Lecture 24

Plan:

1) Ellipsoid for LP
2) If time, examples.

Ellipsoid for LP

- Even for feasibility of $P = \{x : Ax \leq b\}$, are issues!
- Finding starting ellipse, E_0

- Bounding volume of P.

can be handled in general, but

- To avoid numerical details, study important special case: (important for combo. opt.)

$$\boxed{\text{Assume } P = \operatorname{conv}(X) \text{ for } X \subseteq \{0,1\}^n, \ \& \ \dim P = n.}$$

e.g.

$$P = \operatorname{conv}\{1_M : M \text{ matching in } G\} \subseteq \mathbb{R}^E$$

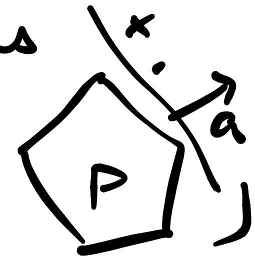
- can handle $\dim P < n$ by eliminating variables if $\operatorname{aff}(P)$ known, tricky if not!

- Given $c \in \mathbb{R}^n$, want to compute

  $$OPT = \max\{c^T x : x \in P\}$$

  in <u>polynomial time</u> given separation oracle for P.

(SEP ORACLE tells us $x \in P$ or gives separating hyperplane $a^T x \le b$)



- What's polynomial time here?

- <u>Input-size</u>:
  - ▷ Assume $c \in \cancel{\mathbb{Q}^n}$ $\mathbb{Z}^n$

    $\mathbb{Q}^n$ b/c must store on machine

    $\mathbb{Z}^n$ by clearing denoms.

▷ assume each entry satisfies
$|c_i| \le M \in \mathbb{N}$;
need $\lceil \log_{10} M \rceil$ digits to
write each c_i

(i.e. $\lceil \log_2 M \rceil$ bits)

$\Rightarrow$ total input size $\le n \lceil \log_2 M \rceil$.

- Thus we'll take polynomial time to mean $\mathrm{poly}(n, \ln M)$ #steps / calls to SEP oracle.

e.g. $10n^3(\log M)^2$ ✓

NOT $2^n \log M$ ✗

NOT nM^2 ✗

Implementing the binary Search

- How long do we need to do binary search?
  - ▷ Know $OPT \in \mathbb{Z}$
  - ▷ Know $|OPT| \leq nM$

