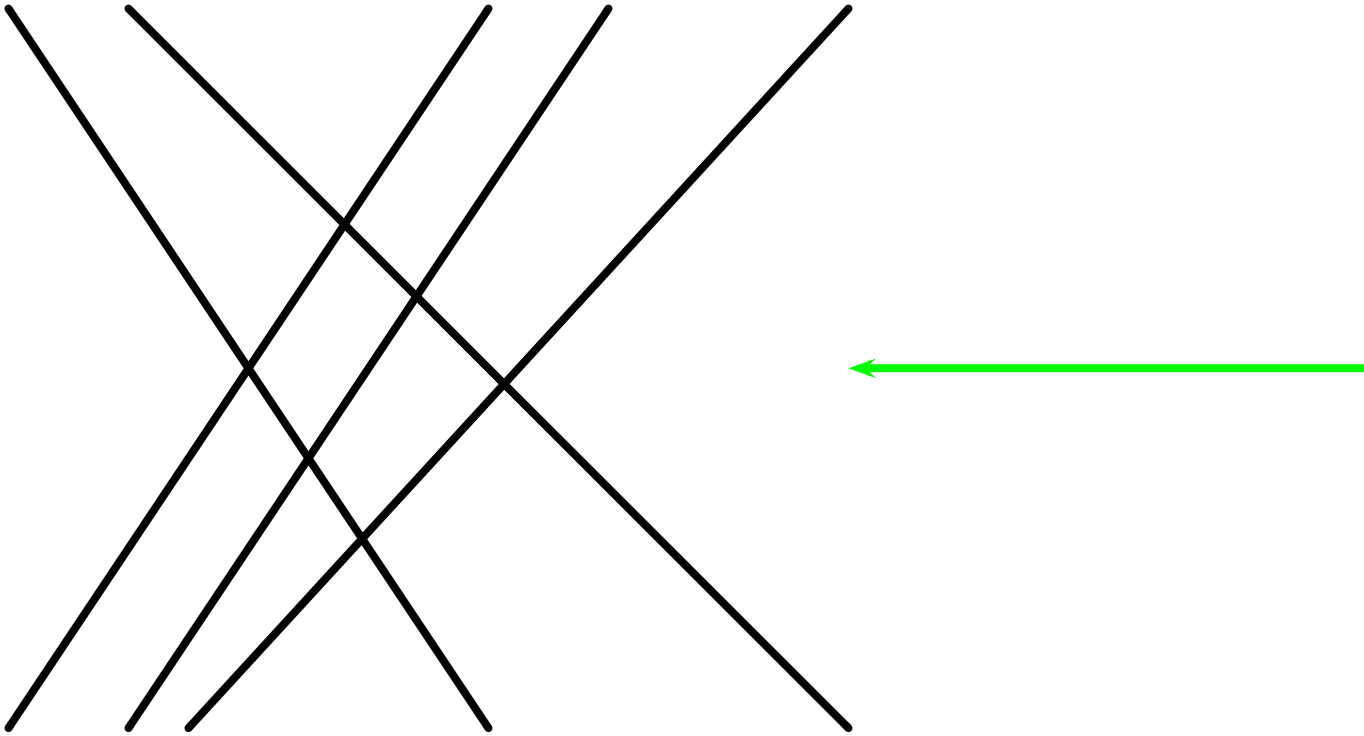
Linear Programming in Subexpon
Time

Linear Programming Algebraic V

for $x, c \in \mathbb{R}^d$; $b \in \mathbb{R}^n$; $A \in \mathbb{R}^{(n \times d)}$

min $c^T x$

subject to $Ax \leq b$



Goals

The Holy Grail

- polynomial time algorithm in unit-cost model

Previous Work

- polynomial time algorithm in bit-cost model
- $O(d!n)$ time algorithm in unit-cost model (Seidel)

Today

- subexponential time algorithm in unit-cost model

Outline

- SolveLP1 — Clarkson (1988)
- SolveLP2 — Clarkson (1988)
- BasisLP — Matousek, Sharir and Welzl (1992
  also Kalai (1992)

