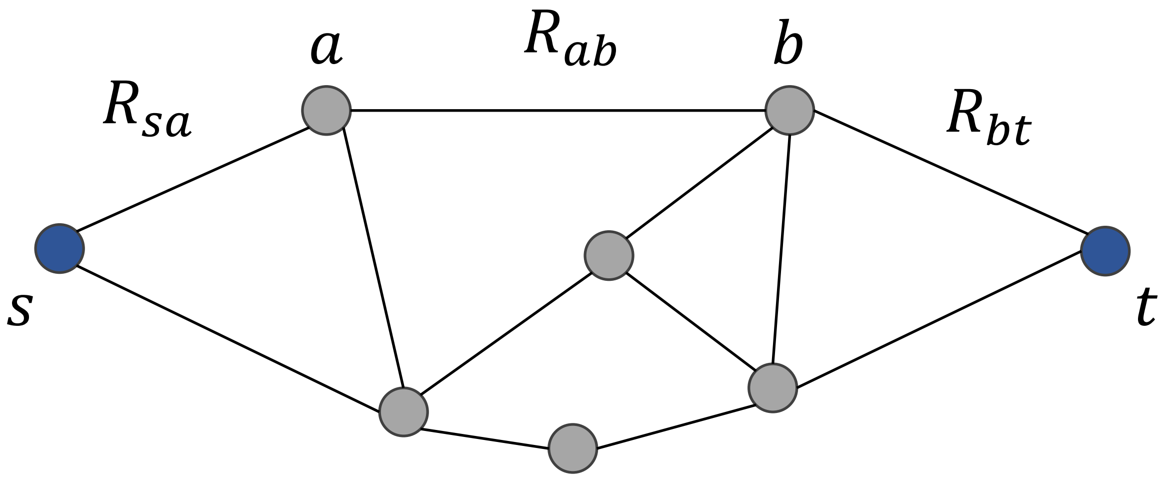
Applications of MWU, Maximum Flow and Vertex Cover

Yury Makarychev
TTIC and the University of Chicago

Electric Networks

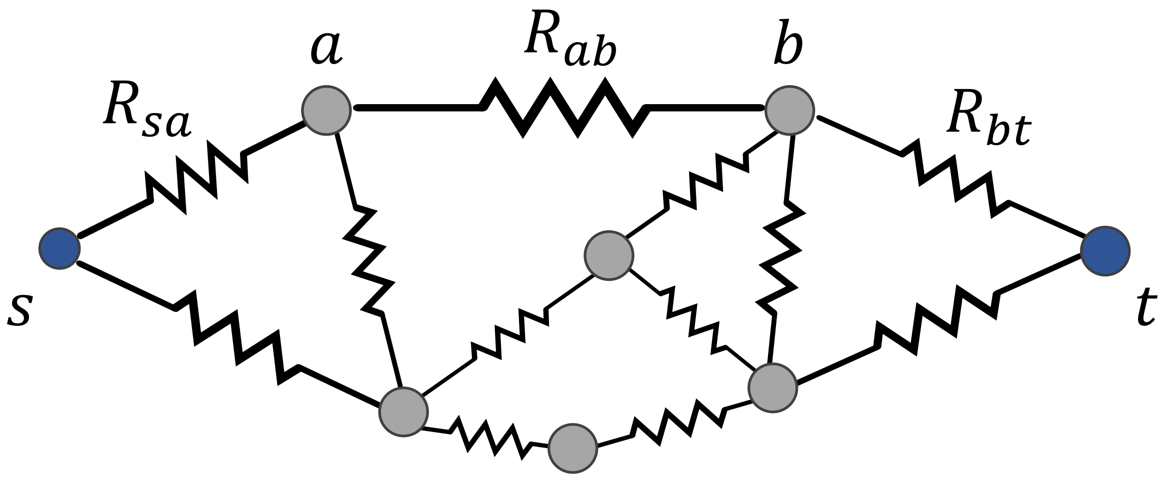
Consider a connected undirected graph $G = (V, E)$ with source s and sink t, in which every edge e is assigned a resistance $R_e > 0$.