$c \in \mathbb{R}^n$

$P = \mathrm{conv}(X)$

$|c_i| \leq M$

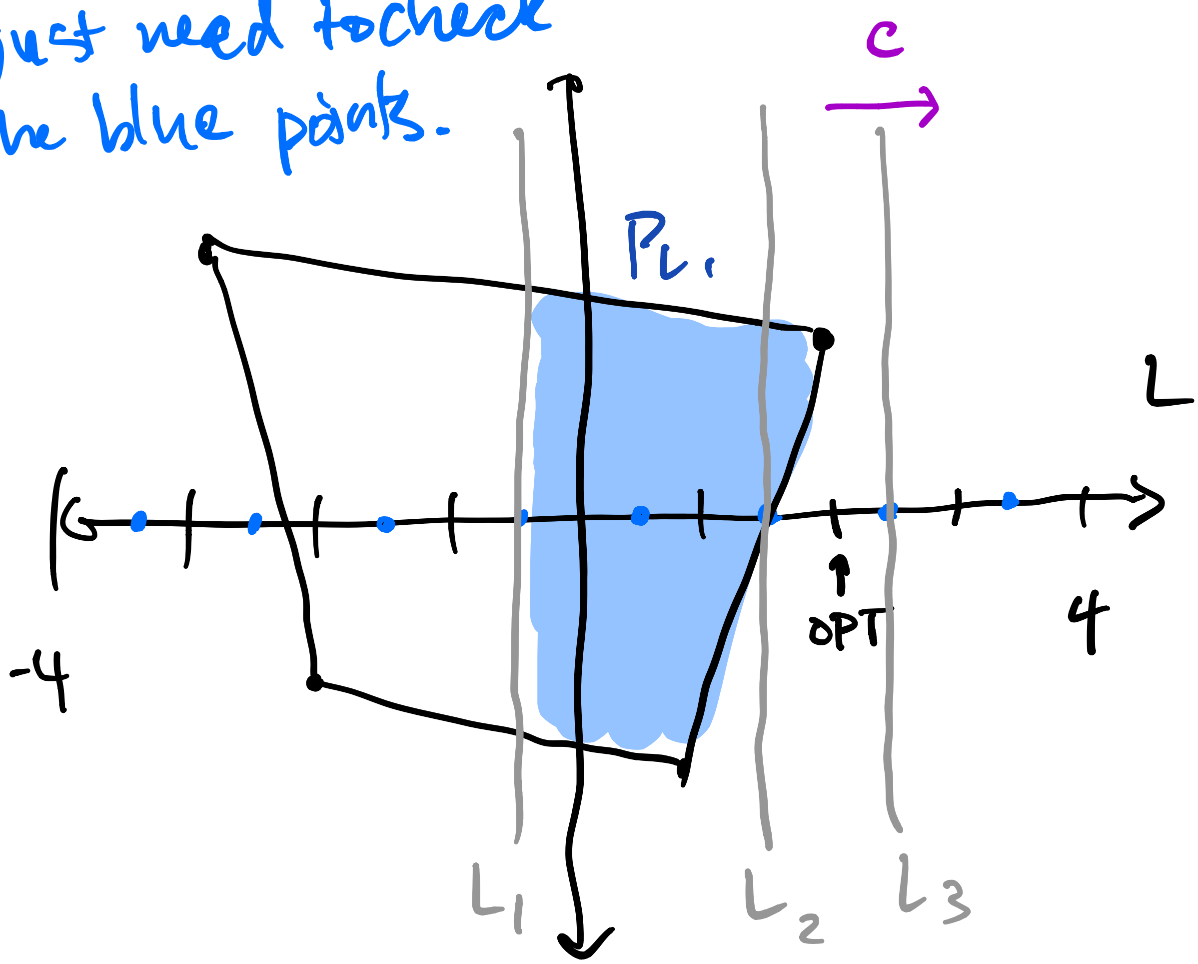
$X \subseteq \{0,1\}^n$

⇒ Thus can *exactly* solve for OPT using binary search; need only check $L = k + \frac{1}{2}$, $k \in \mathbb{Z}$, $|L| < M_n$

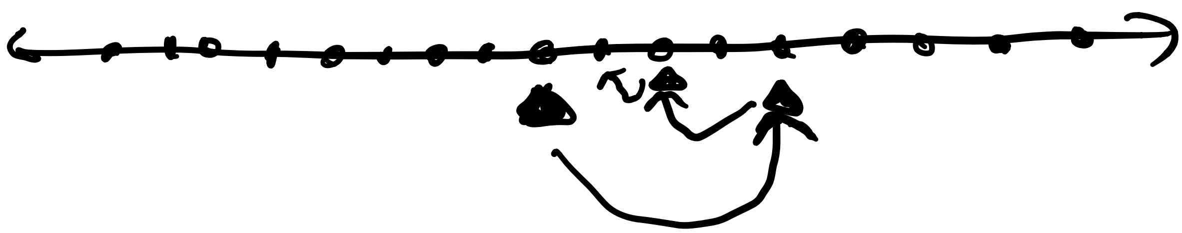
each time we check "is $OPT \geq L$"

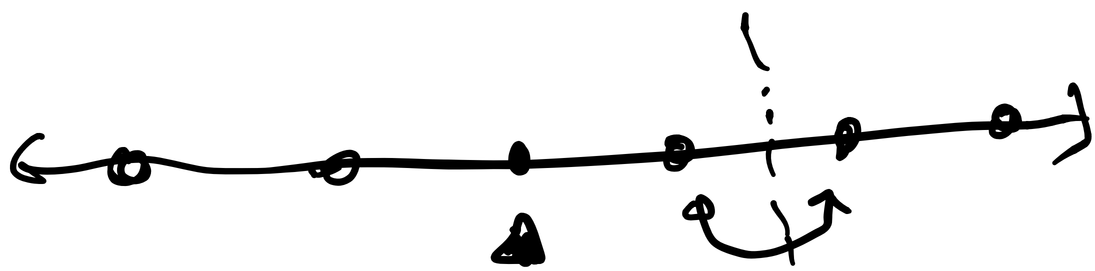
e.g. Suppose $M_n = 4$, $OPT = 2$

just need to check the blue points.



