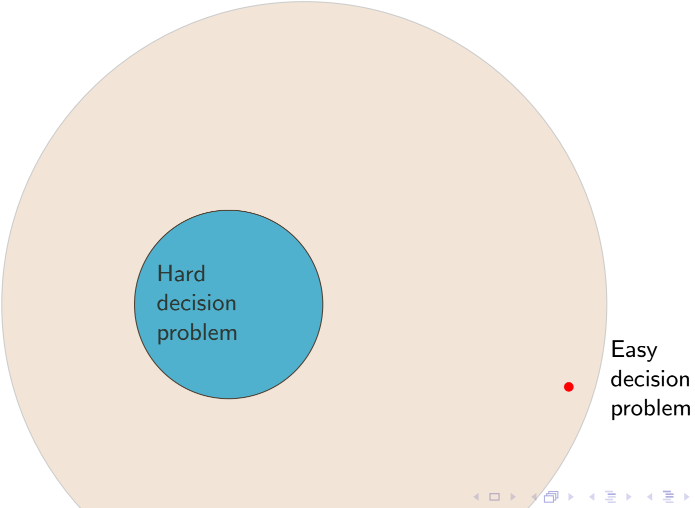
Dantzig Decompositions

A (weighted) running average view (primal space).



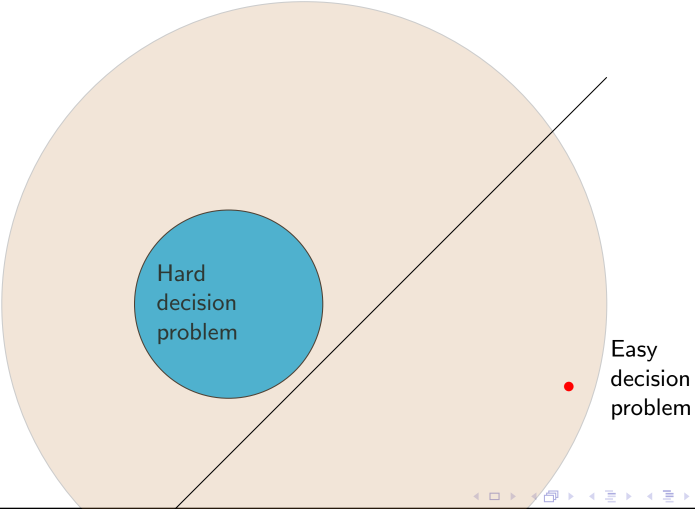
Dantzig Decompositions

A (weighted) running average view (primal space).



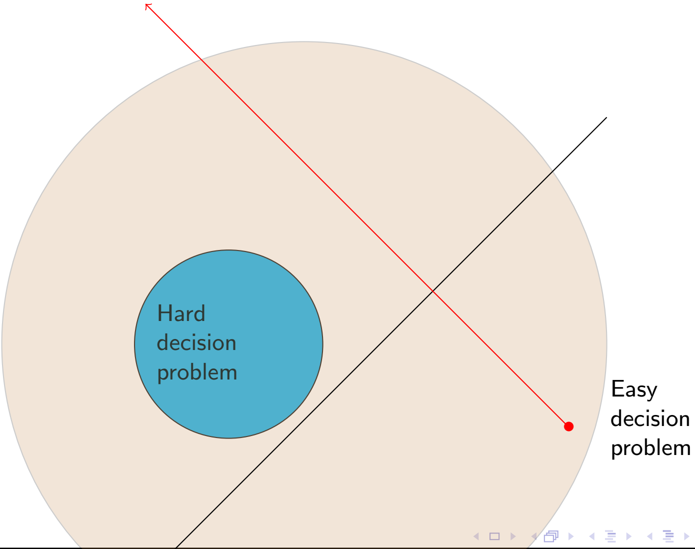
Dantzig Decompositions

A (weighted) running average view (primal space).



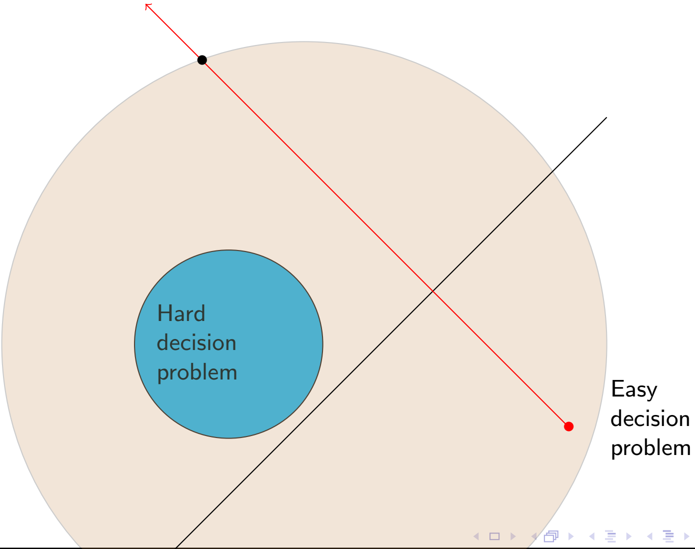
Dantzig Decompositions

A (weighted) running average view (primal space).



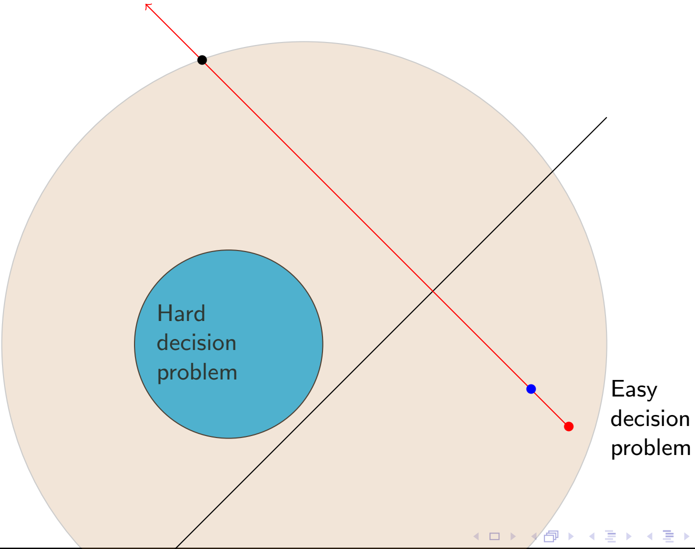
Dantzig Decompositions

A (weighted) running average view (primal space).



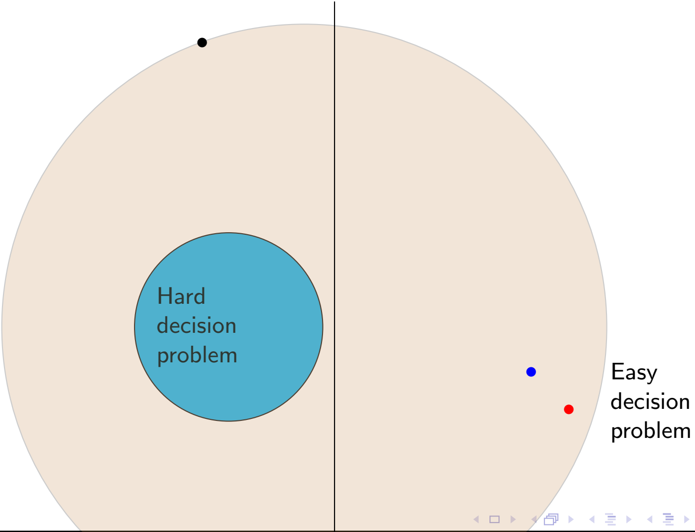
Dantzig Decompositions

A (weighted) running average view (primal space).



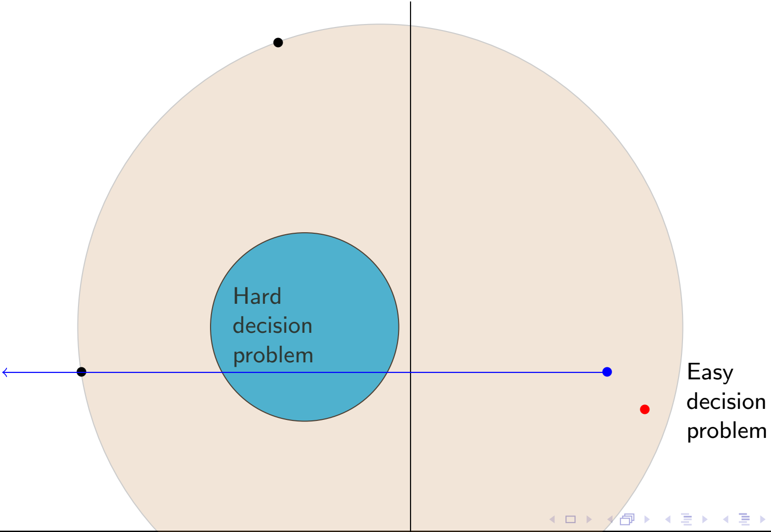
Dantzig Decompositions

A (weighted) running average view (primal space).



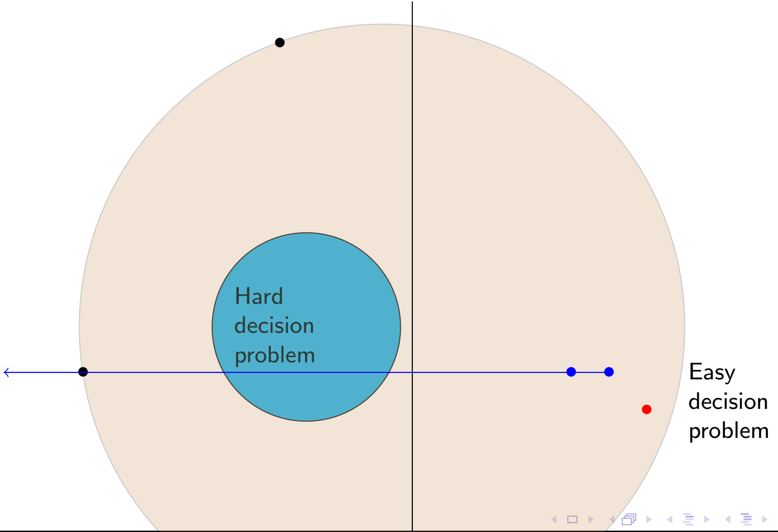
Dantzig Decompositions

A (weighted) running average view (primal space).



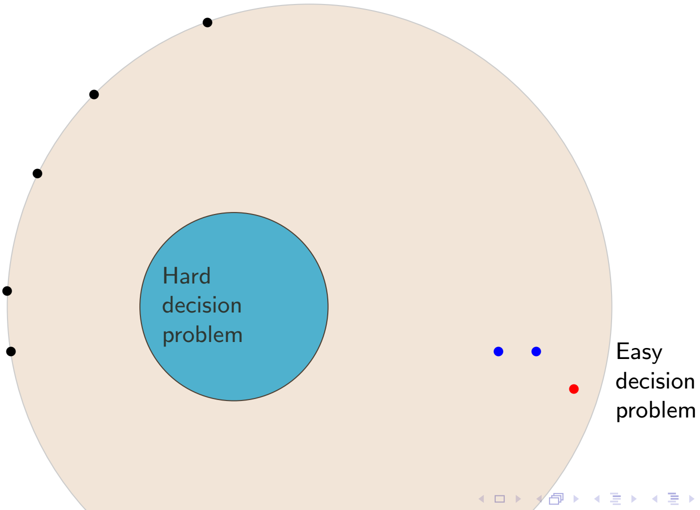
Dantzig Decompositions

A (weighted) running average view (primal space).



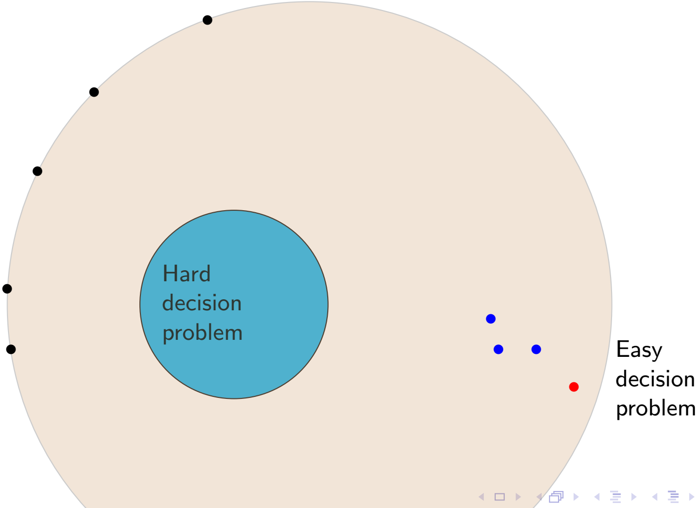
Dantzig Decompositions

A (weighted) running average view (primal space).



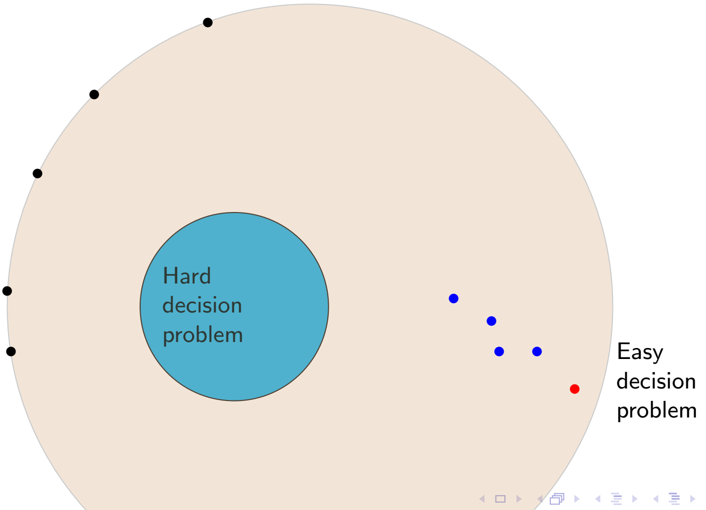
Dantzig Decompositions

A (weighted) running average view (primal space).



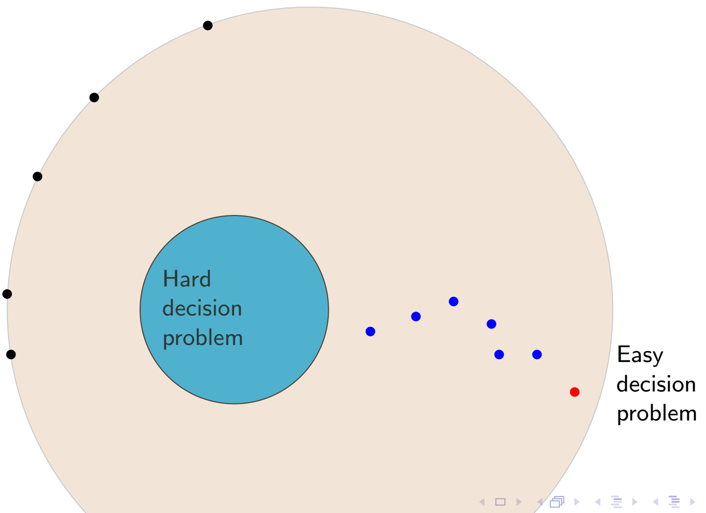
Dantzig Decompositions

A (weighted) running average view (primal space).