Electric Networks

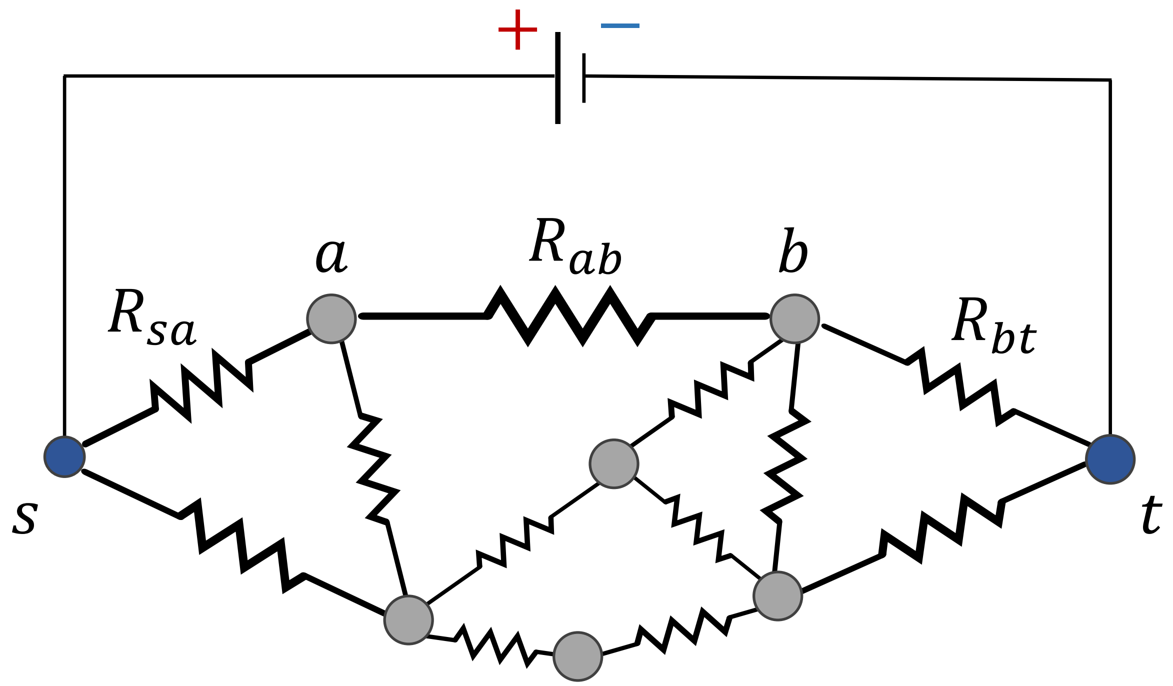
Consider a connected undirected graph $G = (V, E)$ with source s and sink t, in which every edge e is assigned a resistance $R_e > 0$.

Graph G represents an electric network/circuit.