Know $L_2 \leq OPT \leq L_3$, $OPT \in \mathbb{Z}$ ⇒ $OPT = 2$.





this or
this

if only query
things of the form
$K \cdot 2^{n-j}$
in step

- How many steps?

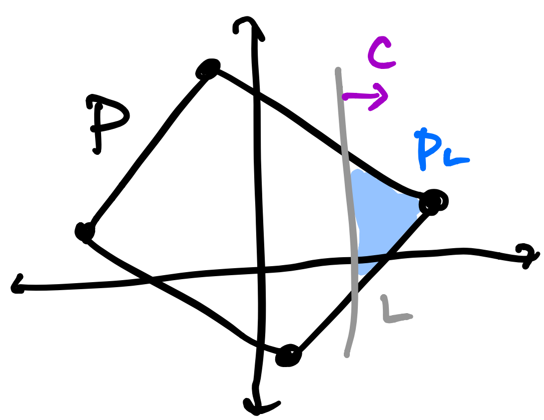
$$\leq \log_2 |2Mn|$$

(# points to check halved at each step..)

- At each step, need to test if $L < OPT$, i.e. if

$$P_L := \{x \in P : c^T x \geq L\}$$

is nonempty.



- For this, use ellipsoid.

Runtime of ellipsoid calls

- **Recall:** to test feasbility with ellipsoid, must know
  - starting ellipsoid $E_0 \supseteq P_L$
  - volume lower bound $\text{vol}\, P_L \geq \delta$ for all $P_L \neq \emptyset$.
- **To test:**
  - run ellipsoid for
    $$2(n+1) \ln \frac{\text{vol}\, E_0}{\delta}$$
    steps. (or until find $x \in P_L$).

▷ if haven't found $x \in P_L$, output that $P_L = \emptyset$.

- Thus we just need lower bound δ on $\mathrm{vol}\, P_L$ & upper bound on $\mathrm{vol}\, E_0$

$\log\left(\frac{1}{\delta}\right), \log \mathrm{vol}\, E_0 \le \mathrm{poly}(n, \log M)$.

Bounding starting ellipsoid

- Simple: $P \subseteq [0,1]^n$

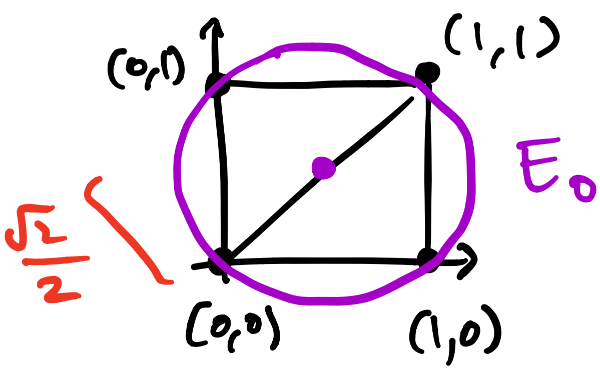
$\Rightarrow P_L \subseteq [0,1]^n$

$\Rightarrow$ any $E_0 \supseteq [0,1]^n$ is ok.

- Can use E_0 = ball centered at $(\frac{1}{2}, \dots, \frac{1}{2})$ radius $\frac{1}{2}\sqrt{n}$.

(∂E_0 goes through all points of $\frac{1}{2}\sqrt{n}$.)

e.g. for $n=2$,



- $\text{Vol } E_0 = \left(\frac{\sqrt{n}}{2}\right)^n \cdot \text{vol (unit ball)}$