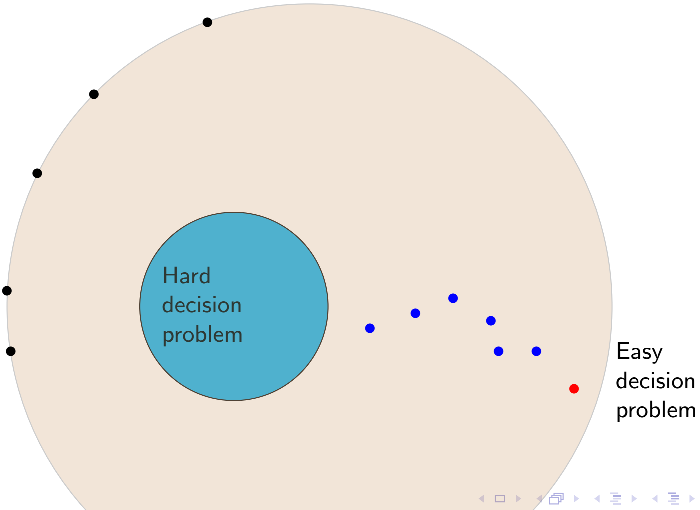
Dantzig Decompositions

A (weighted) running average view (primal space).



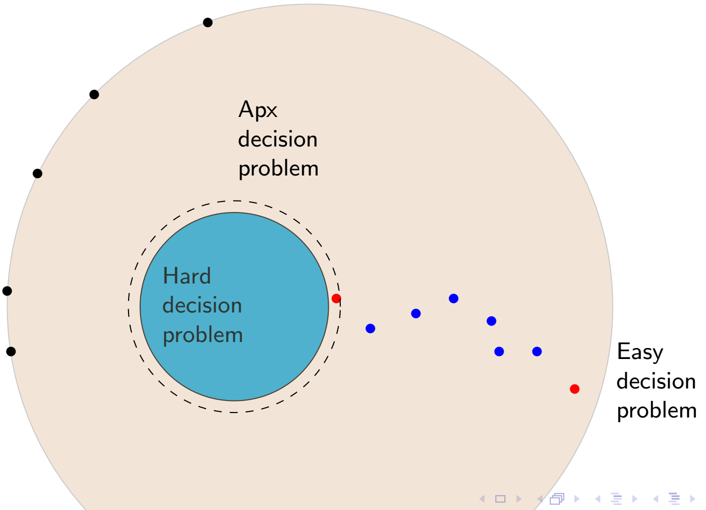
Dantzig Decompositions

A (weighted) running average view (primal space).



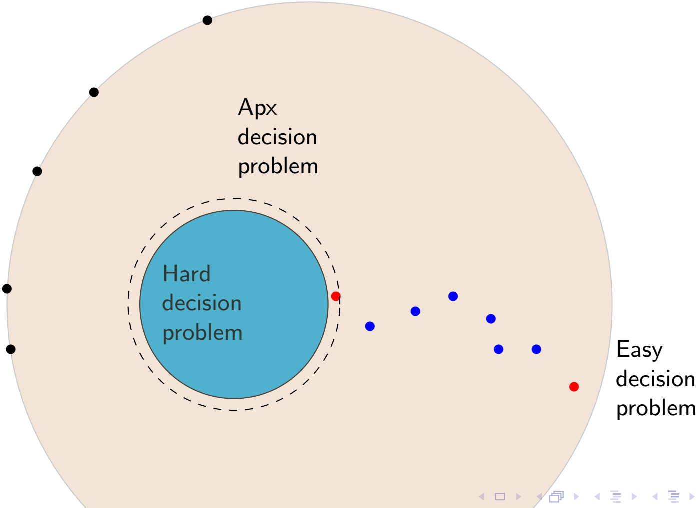
Dantzig Decompositions

A (weighted) running average view (primal space).



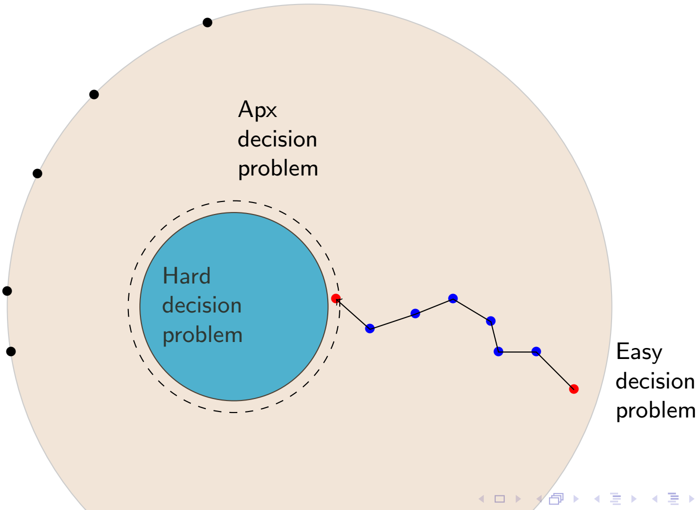
Dantzig Decompositions

A (weighted) running average view (primal space).



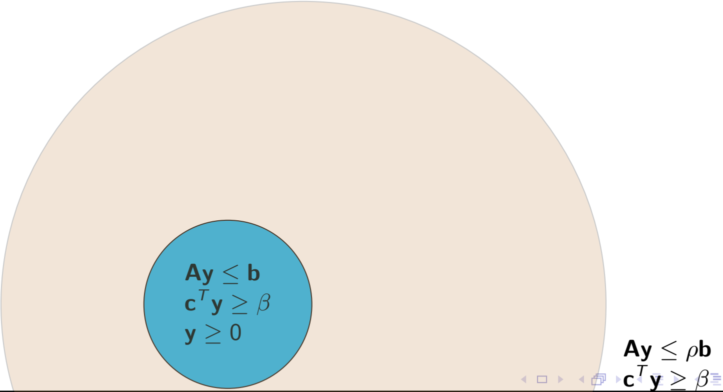
Dantzig Decompositions

A (weighted) running average view (primal space).



Multiplicative Weights: Optimization and Duals

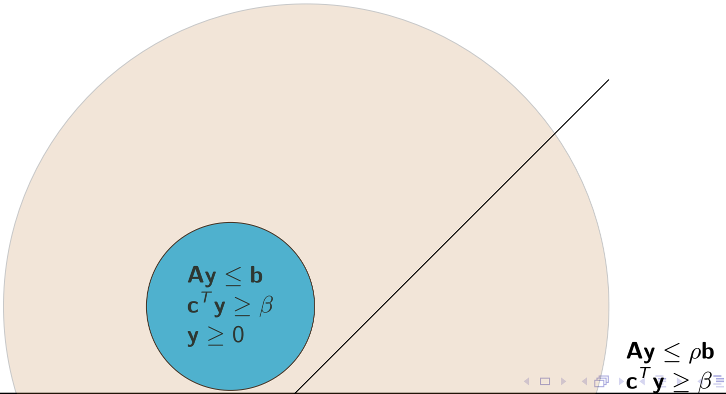
Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u



Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

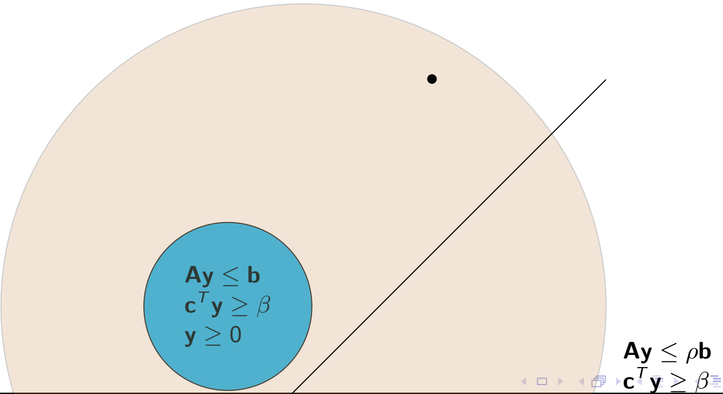
Dual of a hyperplane/constraint?
Dual of a point?



Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

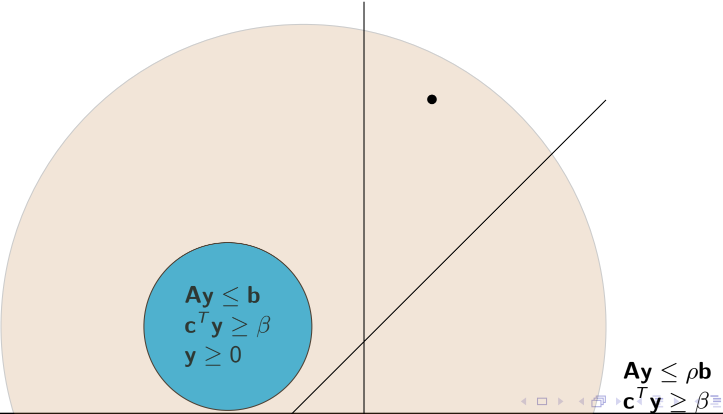
Dual of a hyperplane/constraint? Point in dual space.
Dual of a point?



Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

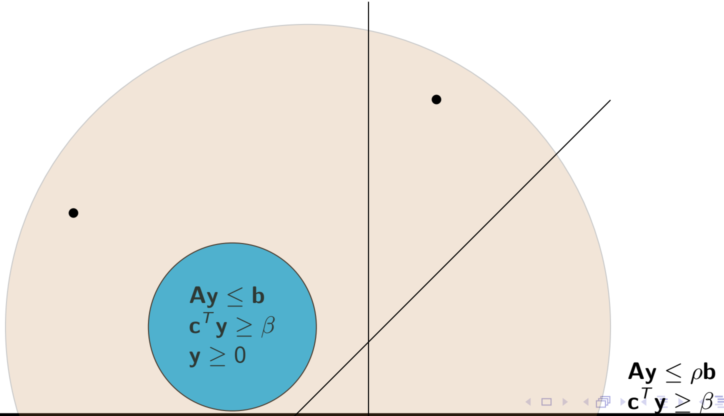
Dual of a hyperplane/constraint? Point in dual space.
Dual of a point? Hyperplane/constraint in dual space.



Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

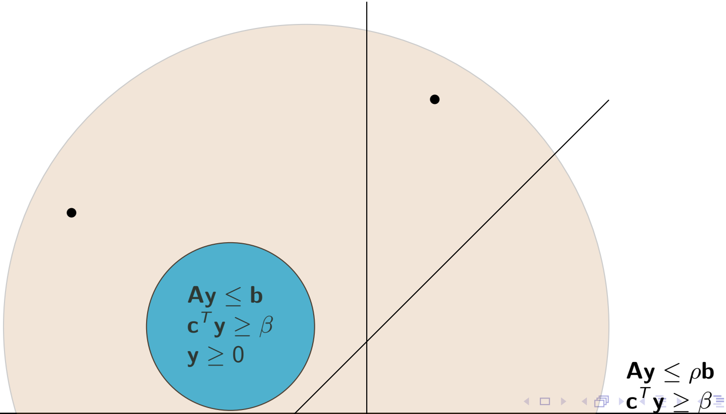
Dual of a hyperplane/constraint? Point in dual space.
Dual of a point? Hyperplane/constraint in dual space.



Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

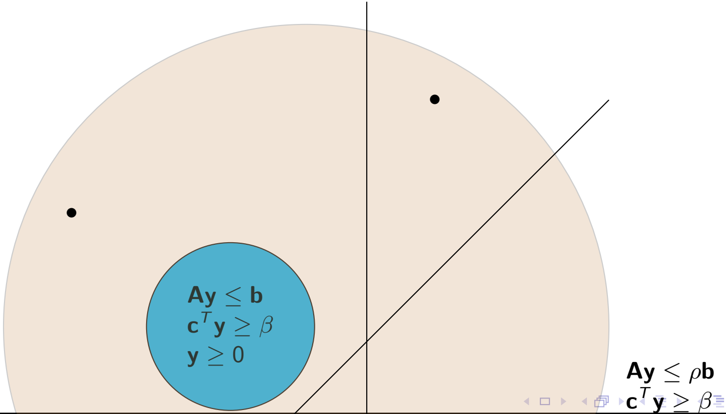
Dual of a hyperplane/constraint? Point in dual space.
Dual of a point? Hyperplane/constraint in dual space.



Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

Dual of a hyperplane/constraint? Point in dual space.
Dual of a point? Hyperplane/constraint in dual space.

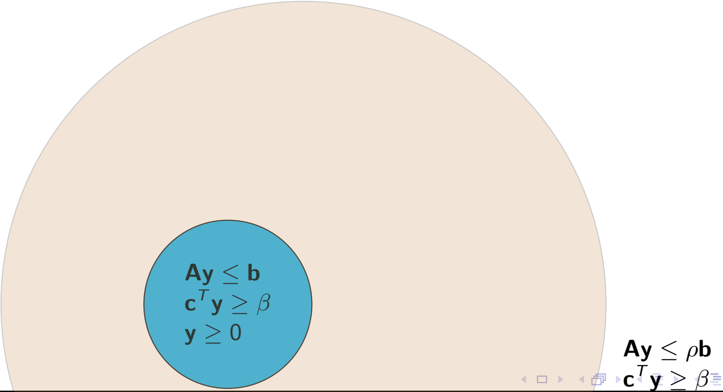


Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

Dual of a hyperplane/constraint? Point in dual space.
Dual of a point? Hyperplane/constraint in dual space.

Suppose we prove [*]: $\exists \mathbf{u}$ s.t. $\mathbf{A}^T\mathbf{u} \geq \mathbf{c}$ and $\rho\mathbf{b}^T\mathbf{u} < \beta$.

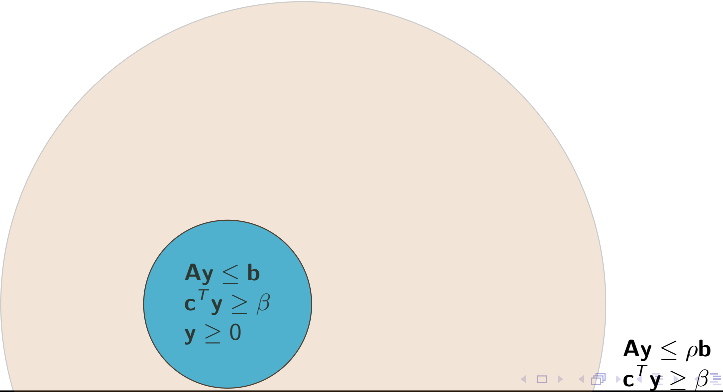


Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

Dual of a hyperplane/constraint? Point in dual space.
Dual of a point? Hyperplane/constraint in dual space.

Suppose we prove [*]: $\exists \mathbf{u}$ s.t. $\mathbf{A}^T\mathbf{u} \geq \mathbf{c}$ and $\rho\mathbf{b}^T\mathbf{u} < \beta$.
Providing a $\mathbf{y}$ corresponds to: we have not yet proved [*].



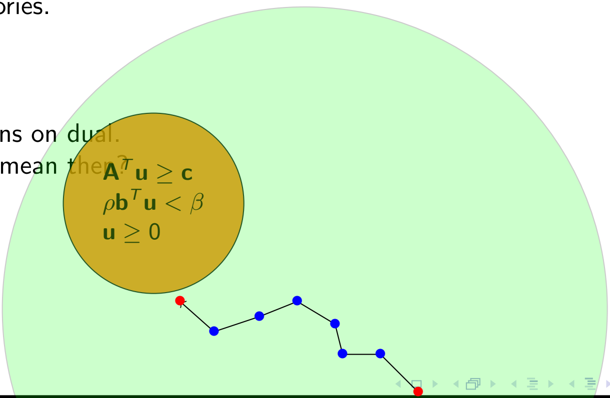
Multiplicative Weights: Optimization and Duals

Instead of tracking violations and averaging solutions at the end,
Consider the process from the perspective of u

Dual of a hyperplane/constraint? Point in dual space.
Dual of a point? Hyperplane/constraint in dual space.

Suppose we prove [*]: $\exists \mathbf{u}$ s.t. $\mathbf{A}^T\mathbf{u} \geq \mathbf{c}$ and $\rho \mathbf{b}^T \mathbf{u} < \beta$.
Providing a **y** corresponds to: we have not yet proved [*].
Think trajectories.

Decompositions on dual.
What does **y** mean then?