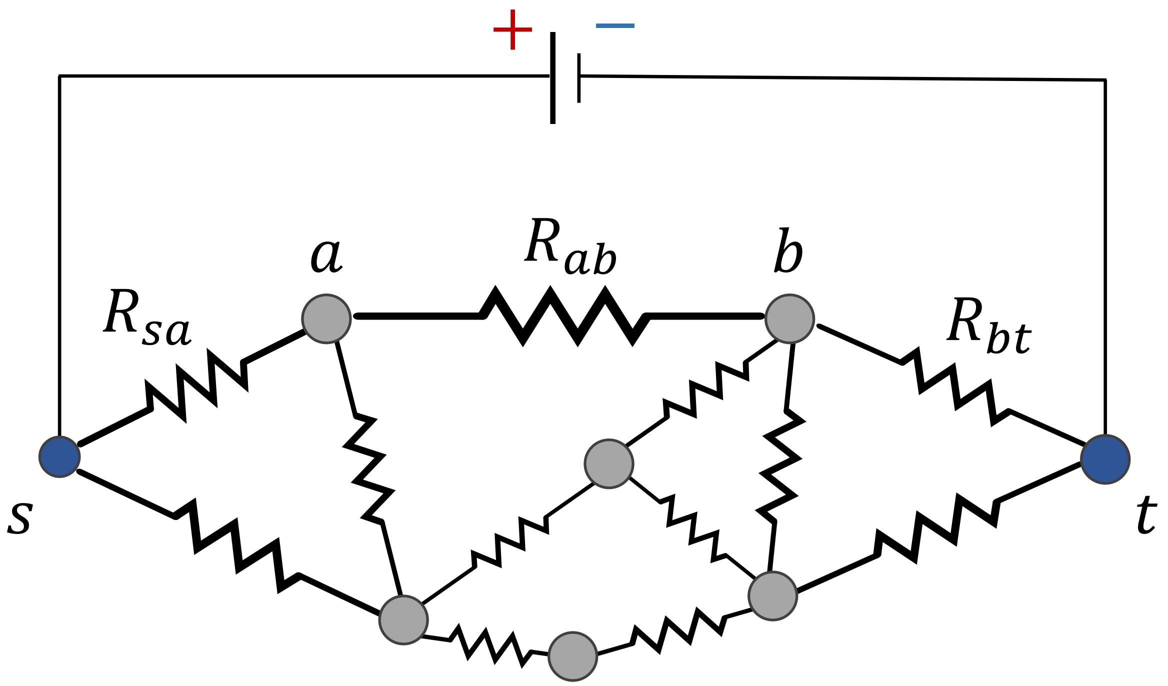
Electric Networks

Let's apply voltage to s and t. What will happen?

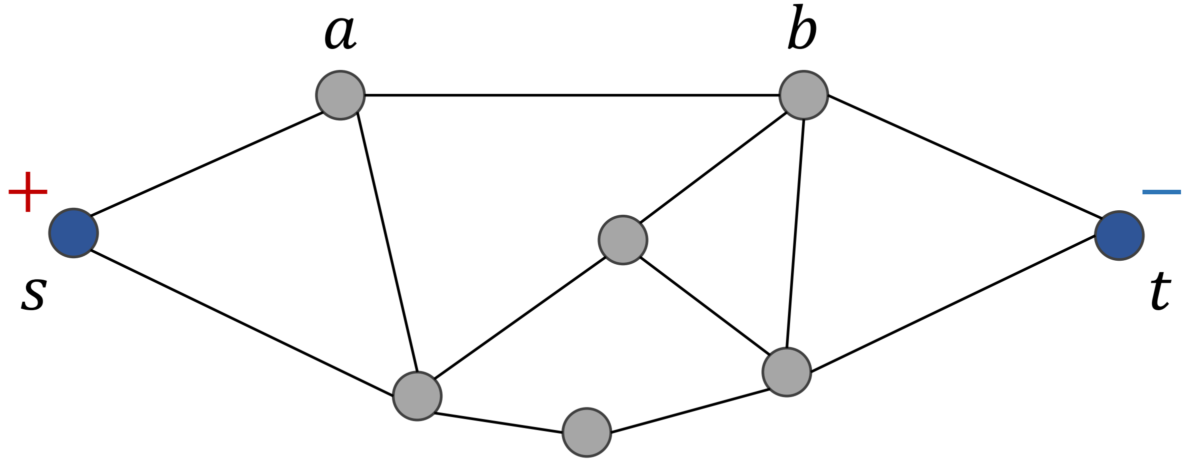


Electric Networks

Let's apply voltage to s and t. Electric current will flow from s to t.



Electricity Flows From s to t



Let I_{ab} be the current through resistor (a, b).

$I_{ab} > 0$ if the electric flow is from a to b; $I_{ab} = -I_{ba} < 0$ if the flow is from b to a.