because unit ball $\subseteq [-1,1]^n$

$$\leq \left(\frac{\sqrt{n}}{2}\right)^n 2^n = n^{\frac{n}{2}}.$$

$$\Rightarrow \ln \text{vol} E_0 \leq \frac{n}{2} \ln n.$$

Bounding Vol P_L

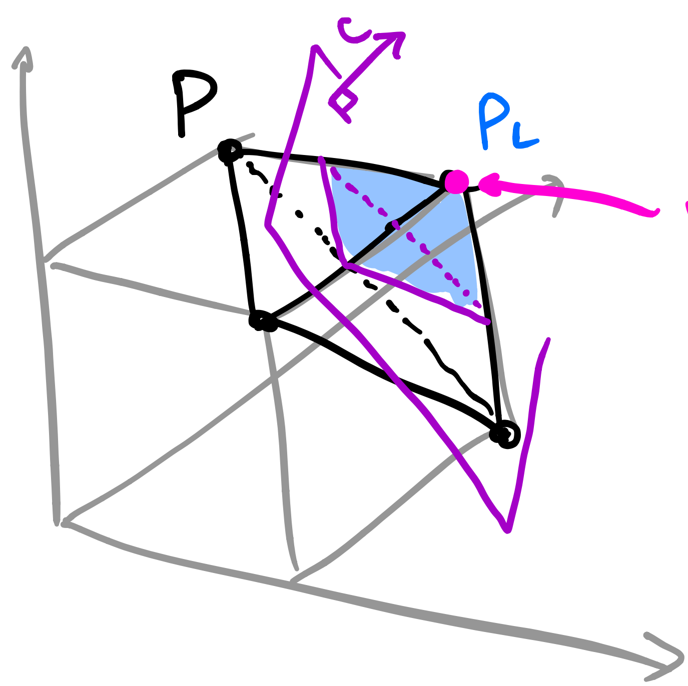
- Need to show
  $P_L \neq \emptyset \Rightarrow \text{Vol } P_L \geq \delta$

  where $\log\left(\frac{1}{\delta}\right) = \text{poly}(n, M)$.

- Since $P_L \neq \emptyset$, contains some optimal vertex $v_0 \in \{0,1\}^n$ of P

  $(c^T v_0 = OPT)$.

e.g. $n = 3$

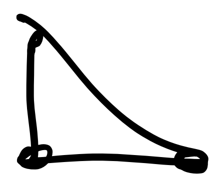


- One way (there are many): find a *simplex* in "corner" of P_L.

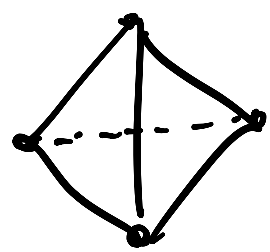
- *Simplex* in $\mathbb{R}^n$ is convex hull of $n+1$ affinely independent points.

e.g.

triangle in $\mathbb{R}^2$



tetrahedron in $\mathbb{R}^3$.

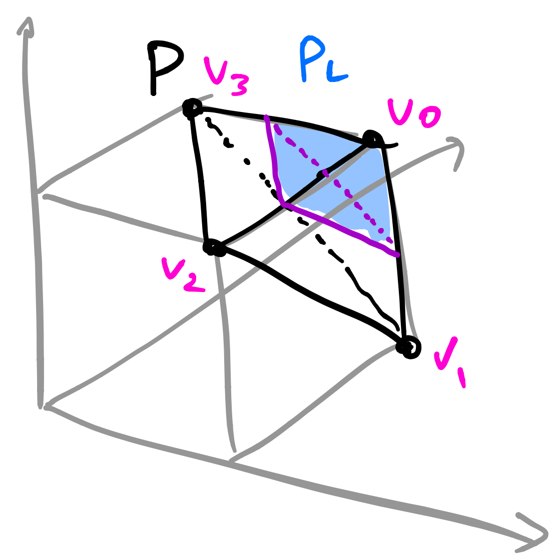


easy to compute volumes of simplices.

- We've assumed P full-dimensional

$\Rightarrow \exists\, v_1, \ldots, v_n \in \{0,1\}^n$ vertices of P s.t. $\operatorname{conv}\{v_0, \ldots, v_n\}$ is full-dimensional. simplex.