So the Dual or the Primal?

How do we choose which to start from?

Which set of constraints would you rather solve?

The one with more variables!
Lot more degrees of freedom.
Easier to approximate. Maybe sparse solutions exist.

Which set of constraints would you rather solve?

The one with more variables!
Lot more degrees of freedom.
Easier to approximate. Maybe sparse solutions exist.
Rewrite relaxations to introduce freedom!

(b) Application to Min. Correlation Clustering

Exponentially many constraints.

How to design an Oracle.

Drag and Drop application of Graph Sparsification/Sketching!

Correlation Clustering: Motivation

Tutorial in KDD 2014. Bonchi, Garcia-Soriano, Liberty.
Clustering of objects known only through relationships.
(Can have wide ranges of edge weights, +ve/-ve.)

Correlation Clustering: Motivation

Tutorial in KDD 2014. Bonchi, Garcia-Soriano, Liberty.
Clustering of objects known only through relationships.
(Can have wide ranges of edge weights, +ve/-ve.)

Consider an Entity Resolution example.

News arcticle 1: **Mr Smith** is devoted to mountain climbing. . . . **Mrs Smith** is a diver and said that she finds diving to be a sublime experience. . . . The goal is to reach new heights, said **Smith**.

Now consider a stream of such articles, with new as well as old entities.

Correlation Clustering: Motivation

Tutorial in KDD 2014. Bonchi, Garcia-Soriano, Liberty.
Clustering of objects known only through relationships.
(Can have wide ranges of edge weights, +ve/-ve.)

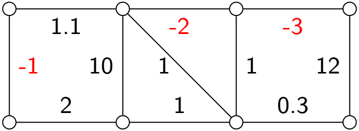
Consider an Entity Resolution example.

News arcticle 1: **Mr Smith** is devoted to mountain climbing. . . . **Mrs Smith** is a diver and said that she finds diving to be a sublime experience. . . . The goal is to reach new heights, said **Smith**.

Now consider a stream of such articles, with new as well as old entities.

Likely **Mr Smith** $\neq$ **Mrs Smith**. Large -ve weight.
The other references can be either. Small weights depending on context.
Weights are not a metric. Have a large range.

Correlation Clustering: A Formulation



Find a grouping that **disagrees** least with the graph.

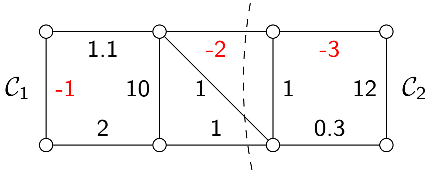
- Count +ve edges out of clusters. Count -ve edges **in** clusters.
- Use as many clusters as you like.

Alternatively we can find a grouping that **agrees** least.

NP Hard. Bansal Blum, Chawla, 04.

Many approximation algorithms are known. For many variants.
Approximations factors were known defore, will not focus on the factor.

Correlation Clustering: A Formulation



Find a grouping that **disagrees** least with the graph.

- Count +ve edges out of clusters. Count -ve edges **in** clusters.
- Use as many clusters as you like.

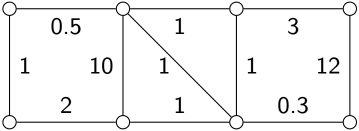
Alternatively we can find a grouping that **agrees** least.

NP Hard. Bansal Blum, Chawla, 04.

Many approximation algorithms are known. For many variants.
Approximations factors were known defore, will not focus on the factor.

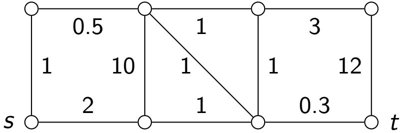
Global Sparsification: There and back again

Think of a problem on graph cuts.



Global Sparsification: There and back again

Think of a problem on graph cuts.

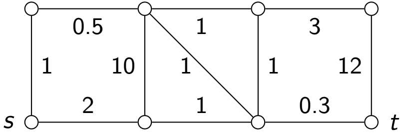


Min s-t Cut?

Sparsification preserves all cuts within $(1 \pm \epsilon)$.

Global Sparsification: There and back again

Think of a problem on graph cuts.

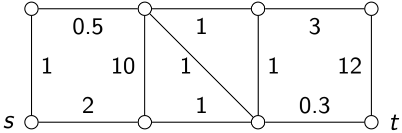


Min s-t Cut? Max s-t Cut? Max Cut?

Sparsification preserves all cuts within $(1 \pm \epsilon)$.

Global Sparsification: There and back again

Think of a problem on graph cuts.

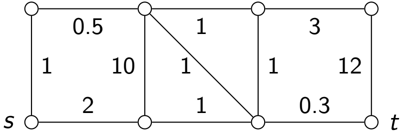


Min s-t Cut? Max s-t Cut? Max Cut? NP Hard. ≥ 0.5 apx uses SDPs.

Sparsification preserves all cuts within $(1 \pm \epsilon)$.

Global Sparsification: There and back again

Think of a problem on graph cuts.



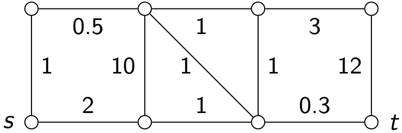
Min s-t Cut? Max s-t Cut? Max Cut? NP Hard. ≥ 0.5 apx uses SDPs.

Sparsification preserves all cuts within $(1 \pm \epsilon)$.

(a) Does not imply anything about finding specific cuts.

Global Sparsification: There and back again

Think of a problem on graph cuts.



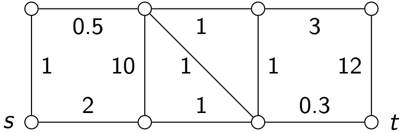
Min s-t Cut? Max s-t Cut? Max Cut? NP Hard. ≥ 0.5 apx uses SDPs.

Sparsification preserves all cuts within $(1 \pm \epsilon)$.

(a) Does not imply anything about finding specific cuts. Yet.

Global Sparsification: There and back again

Think of a problem on graph cuts.



Min s-t Cut? Max s-t Cut? Max Cut? NP Hard. ≥ 0.5 apx uses SDPs.

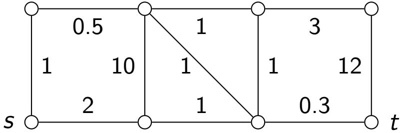
Sparsification preserves all cuts within $(1 \pm \epsilon)$.

(a) Does not imply anything about finding specific cuts. Yet.

(b) Does not obviously save space either!

Global Sparsification: There and back again

Think of a problem on graph cuts.



Min s-t Cut? Max s-t Cut? Max Cut? NP Hard. ≥ 0.5 apx uses SDPs.

Sparsification preserves all cuts within $(1 \pm \epsilon)$.

(a) Does not imply anything about finding specific cuts. Yet.

(b) Does not obviously save space either!

We will see examples both (a)–(b) and how to overcome them.
Lets return to correlation clustering.

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.

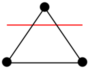
$$
\begin{aligned}
& \min \sum_{(i,j)\in E(+)} w_{ij}(1 - x_{ij}) + \sum_{(i,j)\in E(-)} |w_{ij}| x_{ij} & \\
& x_{ij} \leq 1 & \forall i, j \\
& x_{ij} \geq 0 & \forall i, j \\
& (1 - x_{ij}) + (1 - x_{jk}) \geq (1 - x_{ik}) & \forall i, j, k
\end{aligned}
$$

A linear program.

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.



$$
\begin{aligned}
&\min \sum_{(i,j)\in E(+)} w_{ij}(1 - x_{ij}) + \sum_{(i,j)\in E(-)} |w_{ij}| x_{ij} && \\
&x_{ij} \leq 1 && \forall i, j \\
&x_{ij} \geq 0 && \forall i, j \\
&(1 - x_{ij}) + (1 - x_{jk}) \geq (1 - x_{ik}) && \forall i, j, k
\end{aligned}
$$

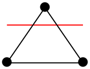
Triangle constraints

A linear program. $\Theta(n^3)$ Constraints, $\Theta(n^2)$ variables.

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.



$$
\begin{aligned}
\min \sum_{(i,j)\in E(+)} w_{ij}(1 - x_{ij}) + \sum_{(i,j)\in E(-)} |w_{ij}| x_{ij} & \\
x_{ij} \le 1 & \quad \forall i,j \\
x_{ij} \ge 0 & \quad \forall i,j \\
(1 - x_{ij}) + (1 - x_{jk}) \ge (1 - x_{ik}) & \quad \forall i,j,k
\end{aligned}
$$

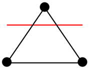
Triangle constraints

A linear program. $\Theta(n^3)$ Constraints, $\Theta(n^2)$ variables.
1 pass lower bound of $|E(-)|$ for any apx via Communication Complexity.

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.



$$\begin{aligned}
&\min \sum_{(i,j)\in E(+)} w_{ij}(1 - x_{ij}) + \sum_{(i,j)\in E(-)} |w_{ij}| x_{ij} && \\
&x_{ij} \leq 1 && \forall i,j \\
&x_{ij} \geq 0 && \forall i,j \\
&(1 - x_{ij}) + (1 - x_{jk}) \geq (1 - x_{ik}) && \forall i,j,k
\end{aligned}$$

Triangle constraints

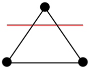
A linear program. $\Theta(n^3)$ Constraints, $\Theta(n^2)$ variables.
1 pass lower bound of $|E(-)|$ for any apx via Communication Complexity.

Sparsify $E(+)$, store $E(-)$? Will have $\tilde{O}(n) + |E(-)|$ variables.

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.



$$\begin{aligned} &\min \sum_{(i,j)\in E(+)} w_{ij}(1 - x_{ij}) + \sum_{(i,j)\in E(-)} |w_{ij}| x_{ij} && \\ &x_{ij} \le 1 && \forall i,j \\ &x_{ij} \ge 0 && \forall i,j \\ &(1 - x_{ij}) + (1 - x_{jk}) \ge (1 - x_{ik}) && \forall i,j,k \end{aligned}$$

Triangle constraints

A linear program. $\Theta(n^3)$ Constraints, $\Theta(n^2)$ variables.
1 pass lower bound of $|E(-)|$ for any apx via Communication Complexity.

Sparsify $E(+)$, store $E(-)$? Will have $\tilde{O}(n) + |E(-)|$ variables.

Does **not** work. The triangle constraints need all $\binom{n}{2}$ variables.

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Set $y_{ij} = 1 - x_{ij}$ for $+$ve edges. $z_{ij} = x_{ij}$ for -ve edges.

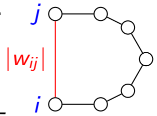
$$\begin{aligned} \min \quad & \sum_{(i,j)\in E(+)} w_{ij} y_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}| z_{ij} & \\ & y_{ij}, z_{ij} \geq 0 & \forall (i,j) \in E \\ & y_{ij}, z_{ij} ? & \end{aligned}$$

Sparsify $E(+)$. Store $E(-)$. $\Theta(n^2) \rightarrow \tilde{O}(n) + |E(-)|$ variables?
$\Theta(n^3)$ Constraints

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Set $y_{ij} = 1 - x_{ij}$ for $+$ve edges. $z_{ij} = x_{ij}$ for -ve edges.



$$\min \sum_{(i,j)\in E(+)} w_{ij} y_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}| z_{ij}$$

$$y_{ij}, z_{ij} \geq 0 \qquad \forall (i,j) \in E$$

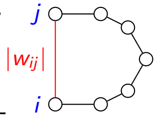
$$\sum_{(u,v)\in P(ij)} y_{uv} + z_{ij} \geq 1 \qquad \forall i, j, \text{and } i\text{-}j \text{ path } P(ij)$$

Sparsify $E(+)$. Store $E(-)$. $\Theta(n^2) \rightarrow \tilde{O}(n) + |E(-)|$ variables.
$\Theta(n^3)$ Constraints $\rightarrow$ Exponentially many constraints!

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Set $y_{ij} = 1 - x_{ij}$ for +ve edges. $z_{ij} = x_{ij}$ for -ve edges.



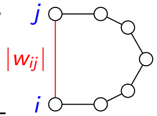
$$\begin{aligned} \min \quad & \sum_{(i,j)\in E(+)} w_{ij} y_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}| z_{ij} & \\ & y_{ij}, z_{ij} \geq 0 & \forall (i,j) \in E \\ & \sum_{(u,v)\in P(ij)} y_{uv} + z_{ij} \geq 1 & \forall i,j, \text{and } i\text{-}j \text{ path } P(ij) \end{aligned}$$

Sparsify $E(+)$. Store $E(-)$. $\Theta(n^2) \to \tilde{O}(n) + |E(-)|$ variables.
$\Theta(n^3)$ Constraints $\to$ Exponentially many constraints!
Solve LP (ellipsoid) & **Ball Growing**: Garg, Vazirani, Yannakakis 93.

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Set $y_{ij} = 1 - x_{ij}$ for $+$ve edges. $z_{ij} = x_{ij}$ for -ve edges.



$$\begin{aligned} \min \quad & \sum_{(i,j)\in E(+)} w_{ij} y_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}| z_{ij} & \\ & y_{ij}, z_{ij} \geq 0 & \forall (i,j) \in E \\ & \sum_{(u,v)\in P(ij)} y_{uv} + z_{ij} \geq 1 & \forall i,j, \text{and } i\text{-}j \text{ path } P(ij) \end{aligned}$$

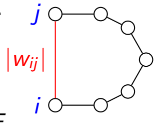
Sparsify $E(+)$. Store $E(-)$. $\Theta(n^2) \rightarrow \tilde{O}(n) + |E(-)|$ variables.
$\Theta(n^3)$ Constraints $\rightarrow$ Exponentially many constraints!
Solve LP (ellipsoid) & **Ball Growing**: Garg, Vazirani, Yannakakis 93.

MWM on the dual. $\tilde{O}(n + |E(-)|)$ space and $\tilde{O}(n^2)$ time.

Min Correlation Clustering

Equivalent to Max-Agreement at optimality. Not in approximation.

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Set $y_{ij} = 1 - x_{ij}$ for $+$ve edges. $z_{ij} = x_{ij}$ for -ve edges.



$$\min \sum_{(i,j)\in E(+)} w_{ij} y_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}| z_{ij}$$

$$y_{ij}, z_{ij} \geq 0 \qquad \forall (i,j) \in E$$

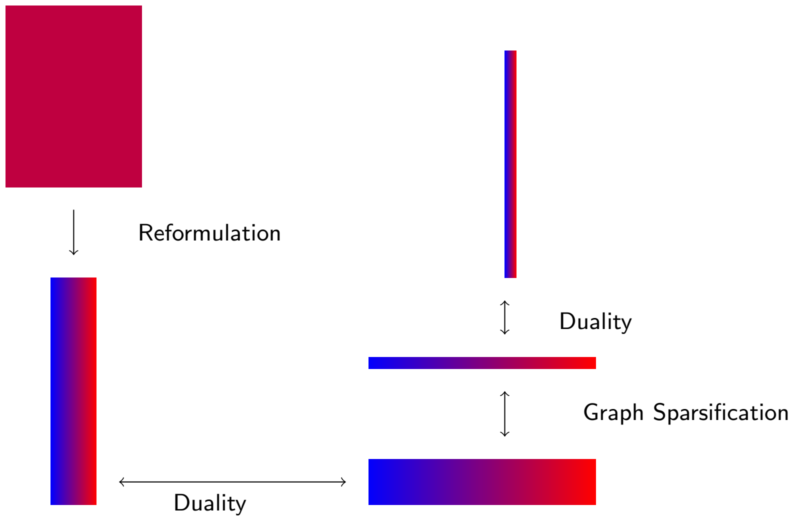
$$\sum_{(u,v)\in P(ij)} y_{uv} + z_{ij} \geq 1 \qquad \forall i,j, \text{and } i\text{-}j \text{ path } P(ij)$$

Sparsify $E(+)$. Store $E(-)$. $\Theta(n^2) \to \tilde{O}(n) + |E(-)|$ variables.
$\Theta(n^3)$ Constraints $\to$ Exponentially many constraints!
Solve LP (ellipsoid) & **Ball Growing**: Garg, Vazirani, Yannakakis 93.