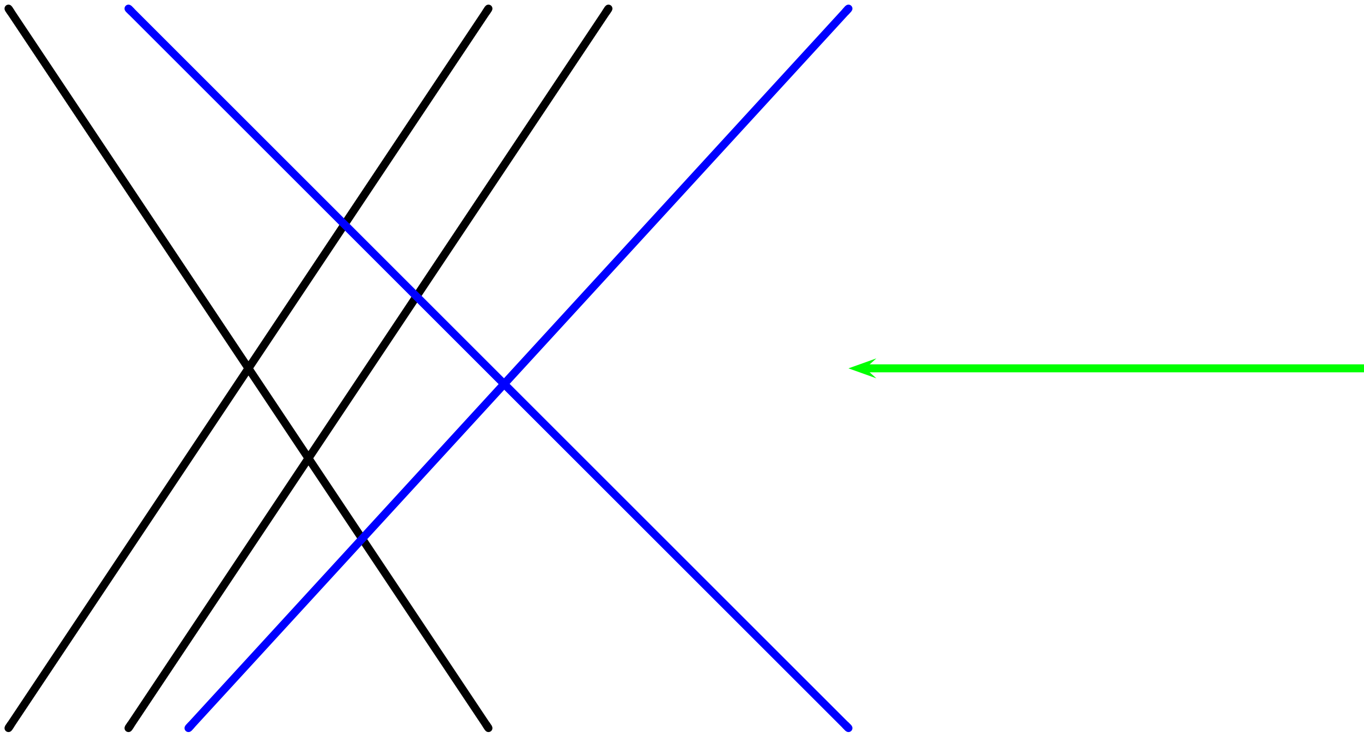
Combined Result: $O(d^2 n + e^{O(\sqrt{d \log d})})$

Definitions

- H . . . . set of n linear constraints.
- $v(H)$ . . . . optimum point subject to H.
- a basis . . . . minimal set B with $v(B) > -\infty$.
- a basis for H . . . basis $B \subseteq H$ s.t. $v(B) = v(H)$.

Note: the size of a basis is d.

Example



Random Sampling

- Randomly reduce large problem to small subprob
- Solve small subproblem.
- If solution solves large problem, you win.
- Otherwise, get hints from wrong solution and try

Algorithm: SolveLP1

SolveLP1(H)

$G := \emptyset$

repeat

$R :=$ random subset of H of size r

$B :=$ SolveLP1($G \cup R$)