e.g. $n = 3$

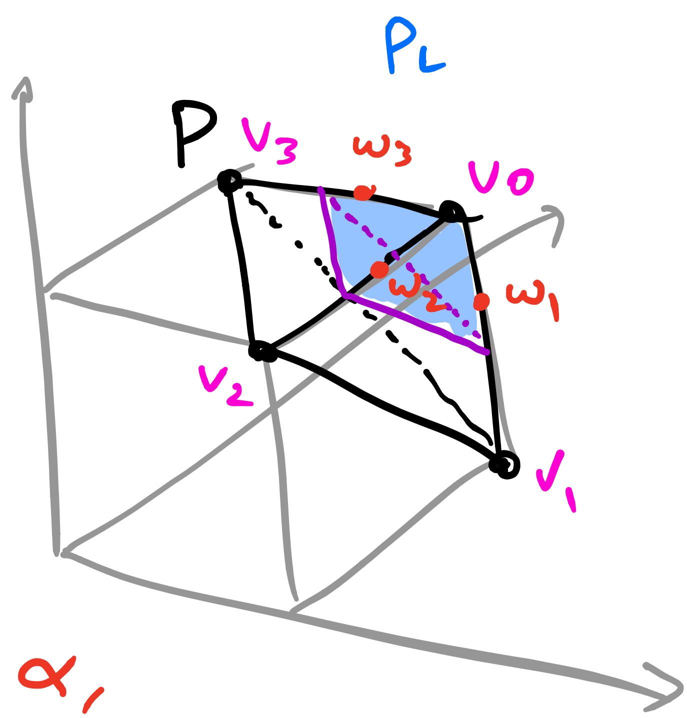


- $v_1 \ldots v_n$ might not be in P_L, but we can "truncate" conv$\{v_0, \ldots v_n\}$.

$$w_i = \begin{cases} v_i & \text{if } c^T v_i \geq L \\ v_0 + \alpha(v_i - v_0) & \text{else} \end{cases}$$

for some small $\alpha > 0$.