MWM on the dual. $\tilde{O}(n + |E(-)|)$ space and $\tilde{O}(n^2)$ time.

Round infeasible primal (the running average). Success $\to$ done.
Failure $\to$ violated constraint(s) $\to$ point needed for MWM on Dual.

Algorithm in a Picture?



(c) SDPs and Max Correlation Clustering

Much more powerful than linear relaxations.

Recurring theme: Known relaxations will not fit.

New problem: What do we do to round?

Max-Agreement and SDPs

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Think of vector programming over unit length vectors. $x_{ij} = v_i \cdot v_j \leq 1$.

$$\begin{aligned}
\max \quad & \sum_{(i,j)\in E(+)} w_{ij}x_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}|(1 - x_{ij}) & \\
& x_{ii} = 1 & \forall i \\
& x_{ij} \geq 0 & \forall i,j \\
& \mathbf{x} \succeq \mathbf{0} &
\end{aligned}$$

Max-Agreement and SDPs

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Think of vector programming over unit length vectors. $x_{ij} = v_i \cdot v_j \leq 1$.

$$
\begin{array}{lll}
\max & \sum_{(i,j)\in E(+)} w_{ij}x_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}|(1 - x_{ij}) & \\
 & x_{ii} = 1 & \forall i \\
 & x_{ij} \geq 0 & \forall i, j \\
 & \mathbf{x} \succeq \mathbf{0} &
\end{array}
$$

MWM (in this context): Collection of constraints. Feasible set: $\mathcal{X}$.
Given $\mathbf{x}$ provide a real symmetric $\mathbf{A}$ (satisfying some **width** bounds)

(a) $\mathbf{A} \circ \mathbf{x} \leq b - \epsilon$, note $\mathbf{A} \circ \mathbf{x} = \sum_{i,j} A_{ij}x_{ij}$.

(b) $\mathbf{A} \circ \mathbf{x}' \geq b$ for all feasible $\mathbf{x}' \in \mathcal{X}$.

Max-Agreement and SDPs

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Think of vector programming over unit length vectors. $x_{ij} = v_i \cdot v_j \le 1$.

$$\begin{array}{lll} \max & \sum_{(i,j)\in E(+)} w_{ij}x_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}|(1-x_{ij}) & \\ & x_{ii} = 1 & \forall i \\ & x_{ij} \ge 0 & \forall i,j \\ & \mathbf{x} \succeq \mathbf{0} & \end{array}$$

MWM (in this context): Collection of constraints. Feasible set: $\mathcal{X}$.
Given $\mathbf{x}$ provide a real symmetric $\mathbf{A}$ (satisfying some **width** bounds)

(a) $\mathbf{A} \circ \mathbf{x} \le b - \epsilon$, note $\mathbf{A} \circ \mathbf{x} = \sum_{i,j} A_{ij}x_{ij}$.

(b) $\mathbf{A} \circ \mathbf{x}' \ge b$ for all feasible $\mathbf{x}' \in \mathcal{X}$.

Why??

Max-Agreement and SDPs

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Think of vector programming over unit length vectors. $x_{ij} = v_i \cdot v_j \leq 1$.

$$
\begin{array}{lll}
\max & \sum_{(i,j)\in E(+)} w_{ij}x_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}|(1 - x_{ij}) & \\
 & x_{ii} = 1 & \forall i \\
 & x_{ij} \geq 0 & \forall i,j \\
 & \mathbf{x} \succeq \mathbf{0} &
\end{array}
$$

MWM (in this context): Collection of constraints. Feasible set: $\mathcal{X}$.
Given $\mathbf{x}$ provide a real symmetric $\mathbf{A}$ (satisfying some **width** bounds)

(a) $\mathbf{A} \circ \mathbf{x} \leq b - \epsilon$, note $\mathbf{A} \circ \mathbf{x} = \sum_{i,j} A_{ij}x_{ij}$.

(b) $\mathbf{A} \circ \mathbf{x}' \geq b$ for all feasible $\mathbf{x}' \in \mathcal{X}$.

Why.

Max-Agreement and SDPs

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Think of vector programming over unit length vectors. $x_{ij} = v_i \cdot v_j \leq 1$.

$$
\begin{aligned}
\beta \leq \quad & \sum_{(i,j)\in E(+)} w_{ij} x_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}|(1 - x_{ij}) & \\
& x_{ii} = 1 & \forall i \\
& x_{ij} \geq 0 & \forall i, j \\
& \mathbf{x} \succeq \mathbf{0} &
\end{aligned}
$$

MWM (in this context): Collection of constraints. Feasible set: $\mathcal{X}$.
Given $\mathbf{x}$ provide a real symmetric $\mathbf{A}$ (satisfying some **width** bounds)

(a) $\mathbf{A} \circ \mathbf{x} \leq b - \epsilon$, note $\mathbf{A} \circ \mathbf{x} = \sum_{i,j} A_{ij} x_{ij}$.

(b) $\mathbf{A} \circ \mathbf{x}' \geq b$ for all feasible $\mathbf{x}' \in \mathcal{X}$.

Why. Does not work (width is high).

Max-Agreement and SDPs

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Think of vector programming over unit length vectors. $x_{ij} = v_i \cdot v_j \leq 1$.

$$\begin{aligned} \beta \leq \quad & \sum_{(i,j)\in E(+)} w_{ij} x_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}| \cancel{(1 - x_{ij})} \frac{x_{ii} + x_{jj} - 2x_{ij}}{2} & \\ & x_{ii} = 1 & \forall i \\ & x_{ij} \geq 0 & \forall i,j \\ & \mathbf{x} \succeq \mathbf{0} & \end{aligned}$$

MWM (in this context): Collection of constraints. Feasible set: $\mathcal{X}$.
Given $\mathbf{x}$ provide a real symmetric $\mathbf{A}$ (satisfying some **width** bounds)

(a) $\mathbf{A} \circ \mathbf{x} \leq b - \epsilon$, note $\mathbf{A} \circ \mathbf{x} = \sum_{i,j} A_{ij} x_{ij}$.

(b) $\mathbf{A} \circ \mathbf{x}' \geq b$ for all feasible $\mathbf{x}' \in \mathcal{X}$.

Why. ~~Does not work (width is high).~~ Linear Space. Linear time. 0.76-apx

Max-Agreement and SDPs

$x_{ij} = 1$ if in same group, and 0 otherwise. $E(+/-) = +/-$ve edge sets.
Think of vector programming over unit length vectors. $x_{ij} = v_i \cdot v_j \leq 1$.

$$\beta \leq \sum_{(i,j)\in E(+)} w_{ij} x_{ij} + \sum_{(i,j)\in E(-)} |w_{ij}| \cancel{(1 - x_{ij})} \frac{x_{ii} + x_{jj} - 2x_{ij}}{2}$$

$$x_{ii} = 1 \qquad \forall i$$

$$x_{ij} \geq 0 \qquad \forall i,j$$

$$\mathbf{x} \succeq \mathbf{0}$$

MWM (in this context): Collection of constraints. Feasible set: $\mathcal{X}$.
Given $\mathbf{x}$ provide a real symmetric $\mathbf{A}$ (satisfying some **width** bounds)

(a) $\mathbf{A} \circ \mathbf{x} \leq b - \epsilon$, note $\mathbf{A} \circ \mathbf{x} = \sum_{i,j} A_{ij} x_{ij}$.

(b) $\mathbf{A} \circ \mathbf{x}' \geq b$ for all feasible $\mathbf{x}' \in \mathcal{X}$.

Why. ~~Does not work (width is high).~~ Linear Space. Linear time. 0.76-apx
Relaxation needs to be compatible with trajectory.
Single pass. Sparsify $E(+)$ and $E(-)$ separately.

(d) Multiple Passes I: Max Bipartite Matching

Optimization over fixed constraint matrices.
Columns revealed one at a time.

Use of Approximation Algorithms for speedup of convergence.

"Primal-Dual meets Primal-Dual".

MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$\begin{array}{llll} \max & \sum_{(i,j)} y_{ij} w_{ij} & & \\ & \sum_{j} y_{ij} & \leq 1 & \forall i \\ & y_{ij} & \geq 0 & \forall (i,j) \end{array}$$

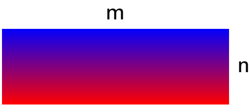
Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph. Different from online learning. Input itself is in small pieces.

MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$
\begin{aligned}
\max \quad & \sum_{(i,j)} y_{ij} w_{ij} & & \\
& \sum_{j} y_{ij} & \leq 1 & \quad \forall i \\
& y_{ij} & \geq 0 & \quad \forall (i,j)
\end{aligned}
$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph. Different from online learning. Input itself is in small pieces.

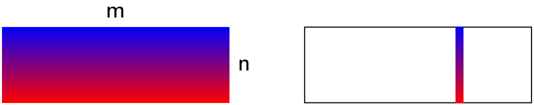


MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$
\begin{array}{llll}
\max & \sum_{(i,j)} y_{ij} w_{ij} & & \\
 & \sum_{j} y_{ij} & \leq 1 & \forall i \\
 & y_{ij} & \geq 0 & \forall (i,j)
\end{array}
$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph. Different from online learning. Input itself is in small pieces.

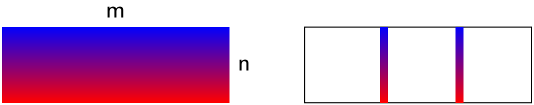


MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i, j) implies $\in E$.)

$$\begin{array}{llll} \max & \sum_{(i,j)} y_{ij} w_{ij} & & \\ & \sum_{j} y_{ij} & \leq 1 & \forall i \\ & y_{ij} & \geq 0 & \forall (i,j) \end{array}$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph. Different from online learning. Input itself is in small pieces.

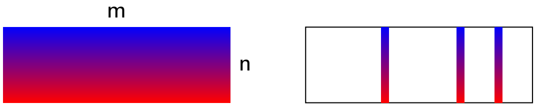


MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$\begin{aligned} \max \quad & \sum_{(i,j)} y_{ij} w_{ij} & & \\ & \sum_{j} y_{ij} & \leq 1 & \quad \forall i \\ & y_{ij} & \geq 0 & \quad \forall (i,j) \end{aligned}$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph.
Different from online learning. Input itself is in small pieces.

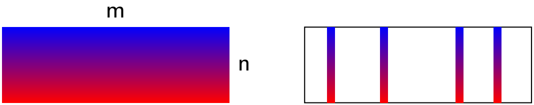


MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$
\begin{aligned}
\max \quad & \sum_{(i,j)} y_{ij} w_{ij} & & \\
& \sum_{j} y_{ij} & \leq 1 & \quad \forall i \\
& y_{ij} & \geq 0 & \quad \forall (i,j)
\end{aligned}
$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph. Different from online learning. Input itself is in small pieces.

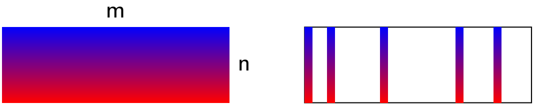


MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i, j) implies $\in E$.)

$$
\begin{array}{llll}
\max & \sum_{(i,j)} y_{ij} w_{ij} & & \\
 & \sum_{j} y_{ij} & \leq 1 & \forall i \\
 & y_{ij} & \geq 0 & \forall (i,j)
\end{array}
$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph.
Different from online learning. Input itself is in small pieces.

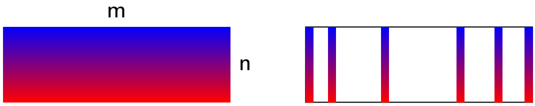


MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i, j) implies $\in E$.)

$$
\begin{aligned}
\max \quad & \sum_{(i,j)} y_{ij} w_{ij} & & \\
& \sum_{j} y_{ij} & \leq 1 \quad & \forall i \\
& y_{ij} & \geq 0 \quad & \forall (i, j)
\end{aligned}
$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph.
Different from online learning. Input itself is in small pieces.

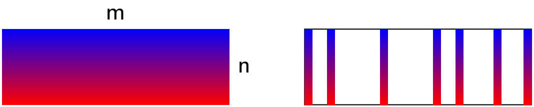


MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$\begin{array}{llll} \max & \sum_{(i,j)} y_{ij} w_{ij} & & \\ & \sum_{j} y_{ij} & \leq 1 & \forall i \\ & y_{ij} & \geq 0 & \forall (i,j) \end{array}$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph.
Different from online learning. Input itself is in small pieces.



MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$\begin{aligned}
\sum_{(i,j)} y_{ij} w_{ij} &\geq \beta \\
\sum_{j} y_{ij} &\leq 1 \quad \forall i \\
y_{ij} &\geq 0 \quad \forall (i,j)
\end{aligned}$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph.

Applying MWM: Point = candidate set of edges, in m-dim space.
Hyperplanes?

MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$
\begin{array}{llll}
 & \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta & \\
\mathbf{u}_i \rightarrow & \sum_{j} y_{ij} & \leq 1 & \forall i \\
 & y_{ij} & \geq 0 & \forall (i,j)
\end{array}
$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph.

Applying MWM: Point = candidate set of edges, in m-dim space.
Hyperplanes?

MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$\begin{array}{llll} & \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta & \\ \mathbf{u}_i \rightarrow & \sum_{j} y_{ij} & \leq 1 & \forall i \\ & y_{ij} & \geq 0 & \forall (i,j) \end{array}$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph.

Applying MWM: Point = candidate set of edges, in m-dim space.
Hyperplanes? $\sum_i u_i \sum_j y_{ij} \leq \sum_i u_i \quad \Leftrightarrow \quad \sum_{(i,j)} y_{ij}(u_i + u_j) \leq \sum_i u_i$.
Store & update $\mathbf{u}$. $O(n)$ storage.

MWM on Streams: Bipartite Matching

Integer and fractional optimums coincide. ($y_{ij} = y_{ji}$, (i,j) implies $\in E$.)

$$
\begin{array}{llll}
 & \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta & \\
\mathbf{u}_i \rightarrow & \sum_{j} y_{ij} & \leq 1 & \forall i \\
 & y_{ij} & \geq 0 & \forall (i,j)
\end{array}
$$

Streams: arbitrary list of m edges, $\ldots, \langle i, j, w_{ij} \rangle, \ldots$ for an n node graph.

Applying MWM: Point = candidate set of edges, in m-dim space.
Hyperplanes? $\sum_i u_i \sum_j y_{ij} \leq \sum_i u_i \quad \Leftrightarrow \quad \sum_{(i,j)} y_{ij}(u_i + u_j) \leq \sum_i u_i$.

$$
\text{Want:} \left\{
\begin{array}{lll}
\sum_{(i,j)} y_{ij}(u_i + u_j) \sum_i u_i & \leq \sum_i u_i & \\
\sum_{(i,j)} y_{ij} w_{ij} & \geq \beta & \\
\sum_j y_{ij} & \leq \rho & \forall i \\
y_{ij} & \geq 0 & \forall (i,j)
\end{array}
\right. .
$$

MWM on Streams: Bipartite Matching

$$\text{Want: } \begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

,

MWM on Streams: Bipartite Matching

Want: $$\begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Now $\exists \mathbf{y}, \ \forall \lambda \geq 0$ $$\begin{cases} \sum_{(i,j)} (w_{ij} - \lambda(u_i + u_j)) y_{ij} & \geq (\beta - \lambda \sum_i u_i) \\ \sum_j y_{ij} & \leq 1 \qquad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

MWM on Streams: Bipartite Matching

Want: $$\begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Now $\exists \mathbf{y}, \ \forall \lambda \geq 0$ $$\begin{cases} \sum_{(i,j)} (w_{ij} - \lambda(u_i + u_j)) y_{ij} & \geq (\beta - \lambda \sum_i u_i) \\ \sum_j y_{ij} & \leq 1 \qquad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Oracle(λ):

MWM on Streams: Bipartite Matching

Want: $$\begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Now $\exists \mathbf{y}, \ \forall \lambda \geq 0$ $$\begin{cases} \sum_{(i,j)} (w_{ij} - \lambda(u_i + u_j)) y_{ij} & \geq (\beta - \lambda \sum_i u_i) \\ \sum_j y_{ij} & \leq 1 \qquad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Oracle(λ):

- Seeing (i,j) compute $(w_{ij} - \lambda(u_i + u_j))$. If -ve, discard.