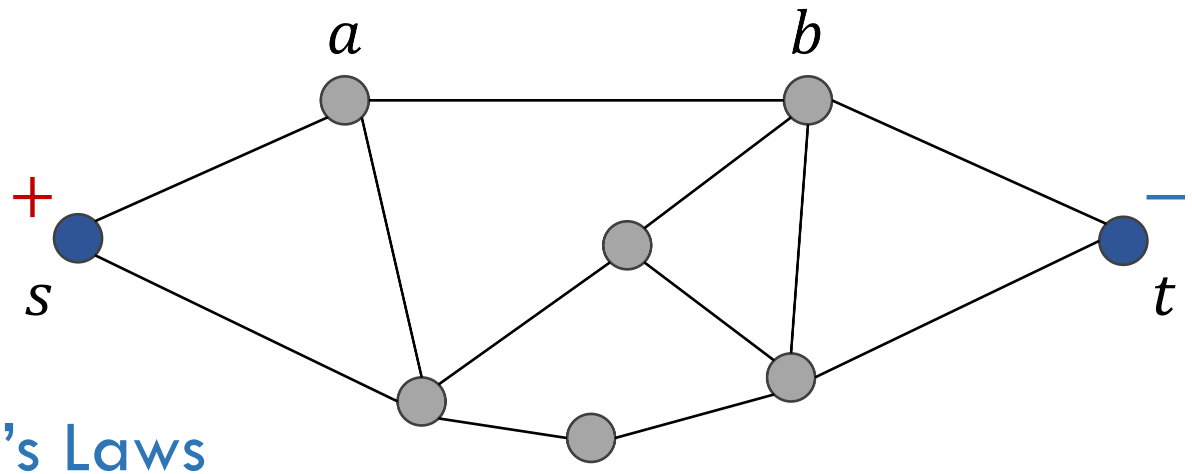
Let v_a be the voltage at vertex/node a of G.

Kirchhoff's Laws

- $I_{ab} = \frac{v_a - v_b}{R_{ab}}$
- $\sum_{b \in N(a)} I_{ab} = 0$ for every $a \neq \{s, t\}$ (electric flow conservation constraints)

We get a system of linear equations with variables $\{I_{ab}\}$ and $\{v_a\}$

Electricity Flows From s to t



Kirchhoff's Laws

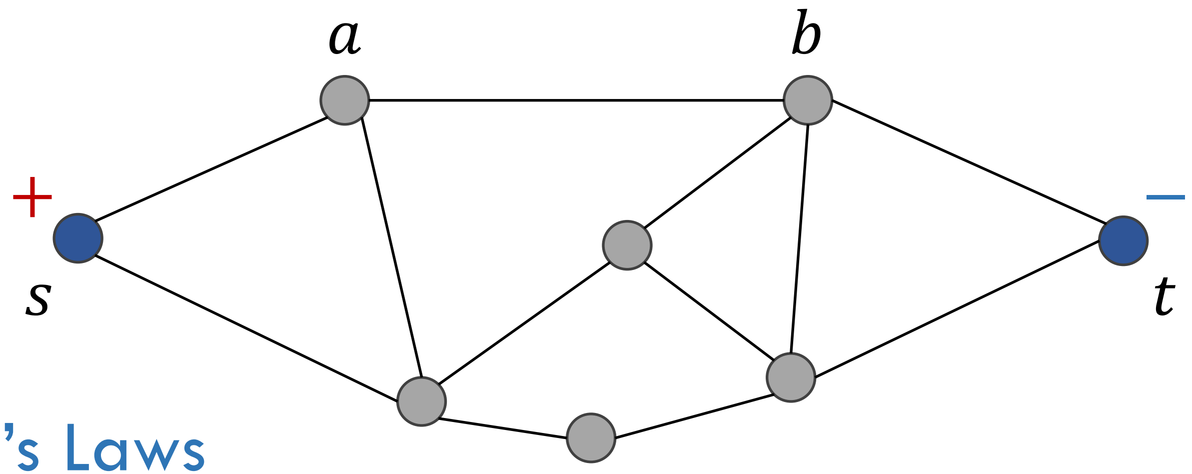
- $I_{ab} = \frac{v_a - v_b}{R_{ab}}$
- $\sum_{b \in N(a)} I_{ab} = 0$ for every $a \neq \{s,t\}$ (electric flow conservation constraints)

We get a system of linear equations with variables $\{I_{ab}\}$ and $\{v_a\}$

Consider an electric flow. Increase the current on every edge α times (α is the same number for all edges). We get a valid electric flow. Other than that, the electric flow is uniquely defined.