e.g. $n = 3$



For some α,

$C := \text{conv}(v_0, w_1 \ldots w_n) \subseteq P_L$

• Can take $\alpha = \frac{1}{2Mn}$,

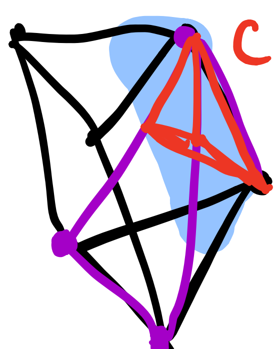
because then

$$c^T w_i = c^T v_0 + \alpha c^T (v_i - v_0)$$

$$= OPT + \alpha c^T (v_i - v_0)$$

$$\geq OPT - \alpha M n$$

$\left[v_i - v_0 \in \{-1, 0, 1\}^n \,\&\, |c_i| \leq M \right.$

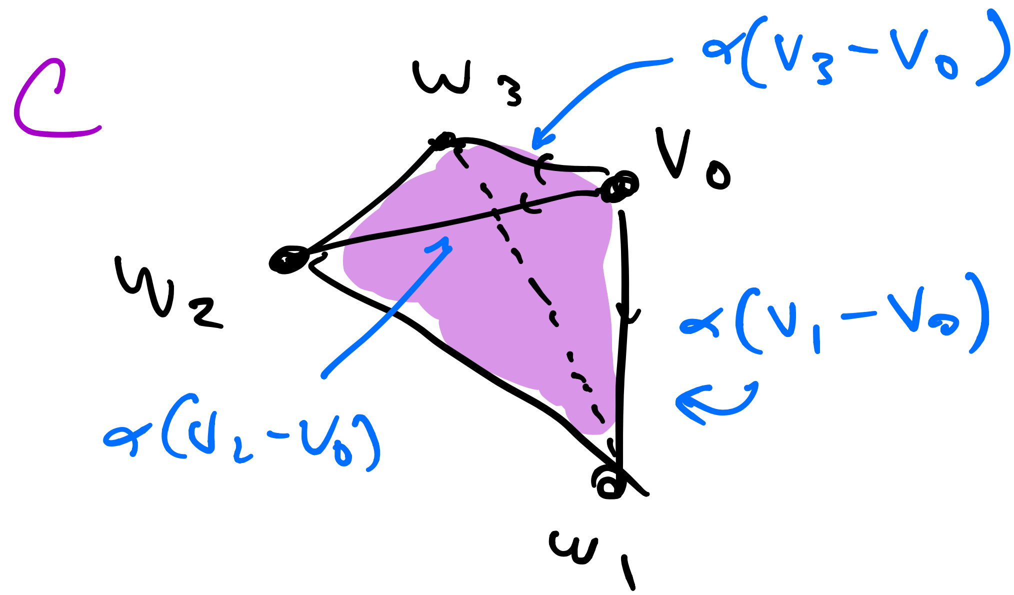


$$\geq \left(L + \tfrac{1}{2}\right) - \frac{1}{2} \geq L.$$

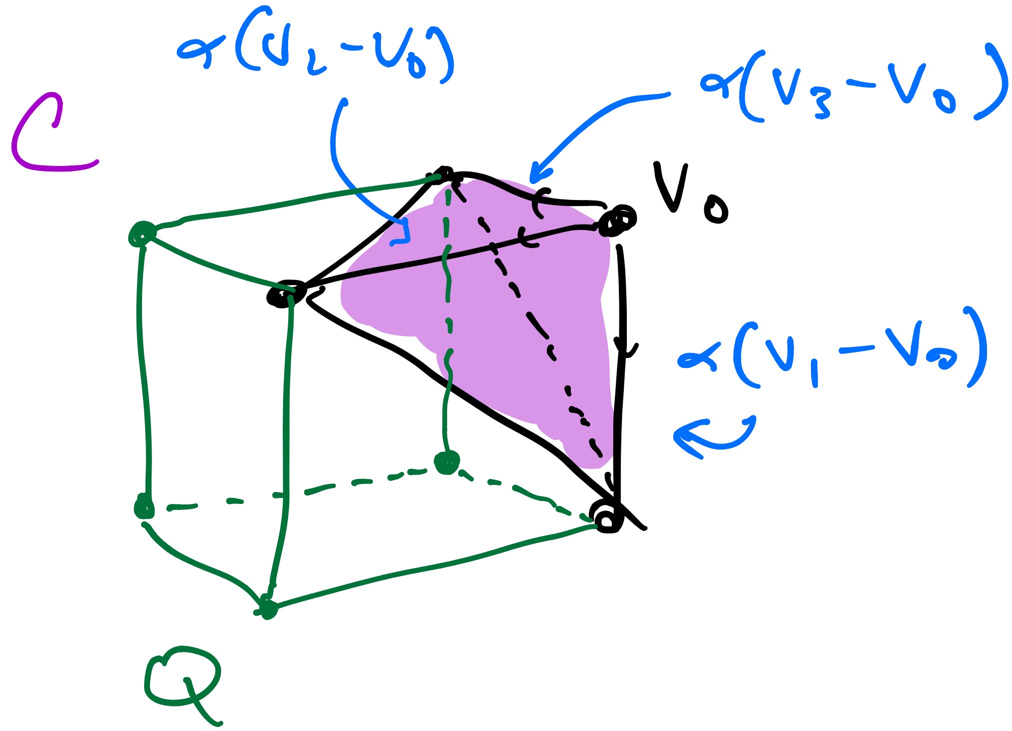
$\Rightarrow w_i \in P_L$.

• Now $\mathrm{vol}\, P_L \geq \mathrm{vol}\, C$,

$C := \operatorname{conv}\{v_0, w_1, \ldots, w_n\}.$



- simplex C = "corner" of parallelipiped Q with sides $\alpha(v_1 - v_0), \ldots, \alpha(v_n - v_0)$.



- $\mathrm{vol}\, C = \frac{1}{n!} \mathrm{vol}(Q)$.

Exercise; WLOG $Q=[0,1]^n$

$C = \{x \in Q : \sum x_i \le 1\}$.

- $\mathrm{vol}\, Q = \alpha^n \mathrm{vol}\, Q'$,

$Q' :=$ parallipiped w/ sides

$(v_1 - v_0), \ldots, (v_n - v_0)$.

- $\mathrm{vol}\, Q' \geq 1$, because sides are lin. indp & they are in $\mathbb{Z}^n$

$$\mathrm{vol}\, Q' = \left| \det \begin{pmatrix} | & & | \\ v_1 - v_0, & \ldots, & v_n - v_0 \\ | & & | \end{pmatrix} \right|$$

$$\geq 1$$

- So together:

and $P_L \geq \frac{1}{n!} \alpha^n \cdot 1 = \frac{1}{n!} \left(\frac{1}{2nM}\right)^n$

$$\geq \frac{1}{n^n} \frac{1}{(2nM)^n} = \frac{1}{(2n^2M)^n}$$

Thus we may take

$$\delta = \frac{1}{(2nM)^n}$$

$\log \frac{1}{\delta} = n \log(2nM)$ ✓.

Overall Runtime

- **# steps of ellipsoid**

$$\leq 2(n+1)\ln\frac{\mathrm{vol}\,E_0}{\delta}$$

$$\leq 2(n+1)\left[\ln\left(n^{n/2}\right)+\ln\left((2n^2M)^n\right)\right]$$

$$=2(n+1)\left[\frac{n}{2}\ln(n)+n\ln(2n^2M)\right]$$

$$=O\left(n^2(\ln n+\ln M)\right).$$

- **# steps of binary search**

$$\leq \log_2(2nM)$$

$$=O(\ln(n)+\log(M))$$

- **Overall:**

$$\boxed{O\left(n^2 (\ln n + \ln M)^2\right)}$$

$= \text{poly}(n, \ln M)$ SEP calls.