MWM on Streams: Bipartite Matching

$$\text{Want:} \begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

$$\textbf{Have } \mathbf{y}, \ \forall \lambda \geq 0 \begin{cases} \sum_{(i,j)} (w_{ij} - \lambda(u_i + u_j))\, y_{ij} & \geq (\beta - \lambda \sum_i u_i)/c \\ \sum_j y_{ij} & \leq 1 \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Oracle(λ):

- Seeing (i,j) compute $(w_{ij} - \lambda(u_i + u_j))$. If -ve, discard.
- Find a streaming $O(n)$ space c approximation on this filtered set.

MWM on Streams: Bipartite Matching

Want: $$\begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Have y,

$$\begin{cases} \sum_{(i,j)} (w_{ij} - \lambda(u_i + u_j))\, y_{ij} & \geq (\beta - \lambda \sum_i u_i)/c \\ \sum_j y_{ij} & \leq 1 \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Oracle(λ):

- Seeing (i,j) compute $(w_{ij} - \lambda(u_i + u_j))$. If -ve, discard.
- Find a streaming $O(n)$ space c approximation on this filtered set.

If Oracle(λ) for $\lambda = 0$ satisfies $\sum_{(i,j)} y_{ij}(u_i + u_j) \leq \sum_i u_i/c$ then we also have: $\sum_{(i,j)} w_{ij} y_{ij} \geq \beta/c$. **(easier case)**

MWM on Streams: Bipartite Matching

Want: $$\begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Have y, $$\begin{cases} \sum_{(i,j)} (w_{ij} - \lambda(u_i + u_j))\, y_{ij} & \geq (\beta - \lambda \sum_i u_i)/c \\ \sum_j y_{ij} & \leq 1 \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Oracle(λ):

- Seeing (i,j) compute $(w_{ij} - \lambda(u_i + u_j))$. If -ve, discard.
- Find a streaming $O(n)$ space c approximation on this filtered set.

For $\lambda = 0$ we have $\sum_{(i,j)} y_{ij}(u_i + u_j) \geq \sum_i u_i / c$.

For $\lambda = \sum_i u_i / \beta$ we have $\sum_{(i,j)} y_{ij}(u_i + u_j) \leq \sum_i u_i / c$. (Set $\mathbf{y} = 0$)

MWM on Streams: Bipartite Matching

Want: $\begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$

Have y, $\begin{cases} \sum_{(i,j)} (u_i + u_j) y_{ij} \leq \sum_i u_i / c \quad \text{and} \quad \sum_{(i,j)} w_{ij} y_{ij} \geq \beta / c \\ \sum_j y_{ij} \leq 1 \quad \forall i \\ y_{ij} \geq 0 \quad \forall (i,j) \end{cases}$

Oracle(λ):

- Seeing (i,j) compute $(w_{ij} - \lambda(u_i + u_j))$. If -ve, discard.
- Find a streaming $O(n)$ space c approximation on this filtered set.

For $\lambda = 0$ we have $\sum_{(i,j)} y_{ij}(u_i + u_j) \geq \sum_i u_i / c$.

For $\lambda = \sum_i u_i / \beta$ we have $\sum_{(i,j)} y_{ij}(u_i + u_j) \leq \sum_i u_i / c$. (Set $\mathbf{y} = 0$)

Binary search (or try values of λ in parallel).

MWM on Streams: Bipartite Matching

Want:
$$\begin{cases} \sum_{(i,j)} y_{ij}(u_i + u_j) & \leq \sum_i u_i \\ \sum_{(i,j)} y_{ij} w_{ij} & \geq \beta \\ \sum_j y_{ij} & \leq \rho \quad \forall i \\ y_{ij} & \geq 0 \quad \forall (i,j) \end{cases}$$

Have y,
$$\begin{cases} \sum_{(i,j)} (u_i + u_j) y_{ij} \leq \sum_i u_i / c \quad \text{and} \quad \sum_{(i,j)} w_{ij} y_{ij} \geq \beta / c \\ \sum_j y_{ij} \leq 1 \quad \forall i \\ y_{ij} \geq 0 \quad \forall (i,j) \end{cases}$$

Oracle(λ):

- Seeing (i,j) compute $(w_{ij} - \lambda(u_i + u_j))$. If -ve, discard.
- Find a streaming $O(n)$ space c approximation on this filtered set.

For $\lambda = 0$ we have $\sum_{(i,j)} y_{ij}(u_i + u_j) \geq \sum_i u_i / c$.

For $\lambda = \sum_i u_i / \beta$ we have $\sum_{(i,j)} y_{ij}(u_i + u_j) \leq \sum_i u_i / c$. (Set $\mathbf{y} = 0$)

Binary search (or try values of λ in parallel).

Multiply **y** by c. Set $\rho = c$ and we have a solution!

MWM based Bipartite Matching for Map-Reduce?

More general than streaming.

Map-Reduce based 8 approximations in $O(\log n)$ rounds exist, e.g., Lattanzi, Mosely, Suri, Vassilivitskii 11.

We can compose them. $O(\log n)$ rounds to get a c-approximation. Repeat $O(c\epsilon^{-2} \log n)$ times to get a $(1+\epsilon)$- fractional solution.

Can also round to an integral solution in small space.
A story for some other time.

(e) Multiple Passes II: Max Non-Bipartite Matching

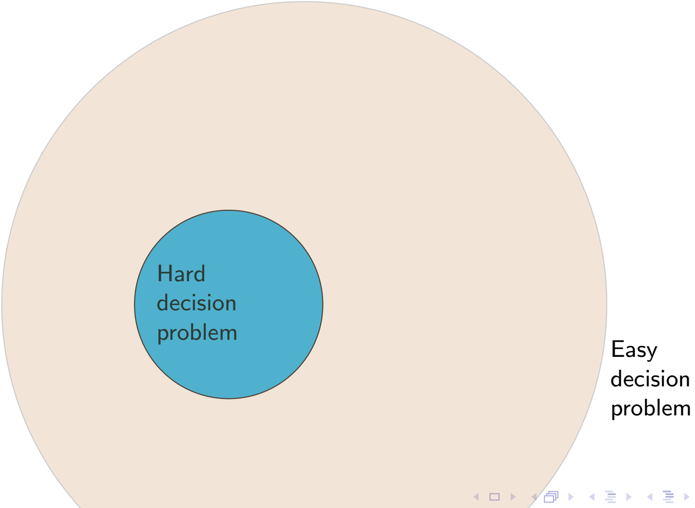
Exponentially many constraints.

Adaptive constraint sparsification. Perturbations.

How to find your way at night in the dark?

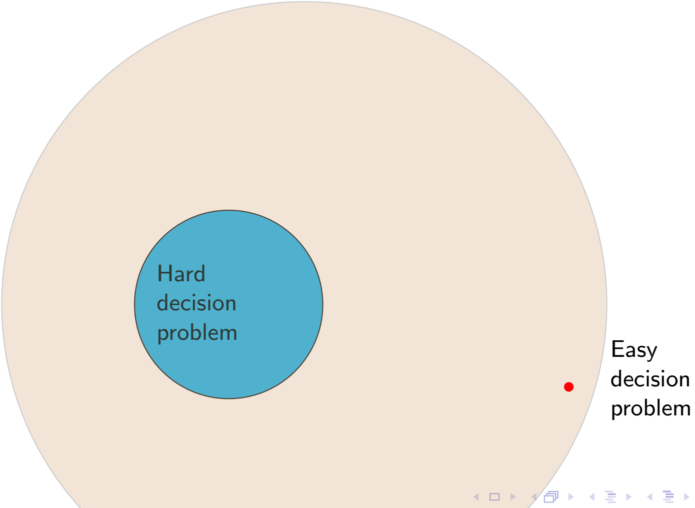
Revisiting Dantzig Decompositions

A running average view (primal space).



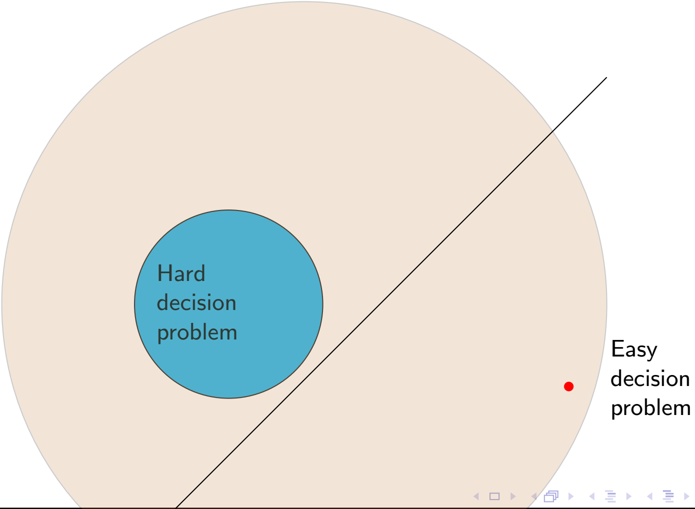
Revisiting Dantzig Decompositions

A running average view (primal space).



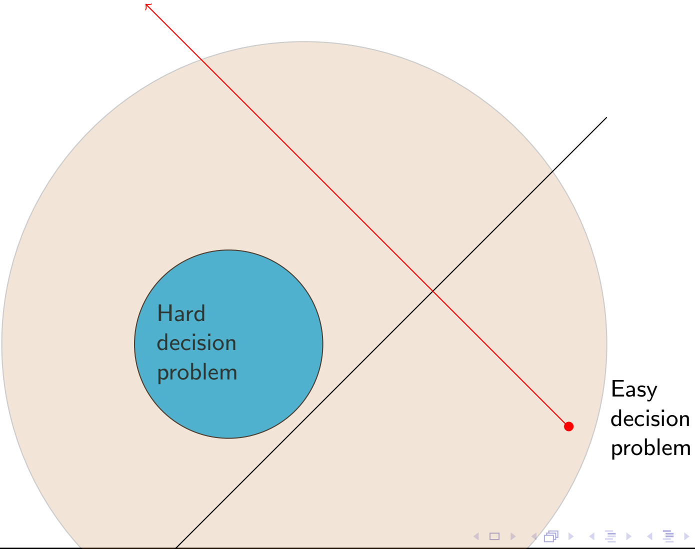
Revisiting Dantzig Decompositions

A running average view (primal space).



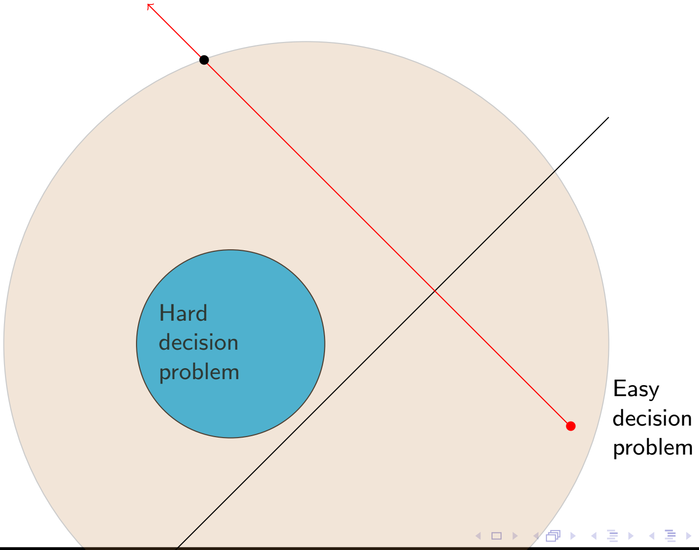
Revisiting Dantzig Decompositions

A running average view (primal space).



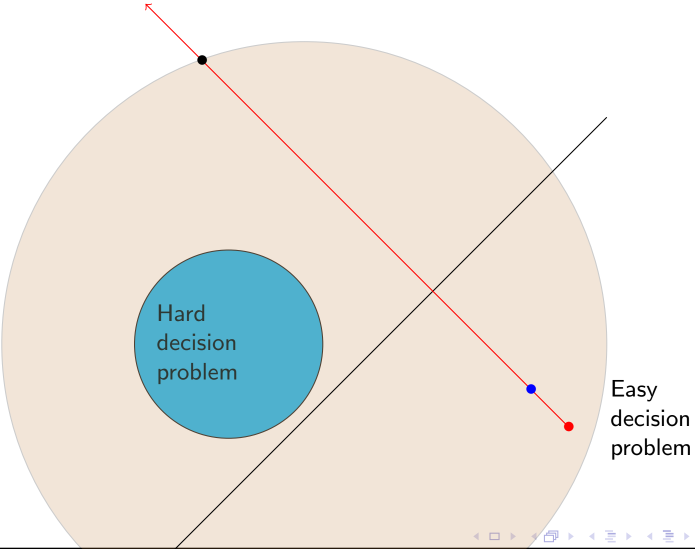
Revisiting Dantzig Decompositions

A running average view (primal space).



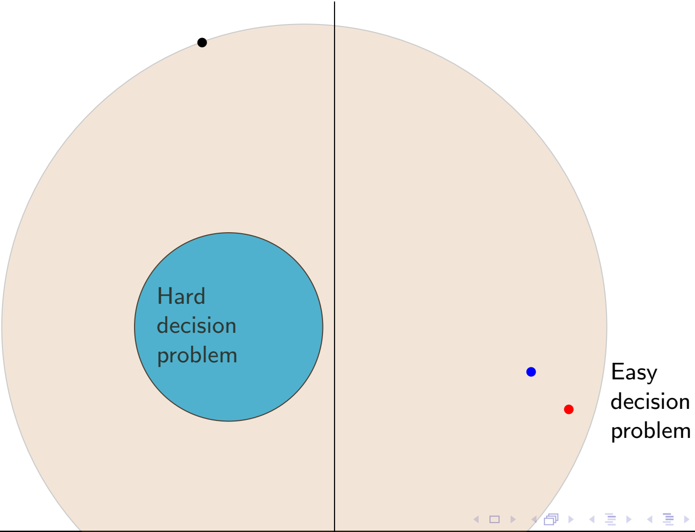
Revisiting Dantzig Decompositions

A running average view (primal space).



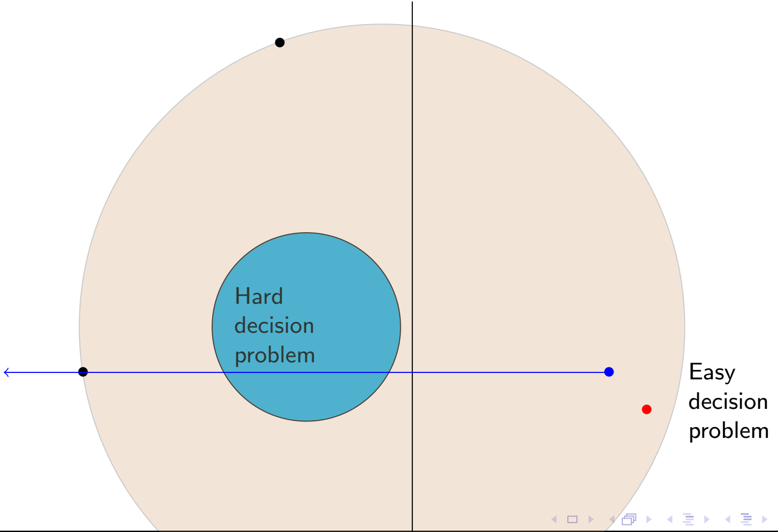
Revisiting Dantzig Decompositions

A running average view (primal space).