$\Rightarrow$ For every F, there is a unique flow with the total current F from s to t.

Electricity Flows From s to t



Kirchhoff's Laws

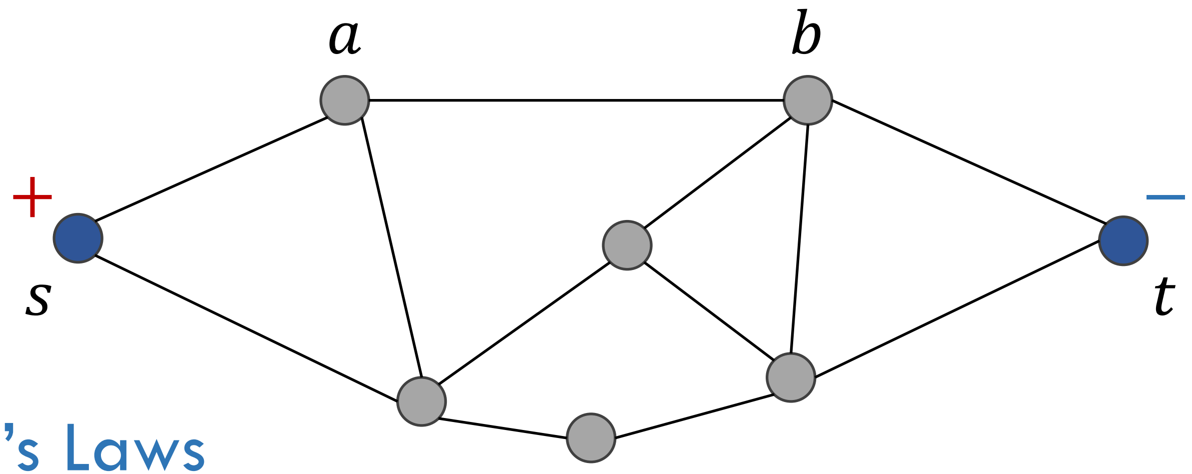
- $I_{ab} = \frac{v_a - v_b}{R_{ab}}$
- $\sum_{b \in N(a)} I_{ab} = 0$ for every $a \neq \{s, t\}$ (electric flow conservation constraints)

We get a system of linear equations with variables $\{I_{ab}\}$ and $\{v_a\}$

For every F, there is a unique flow with the total current F from s to t.

This flow can be found in time $\tilde{O}(m)$ using linear algebra.

Power Dissipation



Kirchhoff's Laws

- $I_{ab} = \frac{v_a - v_b}{R_{ab}}$
- $\sum_{b \in N(a)} I_{ab} = 0$ for every $a \neq \{s, t\}$ (electric flow conservation constraints)

We get a system of linear equations with variables $\{I_{ab}\}$ and $\{v_a\}$

Resistor (a, b) heats up when electricity flows through it. The power dissipation equals

$$P_{ab} = R_{ab}\, I_{ab}^2 = \frac{\big(v(a) - v(b)\big)^2}{R_{ab}} = \big(v(a) - v(b)\big) \cdot I_{ab}$$

The principle of least action

The total power dissipation of the entire network is

$$P = \sum_{(a,b)\in E} P_{ab} = \sum_{(a,b)\in E} R_{ab} I_{ab}^2$$

The principle of least action: Fix F. Consider all possible s-t flows of value F that satisfy the flow conservation constraints. Among them, the electric flow is the one that minimizes the total power dissipation.

min $\sum_{(a,b)\in E} R_{ab} I_{ab}^2$

s.t. I satisfies flow conservation constraints ($\sum_{b\in N(a)} I_{ab} = 0$)

I has value F ($\sum_{b\in N(s)} I_{sb} = F$)

Electric Flow vs Graph Flow

	Electric Flow	Graph Flow
Flow conservation constraint	x	x
Edge capacity constraints	-	x
Minimizes power dissipation	x	-

There is a highly efficient algorithm for finding electric flows. Can we use it to find a graph s-t flow?