To summarize...

Theorem: Given a separation oracle for $P = \text{conv}(X)$, $X \subseteq \{0,1\}^n$, s.t. $\dim P = n$, can max $c^T x$ over P (& hence X) in polynomial time

(in $O(n^2 (\ln n + \ln M)^2)$ calls

to SEP oracle.)

- Side Remark: is not strongly polynomial;

  # iterations depends on c (albeit polynomially).

we could have covered $P = \{x : Ax \leq b\}$ ellipsoid can opt in poly time.

- Éva Tardos '86: can

  Solve LP's $\max\{c^T x : Ax \leq b\}$

  in time poly(input size of A)

  arithmetic ($\times, +, \div$) cost 1 unit.

  (or just poly calls to SEP oracle).

  i.e. indep of c, b!

still uses ellipsoid.

- Thus if $A \in \{-1,0,1\}^{m \times n}$, can solve LP in strongly polynomial time.

but not known for general A!

Example: matroid intersection

- Given $M_1 = (E, \mathcal{I}_1)$, $M_2 = (E, \mathcal{I}_2)$, cost $c: E \to \mathbb{R}$, how to find costliest common indep set?

i.e.
$$\max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} c(S) := \sum_{e \in S} c(e).$$

- Equivalently, maximize $c^T x$ over matroid intersection polytope

$$P_{M_1 \cap M_2} = \mathrm{conv}\{1_S : S \in \mathcal{I}_1 \cap \mathcal{I}_2\}.$$

But how to get a separation oracle?? Exponential # constraints!

- Recall: $P_{M_1 \cap M_2} = P_{M_1} \cap P_{M_2}$
  matroid polytope
- SEP oracle for P_{M_1}, P_{M_2}
  $\Rightarrow$ SEP oracle for $P_{M_1 \cap M_2}$
  (check both P_{M_1}, P_{M_2}.)
- But we only have efficient OPT algorithms for P_{M_1}, P_{M_2}, not SEP!
- From GLS '81, $\exists$ efficient OPT algo.
  $\Rightarrow \exists$ efficient SEP algorithm!

- Thus $\exists$ efficient SEP algos. for P_{M_1}, P_{M_2}, $\Rightarrow$ $\exists$ efficient SEP $P_{M_1} \cap P_{M_2}$.
  $\Rightarrow$ ellipsoid can optimize in poly time.

Example: nonbip. matching

- Given G, cost $c : E \to \mathbb{R}$, find max cost matching M.

- Equivalently, optimize $c^T x$ over matching polytope

$$P = \text{conv}(\{\mathbb{1}_M : M \text{ matching in } G\}).$$

- Recall: Matching polytope given by

$$P = \left\{ x \in \mathbb{R}^E : \sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in V, \right.$$

degree constr.

$$\sum_{e \in E(S)} x_e \leq \frac{|S|-1}{2} \quad \forall S \subseteq V,\ |S| \text{ odd}$$

odd set constr

$$\left. x_e \geq 0 \quad \forall e \in E. \right\}$$

- P is full-dimensional

  Exercise.

- However, separation oracle nontrivial! (Exp. constraints again!)

- Can implement using min T-odd cut algorithm (Padberg-Rao).

Matching polytope SEP oracle:

- Check degree constraints; if violated, return corresp inequality.

- Next check odd set constraints.

How?

▷ For x satisfying degree constraints, need to decide if

$$\sum_{e \in E(S)} x_e \le \frac{|S|-1}{2}$$

$\forall S \subseteq V$, $|S|$ odd, & if not produce violated S.

▷ Assume WLOG $|V|$ even

(else add isolated vertex).