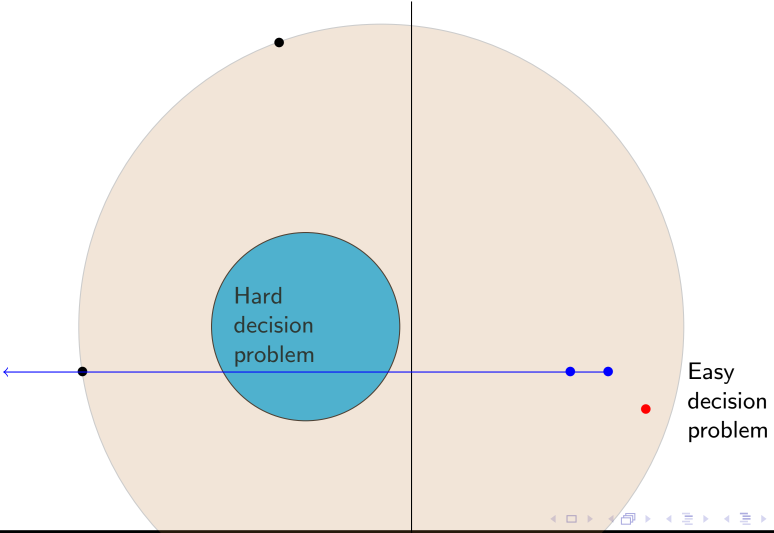
Revisiting Dantzig Decompositions

A running average view (primal space).



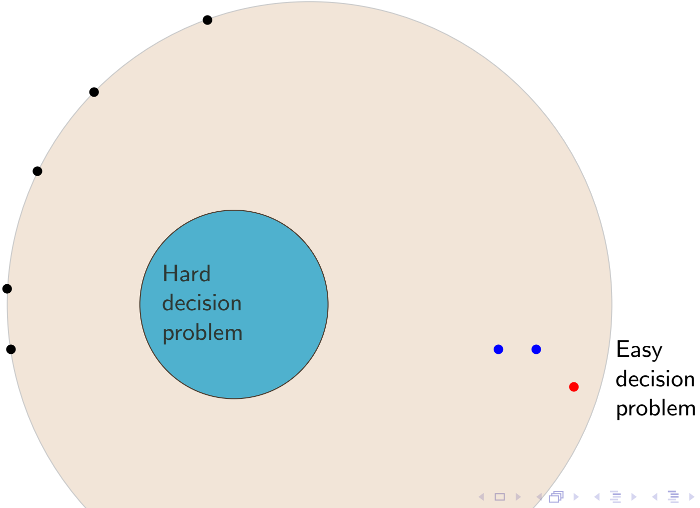
Revisiting Dantzig Decompositions

A running average view (primal space).



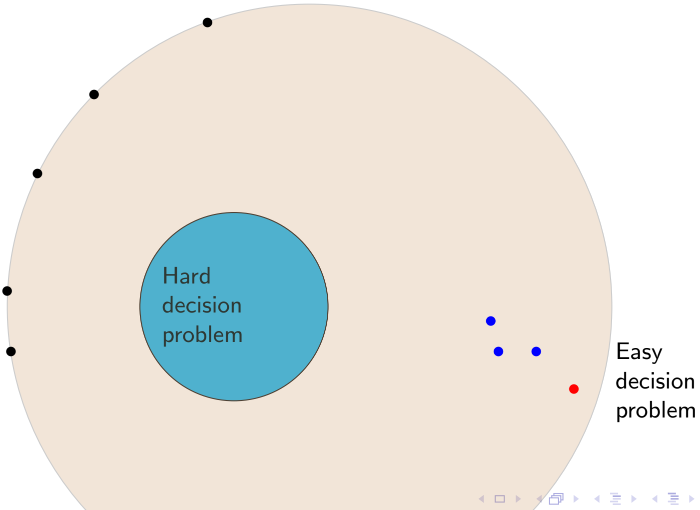
Revisiting Dantzig Decompositions

A running average view (primal space).



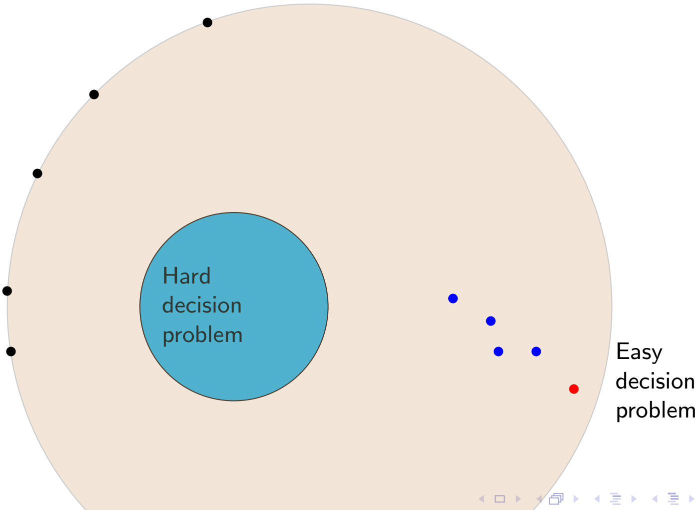
Revisiting Dantzig Decompositions

A running average view (primal space).



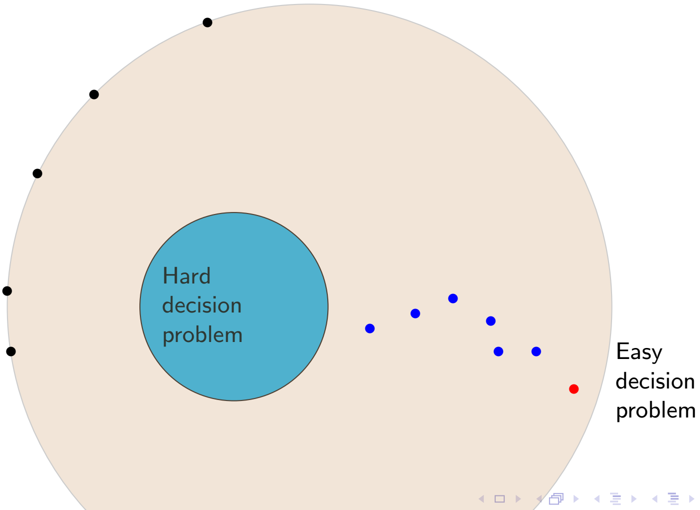
Revisiting Dantzig Decompositions

A running average view (primal space).



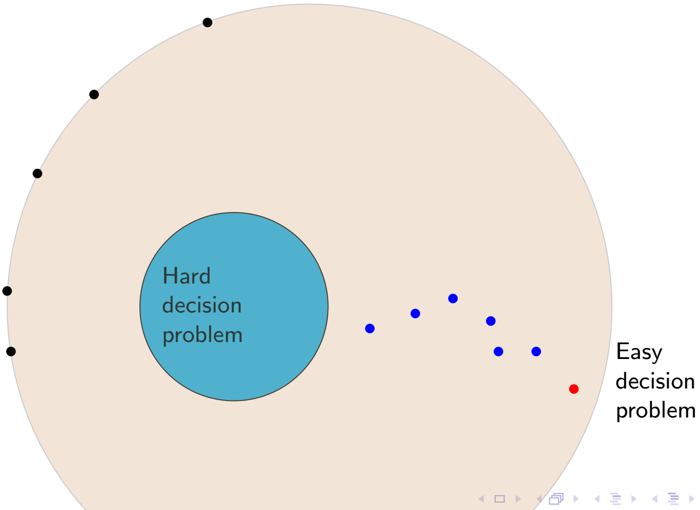
Revisiting Dantzig Decompositions

A running average view (primal space).



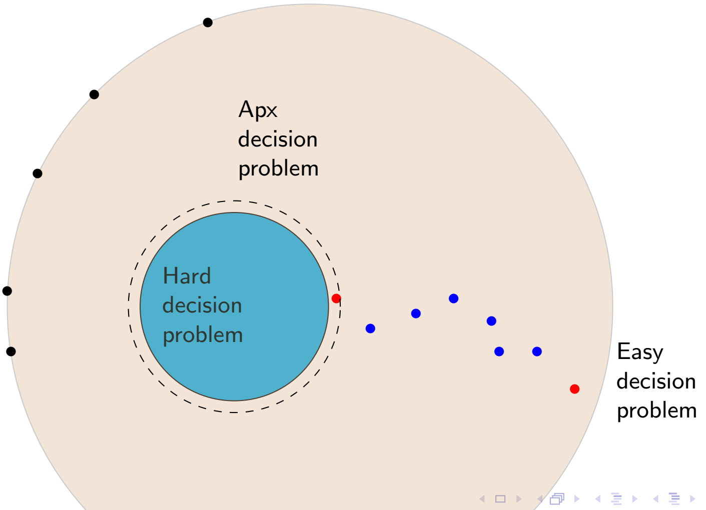
Revisiting Dantzig Decompositions

A running average view (primal space).



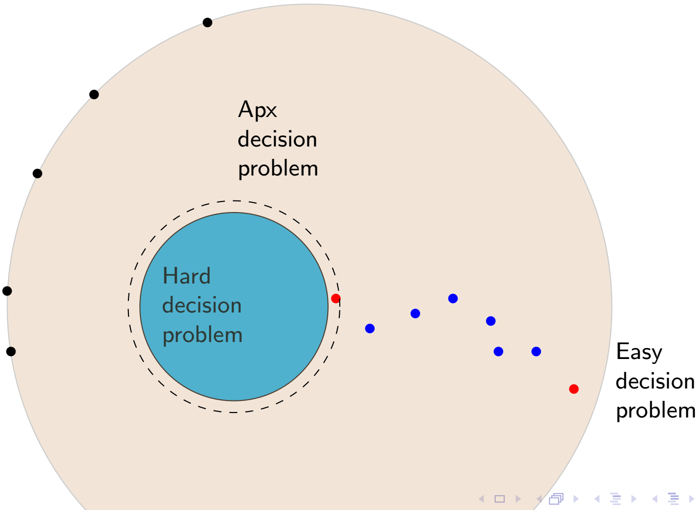
Revisiting Dantzig Decompositions

A running average view (primal space).



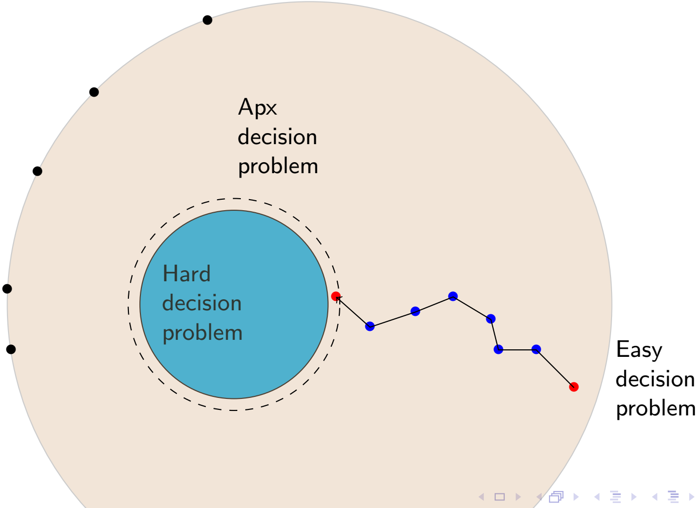
Revisiting Dantzig Decompositions

A running average view (primal space).



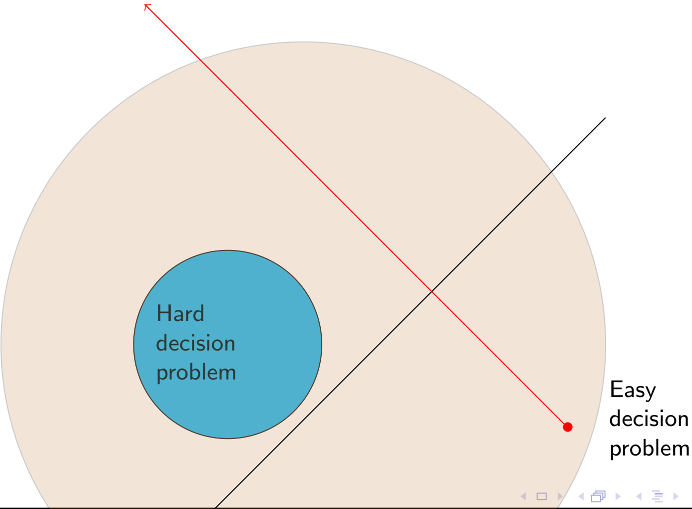
Revisiting Dantzig Decompositions

A running average view (primal space).



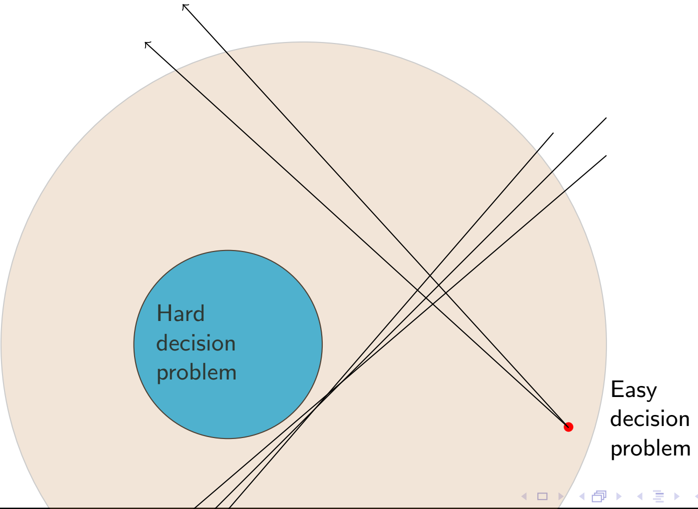
Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



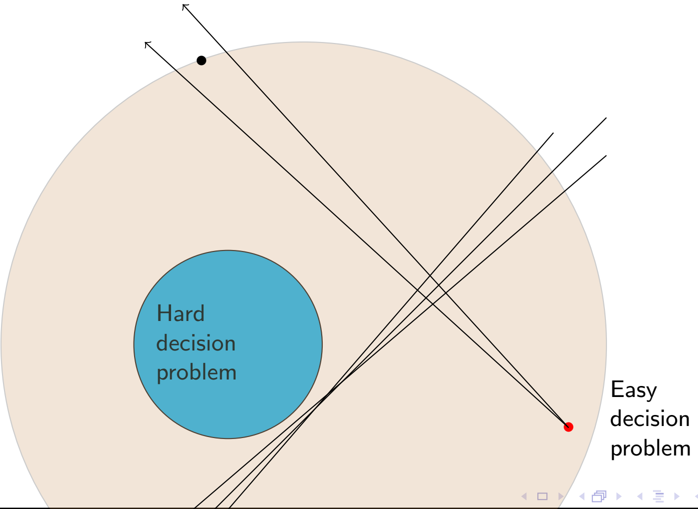
Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



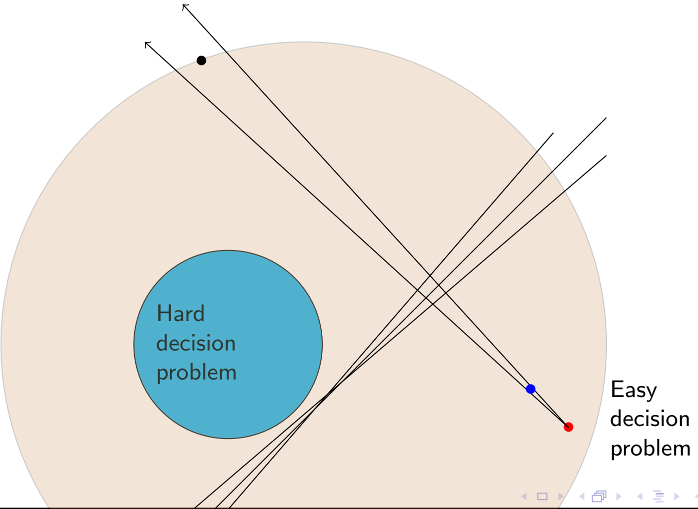
Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