Solving Max Flow: Edge-Glow Game

Consider an instance of Max Flow in an undirected graph $G = (V, E)$ with unit capacities $c_{uv} = 1$.

Assume we want to find a feasible flow of value F. Consider the following game.

Player A (edge player):

- pure strategy: A chooses an edge $e \in E$
- mixed strategy: A chooses a distribution of edge α_e

Player B (flow player):

- B chooses a flow f of value F that satisfies flow conservation constrains, but not necessarily capacity constraints.

Payoff: $|f(e)|$

Solving Max Flow: Edge-Flow Game

Player A (edge player):

- pure strategy: A chooses an edge $e \in E$
- mixed strategy: A chooses a distribution of edge α_e

Player B (flow player):

- B chooses a flow f of value F that satisfies flow conservation constrains, but not necessarily capacity constraints.

Payoff: $|f(e)|$

Claim: there is a feasible s-t flow of value F, if and only if the value of the game is

$$val \leq 1$$

Solving Max Flow: Edge-Flow Game

Payoff: $|f(e)|$

Claim: there is a feasible s-t flow of value F, if and only if the value of the game is

$$val \leq 1$$

Assume that there is a feasible flow f of value F. Then player B chooses f.

No matter what edge e player A chooses:

$$payoff = |f(e)| \leq 1$$

That is, B may guarantee that the payoff is at most 1.

Solving Max Flow: Edge-Flow Game

Payoff: $|f(e)|$

Claim: there is a feasible s-t flow of value F, if and only if the value of the game is

$$val \leq 1$$

Assume that there is *no* feasible flow f of value F. Let f be the strategy of player B.

By our assumption, f is not a feasible solution $\Rightarrow |f(e)| > 1$ for some edge e.

Player A chooses this edge e:

$$payoff = |f(e)| > 1$$

That is, A may guarantee that the payoff is strictly greater than 1.

Solving Max Flow: Edge-Flow Game

Let F^* be the value of the maximum flow.

We assume that we are given $F \leq F^*$ and asked to find a flow of value at least $\frac{F}{1+\varepsilon}$.

Outline of our algorithm:

- Since $F \leq F^*$, the value of the *edge-flow* game is at most 1.
- Using MWU, find a strategy f for B of value at most $1 + \varepsilon$.
  - The value of f is F
  - f satisfies flow conservation constraints
  - $|f(e)| \leq 1 + \varepsilon$ for every edge e. Why?

Q: Is f a feasible flow? If not, can we “fix” it?

Q: Why can we assume that we know F?

Oracle

We want to use Multiplicate Weight Update (MWU) method to find f.

To this end, we need to implement an "oracle" that given probabilities/weights α_e, finds a response for B:

- $\sum_e \alpha_e \, |f(e)| \leq 1$
- flow f satisfies flow conservation constraints
- f has value F

min $\sum_{(a,b)\in E} R_{ab} I_{ab}^2$

s.t. I satisfies flow conservation constraints

I has value F

Oracle

- $\sum_e \alpha_e |f(e)| \leq 1$
- flow f satisfies flow conservation constraints
- f has value F

The oracle defines an electric network on G with appropriately chosen resistances R_e and then finds an electric flow f of value F in this network.

Q: $R_e = ?$ Suggestions?

min $\sum_{(a,b)\in E} R_{ab} I_{ab}^2$

s.t. I satisfies flow conservation constraints

I has value F

Oracle

- $\sum_e \alpha_e |f(e)| \leq 1$
- flow f satisfies flow conservation constraints
- f has value F

The oracle defines an electric network on G with appropriately chosen resistances R_e and then finds an electric flow f of value F in this network.

A1: Let $R_e = \alpha_e$.

min $\sum_{(a,b)\in E} R_{ab} I_{ab}^2$

s.t. I satisfies flow conservation constraints

I has value F

Oracle

- $\sum_e \alpha_e |f(e)| \leq 1$
- flow f satisfies flow conservation constraints
- f has value F

The oracle defines an electric network on G with appropriately chosen resistances R_e and then finds an electric flow f of value F in this network.