$V :=$ set of constraints violated by $v(B$

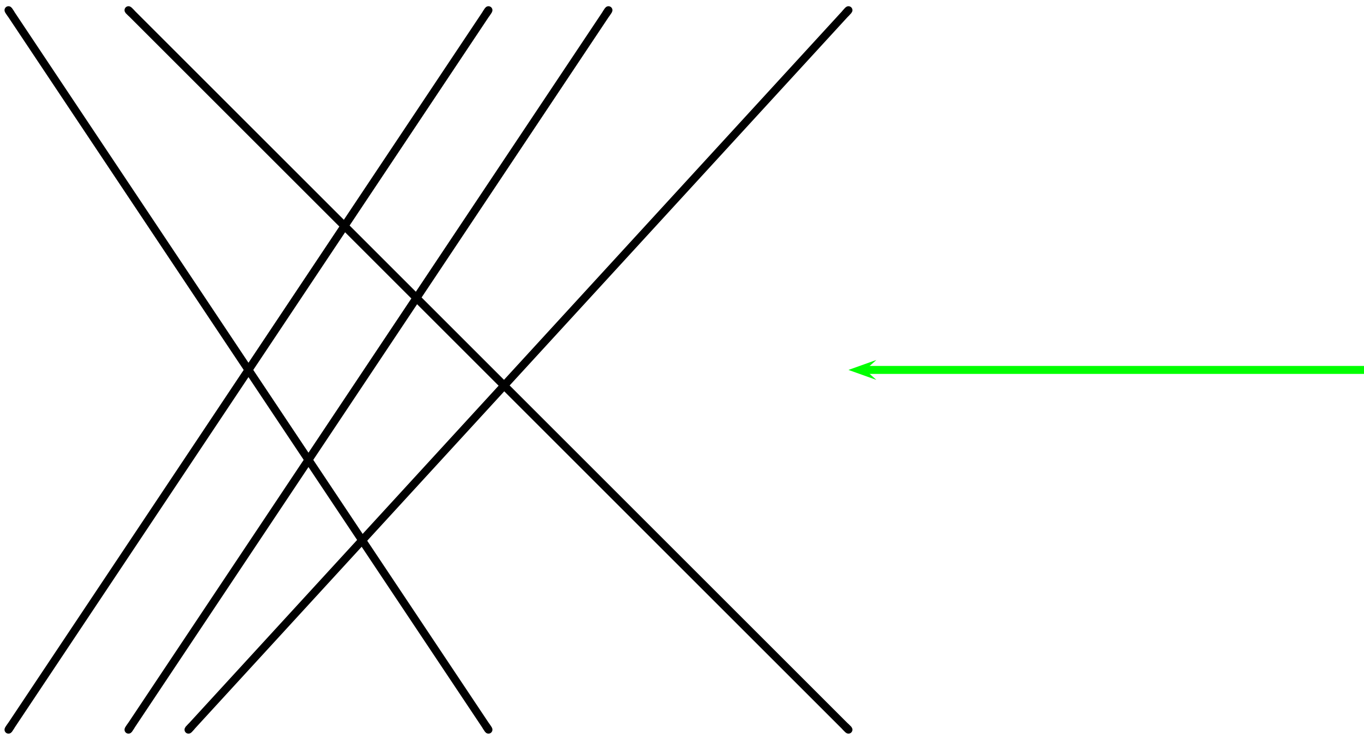
if $|V|$ is small enough, then $G := G \cup V$