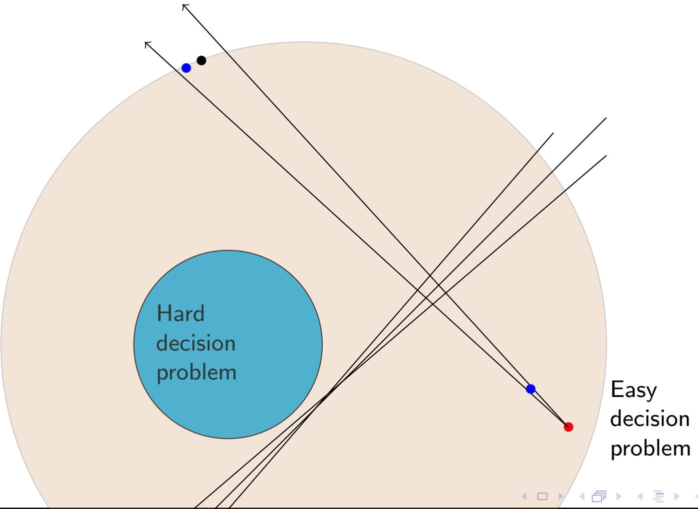
Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



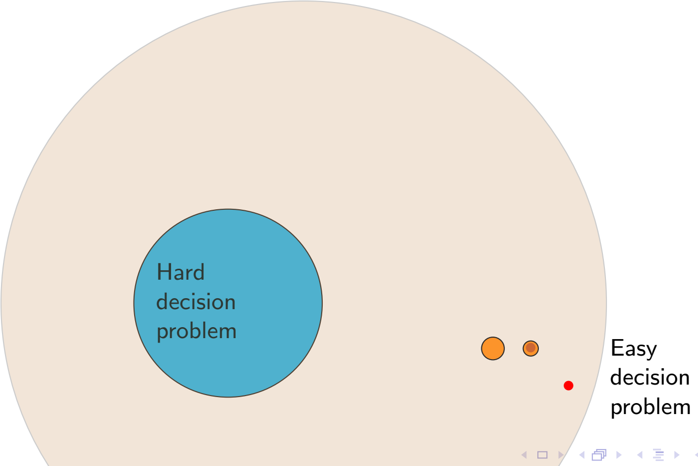
Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



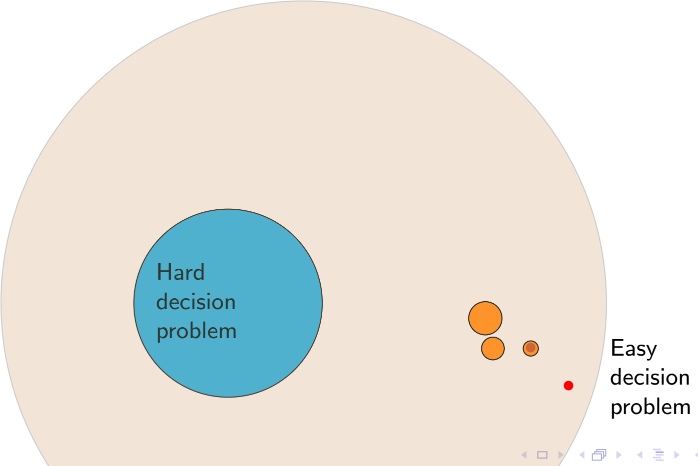
Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



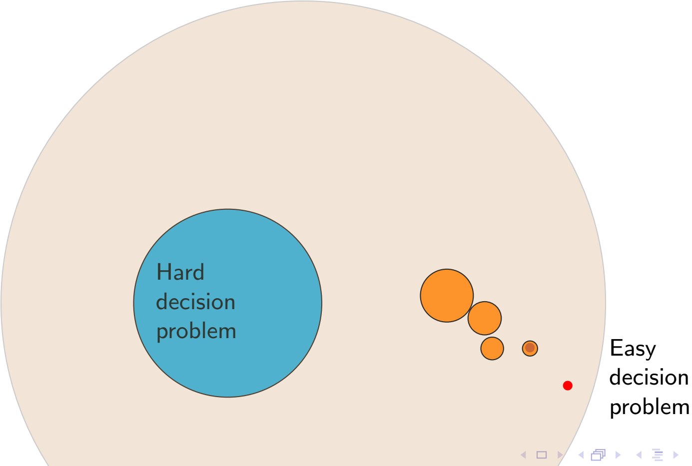
Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



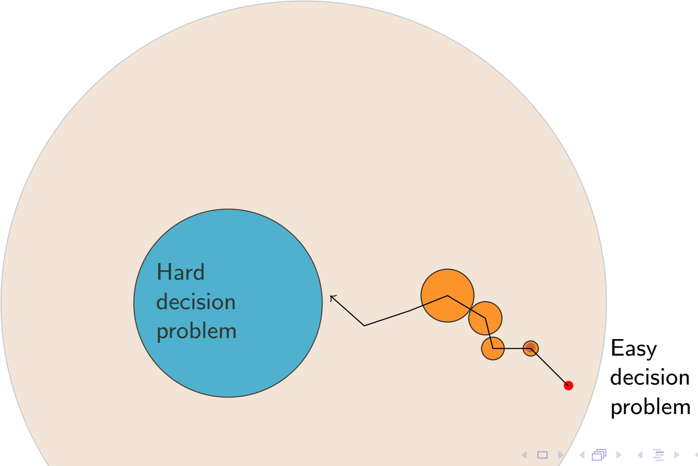
Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



Adaptive Sparsifications and Dantzig Decompositions

What if we sparsify $\mathbf{u}$? What does that mean?



Perturbations

Focus on the violations which are close to max violation.

Modify the polytope to find such violations faster.

Cuts and Constraints

Cuts and Constraints

(Again dropping $(i, j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$\beta^* = \max \quad \sum_{(i,j)} w_{ij} y_{ij}$$

$$\sum_j y_{ij} \quad \leq 1 \quad \forall i$$

$$y_{ij} \quad \geq 0$$

Cuts and Constraints

(Again dropping $(i,j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$\beta^* = \max \quad \sum_{(i,j)} w_{ij} y_{ij}$$

$$\sum_j y_{ij} \leq 1 \quad \forall i \quad \text{(Cut constraint!)}$$

$$y_{ij} \geq 0$$

Cuts and Constraints

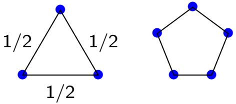
(Again dropping $(i,j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$
\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq 1 \quad \forall i \quad \textcolor{red}{\text{(Cut constraint!)}} \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

Cuts and Constraints

(Again dropping $(i,j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$
\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_{j} y_{ij} \quad & \leq 1 \quad \forall i \quad \text{(Cut constraint!)} \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

Rules out: 

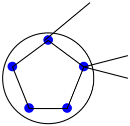
Cuts and Constraints

(Again dropping $(i,j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$
\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq 1 \quad \forall i \quad \text{(Cut constraint!)} \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

$$\sum_{i,j \in U} y_{ij} \leq \lfloor |U|/2 \rfloor \iff \sum_{i \in U} \left(\sum_j y_{ij} \right) - \left(\sum_{i \in U, j \notin U} y_{ij} \right) \leq 2 \lfloor |U|/2 \rfloor$$

$\sum_{i \in U, j \notin U} y_{ij} = Cut(U, V - U)$.



Cuts and Constraints

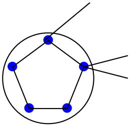
(Again dropping $(i,j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$
\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq 1 \quad \forall i \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

$$
\sum_{i,j \in U} y_{ij} \leq \lfloor |U|/2 \rfloor \quad \Longleftrightarrow \sum_{i \in U} \left(\sum_j y_{ij} \right) - \left(\sum_{i \in U, j \notin U} y_{ij} \right) \leq 2 \lfloor |U|/2 \rfloor
$$

$\sum_{i \in U, j \notin U} y_{ij} = Cut(U, V - U)$.

Find small cuts (with odd vertex sizes).



Cuts and Constraints

(Again dropping $(i,j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$
\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq 1 \quad \forall i \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

Find small cuts (with odd vertex sizes).

Cuts and Constraints

(Again dropping $(i,j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$
\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq 1 \quad \forall i \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

Find small cuts (with odd vertex sizes).
Standard Algorithm: Augment, contract blossoms, ... (many rounds).
Signature: feasible,. . . ,feasible (larger), . . . , feasible, (near) optimal

Cuts and Constraints

(Again dropping $(i,j) \in E$ in the subscripts, $y_{ij} = y_{ji}$.)

$$
\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq 1 \quad \forall i \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

Find small cuts (with odd vertex sizes).
Standard Algorithm: Augment, contract blossoms, ... (many rounds).
Signature: feasible,. . . ,feasible (larger), . . . , feasible, (near) optimal

Signature of this Algorithm:
infeasible,. . . ,infeasible (smaller), . . . , feasible, (near) optimal
Bipartite: $O(\epsilon^{-2} \log n)$ rounds
Non-Bipartite: $O(\epsilon^{-4} \log n)$ rounds

How?

$$
\begin{aligned}
\tilde{\beta} = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq (1 - 4\delta) \quad \forall i \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor - \tfrac{\delta^2 |U|^2}{4} \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

Consider two odd sets with "density" similar to the densest set.
Have to be disjoint or within each other (laminar)!
Reduces to a bipartite problem with different "effective weights".
Near linear time algorithm.

How?

$$\begin{aligned}
\tilde{\beta} = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq (1 - 4\delta) \quad \forall i \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor - \frac{\delta^2 |U|^2}{4} \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}$$

Consider two odd sets with "density" similar to the densest set.
Have to be disjoint or within each other (laminar)!
Reduces to a bipartite problem with different "effective weights".
Near linear time algorithm.
Extends to capacities on vertices and edges.

(f) Multiple Passes III: Non-Bipartite Matching

For a few passes less ...

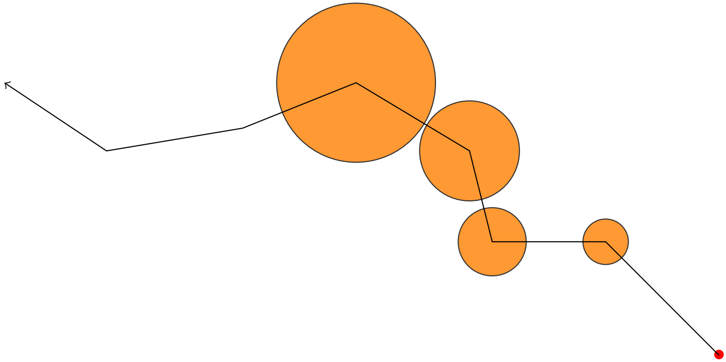
Sparsify non-adaptively in parallel; use sequentially.

Dual-primal versus primal-dual.

New relaxations for Matching.

Sparsify in Parallel, Use Sequentially

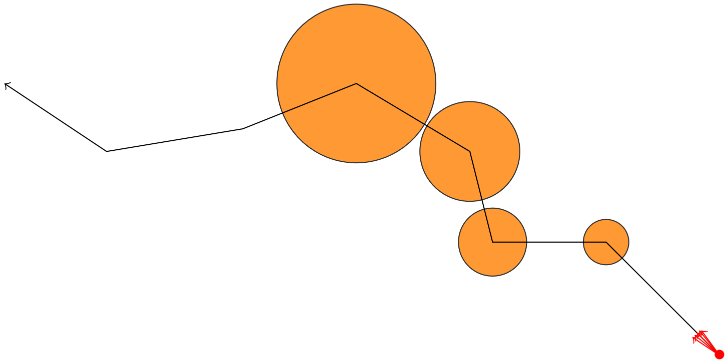
We saw a version of sketch in parallel, use sequentially in connectivity.
Question: Where will we be after 5 steps of MWM?
Recall: If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y}-\mathbf{b}_i)/\mathbf{b}_i\rho}$.



Sparsify in Parallel, Use Sequentially

We saw a version of sketch in parallel, use sequentially in connectivity.
Question: Where will we be after 5 steps of MWM?
Recall: If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y}-\mathbf{b}_i)/\mathbf{b}_i\rho}$.

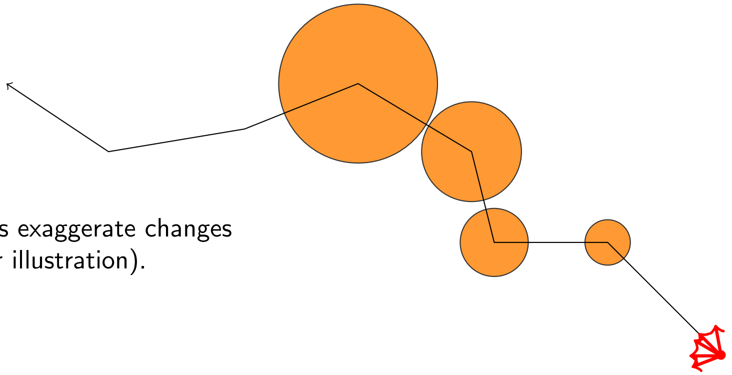
$\mathbf{u}_i(5) \in (1 \pm \epsilon)^5\mathbf{u}_i$. Construct 5 independent sparsifications of $\mathbf{u}$.



Sparsify in Parallel, Use Sequentially

We saw a version of sketch in parallel, use sequentially in connectivity.
Question: Where will we be after 5 steps of MWM?
Recall: If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y}-\mathbf{b}_i)/\mathbf{b}_i\rho}$.

$\mathbf{u}_i(5) \in (1 \pm \epsilon)^5\mathbf{u}_i$. Construct 5 independent sparsifications of $\mathbf{u}$.

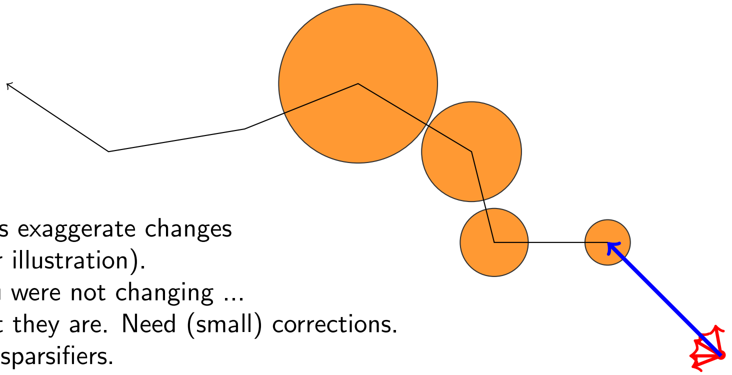


Lets exaggerate changes
(for illustration).

Sparsify in Parallel, Use Sequentially

We saw a version of sketch in parallel, use sequentially in connectivity.
Question: Where will we be after 5 steps of MWM?
Recall: If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y}-\mathbf{b}_i)/\mathbf{b}_i\rho}$.

$\mathbf{u}_i(5) \in (1\pm\epsilon)^5\mathbf{u}_i$. Construct 5 independent sparsifications of $\mathbf{u}$.