A2: Let $R_e = \alpha_e + \varepsilon/m$.

min $\sum_{(a,b)\in E} R_{ab} I_{ab}^2$

s.t. I satisfies flow conservation constraints

I has value F

Oracle

- $\sum_e \alpha_e |f(e)| \le 1$
- flow f satisfies flow conservation constraints
- f has value F

The oracle defines an electric network on G with appropriately chosen resistances R_e and then finds an electric flow f of value F in this network.

A2: Let $R_e = \alpha_e + \varepsilon/m$.

Cauchy-Schwarz: $\sum_i a_i b_i \le \left(\sum_i a_i^2 \cdot \sum_i b_i^2\right)^{1/2}$

$$R_e = \alpha_e + \varepsilon/m$$

Oracle

Let $\hat{f}$ be a feasible graph graph flow of value F. Its power dissipation is

$$\hat{P} = \sum_e R_e \hat{f}(e)^2 \leq \sum_e R_e = \sum_e \alpha_e + \sum_e \varepsilon/m = 1 + \varepsilon$$

Now, for electric flow f, we have:

$$\sum_e R_e |f(e)| = \sum_e \left(\sqrt{R_e}\,|f(e)|\right)\sqrt{R_e} \leq \left(\underbrace{\sum_e \left(\sqrt{R_e}\,|f(e)|\right)^2}_{= P^*} \underbrace{\sum_e R_e}_{= 1+\varepsilon}\right)^{\frac{1}{2}}$$

$$R_e = \alpha_e + \varepsilon/m$$

Oracle

$$\hat{P} = \sum_e R_e \hat{f}(e)^2 \leq \sum_e R_e = \sum_e \alpha_e + \sum_e {}^{\varepsilon}/_{m} = 1 + \varepsilon$$

Now:

$$\sum_e R_e |f(e)| \leq \sqrt{1+\varepsilon}\,\sqrt{P^*} \leq \sqrt{1+\varepsilon}\sqrt{\hat{P}} \leq (1+\varepsilon)^{3/2} < 1 + O(\varepsilon)$$

The value of response f for B is at most $1 + O(\varepsilon)$.

$$R_e = \alpha_e + \varepsilon/m$$

Running Time: Bounding the Width

$$\hat{P} = \sum_e R_e \hat{f}(e)^2 \leq \sum_e R_e = \sum_e \alpha_e + \sum_e {}^{\varepsilon}/_{m} = 1 + \varepsilon$$

The oracle width is $\rho = \max_e |f(e)|$.

$$1 + \varepsilon \geq \hat{P} \geq \sum_e R_e f(e)^2 \geq \sum_e \frac{\varepsilon}{m} f(e)^2 \geq \frac{\varepsilon}{m} \cdot \max_e f(e)^2 = \varepsilon\rho^2/m$$

Thus, $\rho \leq O\left(\sqrt{m/\varepsilon}\right)$.

$$R_e = \alpha_e + \varepsilon/m$$

Oracle

Running time: $\tilde{O}(T(m+n))$ where $T = O\left(\frac{\rho \log n}{\varepsilon^2}\right) = O\left(\frac{m^{1/2} \log n}{\varepsilon^{5/2}}\right)$.

$$\tilde{O}(m^{3/2}/\varepsilon^{5/2})$$

Using similar ideas, we can get running time $\tilde{O}_\varepsilon(mn^{1/3})$.

Approximately Solving Vertex Cover LP

Solving LPs using MWU: Vertex Cover

We can use MWU to solve LPs. Consider a specific example.

variables: x_u

min $\sum_{u \in \mathrm{V}} x_u$

$x_u + x_v \geq 1$ for every $(u, v) \in E$

$x_u \geq 0$ for every $u \in V$

Assume that we know that we are given $k \geq LP$ and want to find a solution of cost at most $(1 + \varepsilon)k$.