▷ For $v \in V$, define

$$s_v = 1 - \sum_{e \in \delta(v)} x_e$$

▷ **observe**: Given $S \subseteq V$,

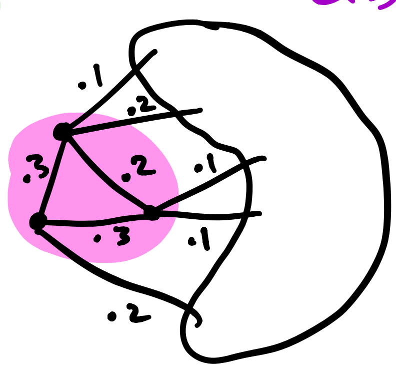
$$\sum_{v \in S} s_v + \sum_{e \in \delta(S)} x_e = |S| - 2 \sum_{e \in E(S)} x_e .$$

Pf

$$|S| - \underbrace{\sum_{v \in S} \sum_{\delta(v)} x_e}_{e \in E(S) \text{ double counted.}} + \sum_{\delta(S)} x_e = |S| - 2\sum_{E(S)} x_e \cancel{- \sum_{\delta(S)} x_e + \sum_{\delta(S)} x_e}$$

e.g.

S



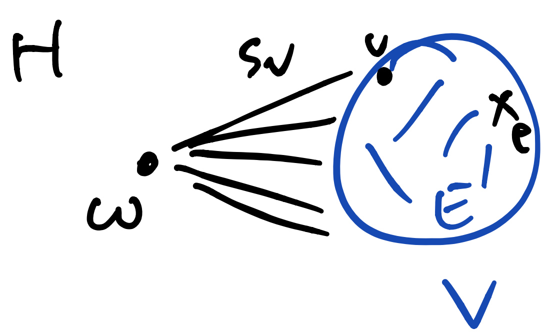
Thus odd set constr. holds for S $\iff$ $\sum_{v \in S} s_v + \sum_{e \in \delta(S)} x_e \geq 1$

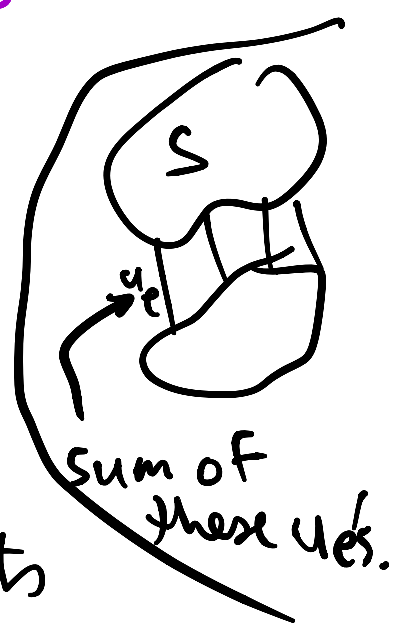
▷ Define new graph H with

vertex set = V + new vert. w

edge set = E + all edges $w \leftrightarrow V$.

<u>picture</u>:





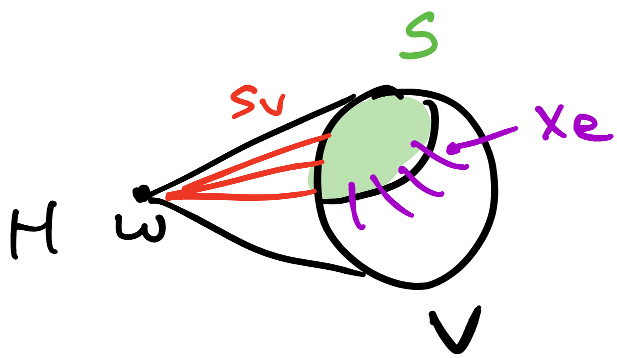
▷ Define edge weights

$$u_e = \begin{cases} x_e & \text{if } e \in E \\ s_v & \text{if } e = (v,w). \end{cases}$$

▷ For a cut S in H, may assume $w \notin S$ by taking complements.

▷ cut $S \subseteq V$ in H has value

$$\sum_{e\in\delta_H(S)} u_e = \sum_{v\in S} s_v + \sum_{e\in\delta(S)} x_e$$



▷ Thus $\displaystyle\sum_{v\in S} s_v + \sum_{e\in\delta(S)} x_e \geq 1$

$\forall$ odd $S \subseteq V$ $\iff$

min V-odd cut in H has value ≥ 1.

Recall: for T even, min T-odd cut is to find min cut S subject to $|S\cap T|$ odd.

- we have seen how to compute min T-odd cut; do so for $T = V$ in H.
  - if $\exists$ V-odd cut S for H w/ value < 1, S is violated; return S
  - if not, $x \in P$. $\square$