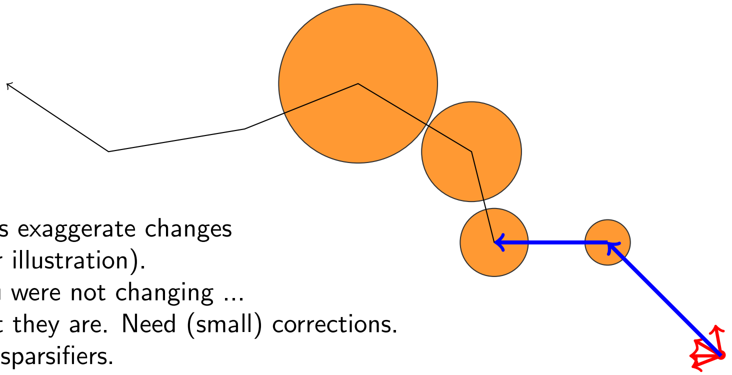


Lets exaggerate changes
(for illustration).
If $\mathbf{u}$ were not changing ...
But they are. Need (small) corrections.
Presparsifiers.

Sparsify in Parallel, Use Sequentially

We saw a version of sketch in parallel, use sequentially in connectivity.
Question: Where will we be after 5 steps of MWM?
Recall: If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y}-\mathbf{b}_i)/\mathbf{b}_i\rho}$.

$\mathbf{u}_i(5) \in (1 \pm \epsilon)^5 \mathbf{u}_i$. Construct 5 independent sparsifications of $\mathbf{u}$.

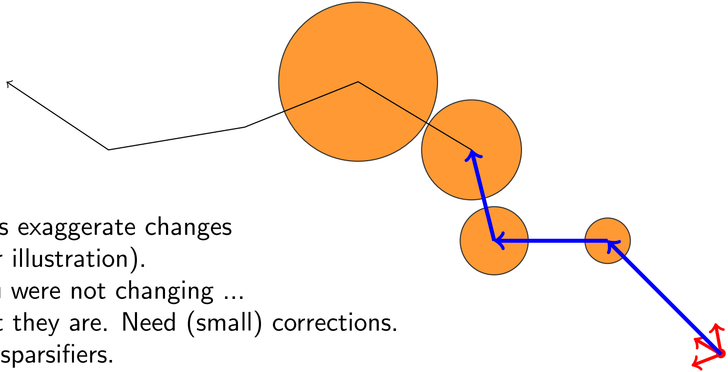


Lets exaggerate changes
(for illustration).
If $\mathbf{u}$ were not changing ...
But they are. Need (small) corrections.
Presparsifiers.

Sparsify in Parallel, Use Sequentially

We saw a version of sketch in parallel, use sequentially in connectivity.
Question: Where will we be after 5 steps of MWM?
Recall: If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y}-\mathbf{b}_i)/\mathbf{b}_i\rho}$.

$\mathbf{u}_i(5) \in (1 \pm \epsilon)^5 \mathbf{u}_i$. Construct 5 independent sparsifications of $\mathbf{u}$.

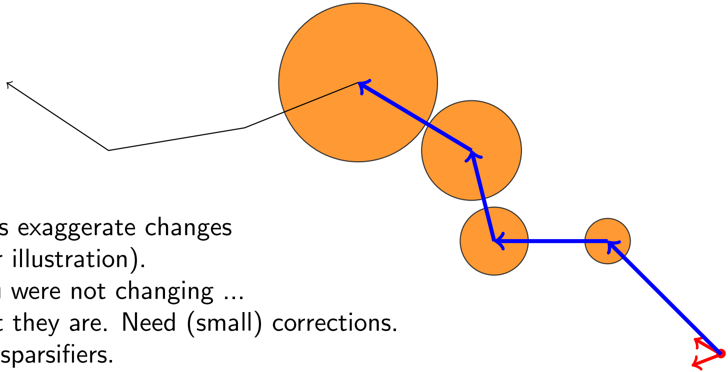


Lets exaggerate changes
(for illustration).
If $\mathbf{u}$ were not changing ...
But they are. Need (small) corrections.
Presparsifiers.

Sparsify in Parallel, Use Sequentially

We saw a version of sketch in parallel, use sequentially in connectivity.
Question: Where will we be after 5 steps of MWM?
Recall: If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y}-\mathbf{b}_i)/\mathbf{b}_i\rho}$.

$\mathbf{u}_i(5) \in (1 \pm \epsilon)^5\mathbf{u}_i$. Construct 5 independent sparsifications of $\mathbf{u}$.

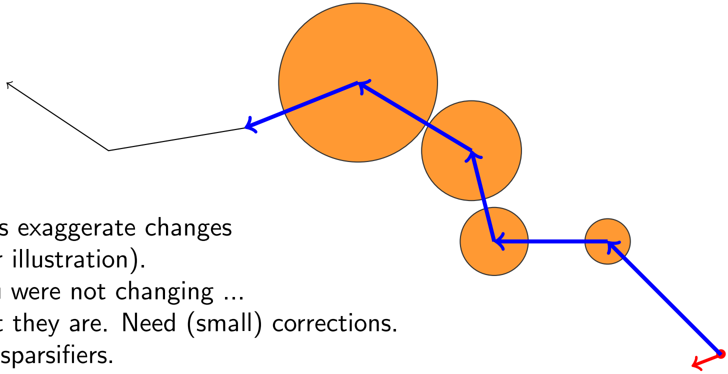


Lets exaggerate changes
(for illustration).
If $\mathbf{u}$ were not changing ...
But they are. Need (small) corrections.
Presparsifiers.

Sparsify in Parallel, Use Sequentially

We saw a version of sketch in parallel, use sequentially in connectivity.
Question: Where will we be after 5 steps of MWM?
Recall: If $\mathbf{A}_i\mathbf{y} > \mathbf{b}_i$: raise $\mathbf{u}_i$, i.e., $\mathbf{u}_i \leftarrow \mathbf{u}_i(1+\epsilon)^{(\mathbf{A}_i\mathbf{y}-\mathbf{b}_i)/\mathbf{b}_i\rho}$.

$\mathbf{u}_i(5) \in (1 \pm \epsilon)^5 \mathbf{u}_i$. Construct 5 independent sparsifications of $\mathbf{u}$.



Lets exaggerate changes
(for illustration).
If $\mathbf{u}$ were not changing ...
But they are. Need (small) corrections.
Presparsifiers.

Non-Bipartite Matching in Small Passes

A **natural** algorithm for non-bipartite matching.

1. Find an initial solution of the dual Problem. (A trend.)
2. Assign $u_{ij} = 1$ for all edges.
3. For $O(10/\epsilon)$ steps:
   3.1 Compute t sparsifiers with $n^{1.1}$ edges using u_{ij}.
   3.2 Find the best weighted matching in the edges in the t sparsifications. (w_{ij} unchanged).
   3.3 Keep the largest weight matching found (say β) so far.
   3.4 Recompute u_{ij}

Recompute:

1. $t = O(\frac{1}{\epsilon} \log n)$
2. Simulate t steps of a primal-dual algorithm trying to prove Feasible Dual $\leq \beta(1 + O(\epsilon))$.
3. Adjust the sparsification in between.

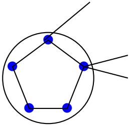
Cuts, Duals and Graph Sparsification

$$\begin{aligned} \beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\ \sum_j y_{ij} \quad & \leq 1 \quad \forall i \\ \sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\ y_{ij} \quad & \geq 0 \end{aligned}$$

$$\sum_{i,j \in U} y_{ij} \leq \lfloor |U|/2 \rfloor \quad \Longleftrightarrow \sum_{i \in U} \left(\sum_j y_{ij} \right) - \left(\sum_{i \in U, j \notin U} y_{ij} \right) \leq 2 \lfloor |U|/2 \rfloor$$

$\sum_{i \in U, j \notin U} y_{ij} = Cut(U, V - U)$.

Find small cuts (with odd vertex sizes).



Cuts, Duals and Graph Sparsification

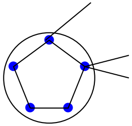
$$\begin{aligned}\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\ \sum_j y_{ij} \quad & \leq 1 \quad \forall i \\ \sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\ y_{ij} \quad & \geq 0\end{aligned}$$

$$\begin{aligned}& \beta^* = \min \sum_i x_i + \sum_U \left\lfloor \frac{|U|}{2} \right\rfloor z_U \\ \mathbf{u}_{ij}: \quad & x_i + x_j + \sum_{i,j \in U} z_U \geq w_{ij} \quad \forall (i,j) \in E \\ & x_i, z_U \geq 0\end{aligned}$$

$$\sum_{i,j \in U} y_{ij} \leq \lfloor |U|/2 \rfloor \quad \iff \sum_{i \in U} \left(\sum_j y_{ij} \right) - \left(\sum_{i \in U, j \notin U} y_{ij} \right) \leq 2 \lfloor |U|/2 \rfloor$$

$\sum_{i \in U, j \notin U} y_{ij} = Cut(U, V - U)$.

Find small cuts (with odd vertex sizes).



Cuts, Duals and Graph Sparsification

$$\begin{aligned} \beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\ \sum_j y_{ij} \quad & \leq 1 \quad \forall i \\ \sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\ y_{ij} \quad & \geq 0 \end{aligned}$$

$$\begin{aligned} & \beta^* = \min \sum_i x_i + \sum_U \left\lfloor \frac{|U|}{2} \right\rfloor z_U \\ \mathbf{u}_{ij}: \quad & x_i + x_j + \sum_{i,j \in U} z_U \geq w_{ij} \quad \forall (i,j) \in E \\ & x_i, z_U \geq 0 \end{aligned}$$

Standard Algorithm: Augment, contract blossoms, ... (many rounds).
Signature: feasible,. . . ,feasible (larger), . . . , feasible, (near) optimal

Cuts, Duals and Graph Sparsification

$$\begin{aligned} \beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\ \sum_j y_{ij} \quad & \leq 1 \quad \forall i \\ \sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\ y_{ij} \quad & \geq 0 \end{aligned}$$

$$\begin{aligned} & \beta^* = \min \sum_i x_i + \sum_U \left\lfloor \frac{|U|}{2} \right\rfloor z_U \\ \mathbf{u}_{ij} : \quad & x_i + x_j + \sum_{i,j \in U} z_U \geq w_{ij} \quad \forall (i,j) \in E \\ & x_i, z_U \geq 0 \end{aligned}$$

Standard Algorithm: Augment, contract blossoms, ... (many rounds).
Signature: feasible,. . . ,feasible (larger), . . . , feasible, (near) optimal

Signature of previous algorithm:
infeasible,. . . ,infeasible (smaller), . . . , feasible, (near) optimal

New algorithm?

Cuts, Duals and Graph Sparsification

$$
\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\
\sum_j y_{ij} \quad & \leq 1 \quad \forall i \\
\sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}
$$

$$
\begin{aligned}
& \beta^* = \min \sum_i x_i + \sum_U \left\lfloor \frac{|U|}{2} \right\rfloor z_U \\
\mathbf{u}_{ij}: \quad & x_i + x_j + \sum_{i,j \in U} z_U \geq w_{ij} \quad \forall (i,j) \in E \\
& x_i, z_U \geq 0
\end{aligned}
$$

Standard Algorithm: Augment, contract blossoms, ... (many rounds).
Signature: feasible,. . . ,feasible (larger), . . . , feasible, (near) optimal

Signature of previous algorithm:
infeasible,. . . ,infeasible (smaller), . . . , feasible, (near) optimal

New algorithm?
infeasible dual, . . . , (estimate of β^* is increasing), . . . , (near) optimal
($O(1/\epsilon)$ rounds, sparsification)

Cuts, Duals and Graph Sparsification

$$\begin{aligned} \beta^* = \max \quad & \sum_{(i,j)} w_{ij} y_{ij} \\ \sum_j y_{ij} \quad & \leq 1 \quad \forall i \\ \sum_{i,j \in U} y_{ij} \quad & \leq \lfloor |U|/2 \rfloor \quad \forall U \\ y_{ij} \quad & \geq 0 \end{aligned}$$

$$\begin{aligned} & \beta^* = \min \sum_i x_i + \sum_U \left\lfloor \frac{|U|}{2} \right\rfloor z_U \\ \mathbf{u}_{ij} : \quad & x_i + x_j + \sum_{i,j \in U} z_U \geq w_{ij} \quad \forall (i,j) \in E \\ & x_i, z_U \geq 0 \end{aligned}$$

Standard Algorithm: Augment, contract blossoms, ... (many rounds).
Signature: feasible,. . . ,feasible (larger), . . . , feasible, (near) optimal

Signature of previous algorithm:
infeasible,. . . ,infeasible (smaller), . . . , feasible, (near) optimal

New algorithm?
infeasible dual, . . . , (estimate of β^* is increasing), . . . , (near) optimal
($O(1/\epsilon)$ rounds, sparsification)
. . . . . . keep best matching seen so far, ... (near) optimal

New Relaxations for Maximum Matching, $\ldots 3, 2, 1$

Lets consider $w_{ij} = 1$.

$$\begin{aligned}
\beta^* = \max \quad & \sum_{(i,j)} y_{ij} - 3\sum_i \mu_i \\
\sum_j y_{ij} - 2\mu_i \quad & \leq 1 \quad \forall i \\
\sum_{i,j \in U} y_{ij} - \sum_{i \in U} \mu_i \quad & \leq \lfloor |U|/2 \rfloor + \quad \forall U \\
y_{ij} \quad & \geq 0
\end{aligned}$$

New Relaxations for Maximum Matching, $\ldots 3, 2, 1$

Lets consider $w_{ij} = 1$.

$$
\begin{array}{lll}
\beta^* = \max & \sum_{(i,j)} y_{ij} - 3 \sum_i \mu_i & \\
\sum_j y_{ij} - 2\mu_i & \leq 1 & \forall i \\
\sum_{i,j \in U} y_{ij} - \sum_{i \in U} \mu_i & \leq \lfloor |U|/2 \rfloor + & \forall U \\
y_{ij} & \geq 0 &
\end{array}
$$

$$
\begin{array}{ll}
& \beta^* = \min \sum_i x_i + \sum_U \left\lfloor \frac{|U|}{2} \right\rfloor z_U \\
\mathbf{u}_{ij} : & x_i + x_j + \sum_{i,j \in U} z_U \geq w_{ij} \quad \forall (i,j) \in E \\
& 2x_i + \sum_{i \in U} z_U \leq 3 \quad \forall i \in V \\
& x_i, z_U \geq 0
\end{array}
$$

Wrap up

(1) Primitives: Sampling, Sketching and Sparsification.

Wrap up

(1) Primitives: Sampling, Sketching and Sparsification.

(2) LPs/SDPs (MWM) on Streams.

Wrap up

(1) Primitives: Sampling, Sketching and Sparsification.
(2) LPs/SDPs (MWM) on Streams.
(3) Remember a small number of weight values.

Wrap up

(1) Primitives: Sampling, Sketching and Sparsification.
(2) LPs/SDPs (MWM) on Streams.
(3) Remember a small number of weight values.
(4) Compute in sketch (sparsified) space entirely. Correlation clustering.

Wrap up

(1) Primitives: Sampling, Sketching and Sparsification.
(2) LPs/SDPs (MWM) on Streams.
(3) Remember a small number of weight values.
(4) Compute in sketch (sparsified) space entirely. Correlation clustering.
(5) May need to change the natural relaxations (convergence speed).

Wrap up

(1) Primitives: Sampling, Sketching and Sparsification.
(2) LPs/SDPs (MWM) on Streams.
(3) Remember a small number of weight values.
(4) Compute in sketch (sparsified) space entirely. Correlation clustering.
(5) May need to change the natural relaxations (convergence speed).
(6) May need new relaxations for rounding.

Wrap up

(1) Primitives: Sampling, Sketching and Sparsification.
(2) LPs/SDPs (MWM) on Streams.
(3) Remember a small number of weight values.
(4) Compute in sketch (sparsified) space entirely. Correlation clustering.
(5) May need to change the natural relaxations (convergence speed).
(6) May need new relaxations for rounding.
(7) Think differently. The real voyage of discovery ...

Thank You