Vertex Cover

$\sum_{u \in \mathrm{V}} x_u = k$

$x_u + x_v \geq 1$ for every $(u, v) \in E$

$x_u \geq 0$ for every $u \in V$

Consider the following game:

Player A (edge player)

Pure strategy: an edge e

Mixed strategy: a distribution of edges α_e

Player B (solution player)

an assignment x_u s.t. (i) $x_u \in [0,1]$ for all u and (ii) $\sum_u x_u = k$

Payoff: $\sum_e \alpha_e (1 - x_u - x_v)$

$\sum_{u \in V} x_u = k$
$x_u + x_v \geq 1$
$x_u \geq 0$

Vertex Cover

A: a distribution of edges α_e

B: an assignment x_u s.t. (i) $x_u \in [0,1]$ for all u and (ii) $\sum_u x_u = k$

Payoff: $f(\alpha, x) = \sum_{(u,v)} \alpha_{(u,v)}(1 - x_u - x_v)$

Claim: x is a feasible LP solution if and only if x is a strategy for B with guaranteed payoff $f(e, x) \leq 0$.

Proof: if x is a feasible solution then x is a feasible strategy and for every (u, v)

$$f\big((u,v), x\big) = 1 - x_u - x_v \leq 0$$

If x is a strategy as in the statement, then for every (u, v)

$$1 - x_u - x_v = f\big((u,v), x\big) \leq 0$$

$\sum_{u \in V} x_u = k$
$x_u + x_v \geq 1$
$x_u \geq 0$

Vertex Cover

A: a distribution of edges α_e

B: an assignment x_u s.t. (i) $x_u \in [0,1]$ for all u and (ii) $\sum_u x_u = k$

Payoff: $f(\alpha, x) = \sum_{(u,v)} \alpha_{(u,v)}(1 - x_u - x_v)$

Algorithm outline:

- Find k using binary search
- Find a nearly-optimal strategy x for play B.
- We have: $1 - x_u - x_v \leq \varepsilon$ for every edge (u, v).
- That is, $x_u + x_v \geq 1 - \varepsilon$
- Rescale: $x_u = \frac{1}{1-\varepsilon} x_u$. The new value is $\frac{k}{1-\varepsilon}$.

Oracle

A: a distribution of edges α_e

B: an assignment x_u s.t. (i) $x_u \in [0,1]$ for all u and (ii) $\sum_u x_u = k$

Payoff: $f(\alpha, x) = \sum_{(u,v)} \alpha_{(u,v)}(1 - x_u - x_v)$

Oracle: given α, find the best response x for this α.

$$f(\alpha, x) = \sum_{(u,v)} \alpha_{(u,v)}(1 - x_u - x_v) = A - \sum_u c_u x_u$$

where $A = \sum \alpha_{(u,v)}$ and $c_u = \left(\sum_{v \in N(u)} \alpha_{(u,v)}\right)$.

Q: What is the best response x?

Oracle

Oracle: given α, find the best response x for this α.

$$f(\alpha, x) = \sum_{(u,v)} \alpha_{(u,v)}(1 - x_u - x_v) = A - \sum_u c_u x_u$$

Q: What is the best response x (the one that minimizes f)?

Oracle

Oracle: given α, find the best response x for this α.

$$f(\alpha, x) = \sum_{(u,v)} \alpha_{(u,v)}(1 - x_u - x_v) = A - \sum_u c_u x_u$$

Q: What is the best response x (the one that minimizes f)?

A: Find k largest coefficients $c_{u_1}, \ldots, c_{u_k}$ and let

$$x_{u_1} = \cdots = x_{u_k} = 1 \text{ and all other } x_u = 0$$

Time for computing x: $O(m)$ time to compute c_u, $O(n \log n)$ time to sort all c_u and choose k largest

$$O(m + n \log n) \to O(m + n) = O(m) \quad \text{(assuming } m \geq n\text{)}$$

Oracle Width and Running Time

The oracle width is $\rho = 1$:

$$f(e, x) = 1 - x_u - x_v \in [-1,1]$$

The running time is

$$O\big(T(\underbrace{m}_{\text{oracle}} + \underbrace{m}_{\text{computing } f(e,x) \text{ for all } e})\big) = O(Tm)$$

$$T = O\left(\frac{\log n}{\varepsilon^2}\right)$$

Running time: $O\left(\frac{m \log n}{\varepsilon^2}\right)$