until $V = \emptyset$

Example

$r = 2$, "small enough" $= 1$

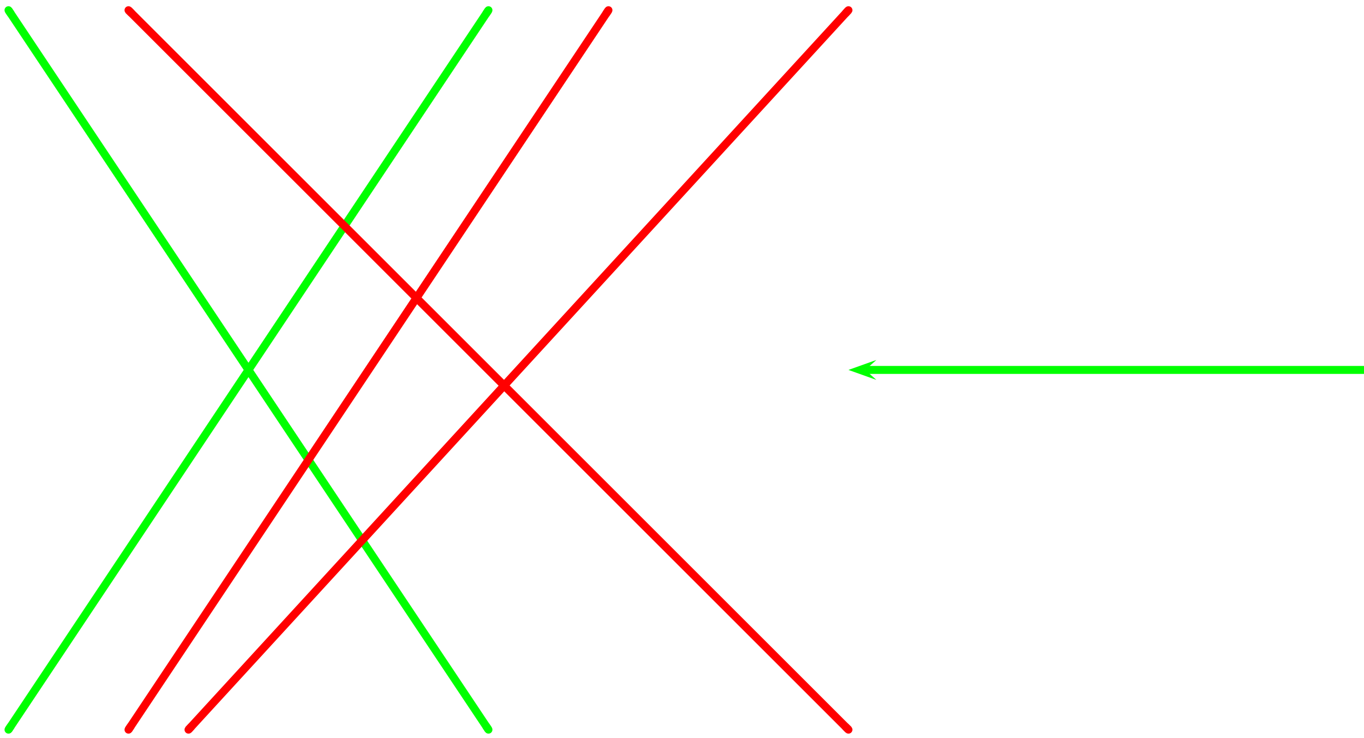
key: R = green V = red G = blue



Example

$r = 2$, “small enough” $= 1$

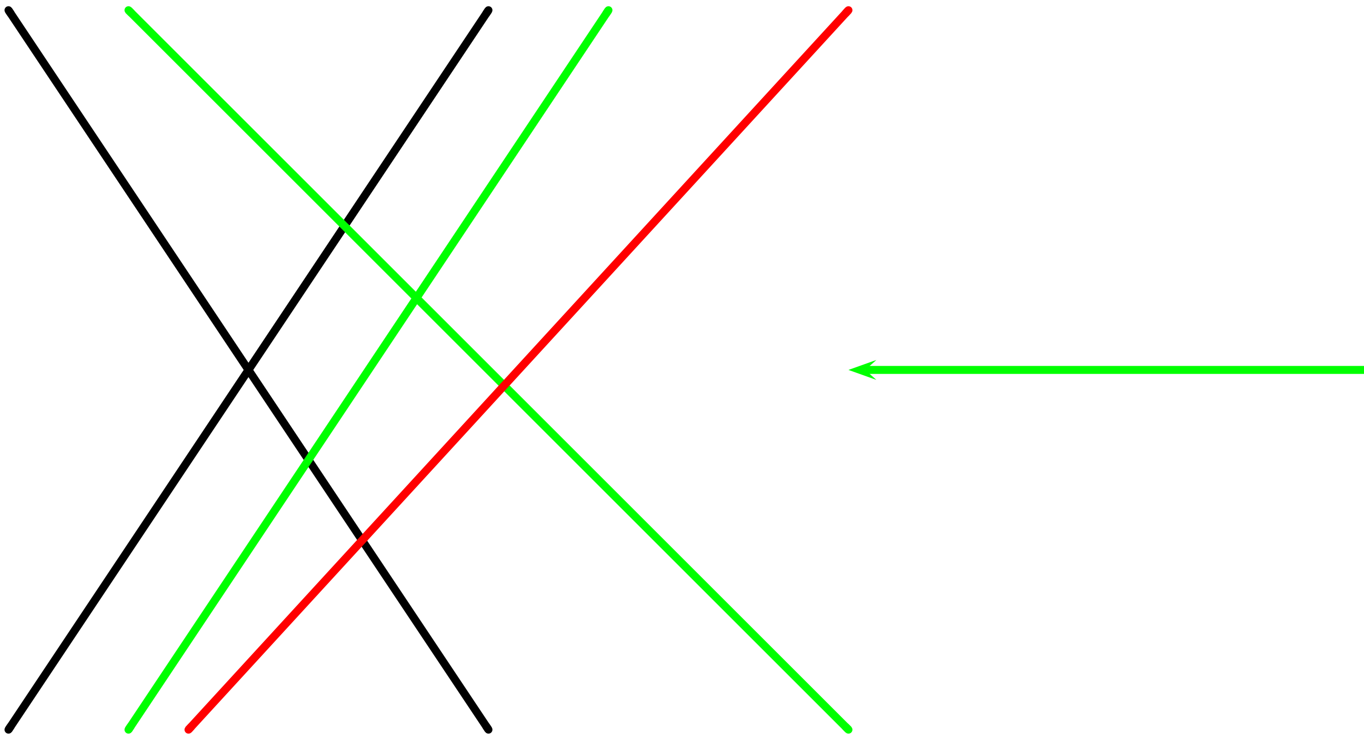
key: R = green V = red G = blue



Example

$r = 2$, "small enough" $= 1$

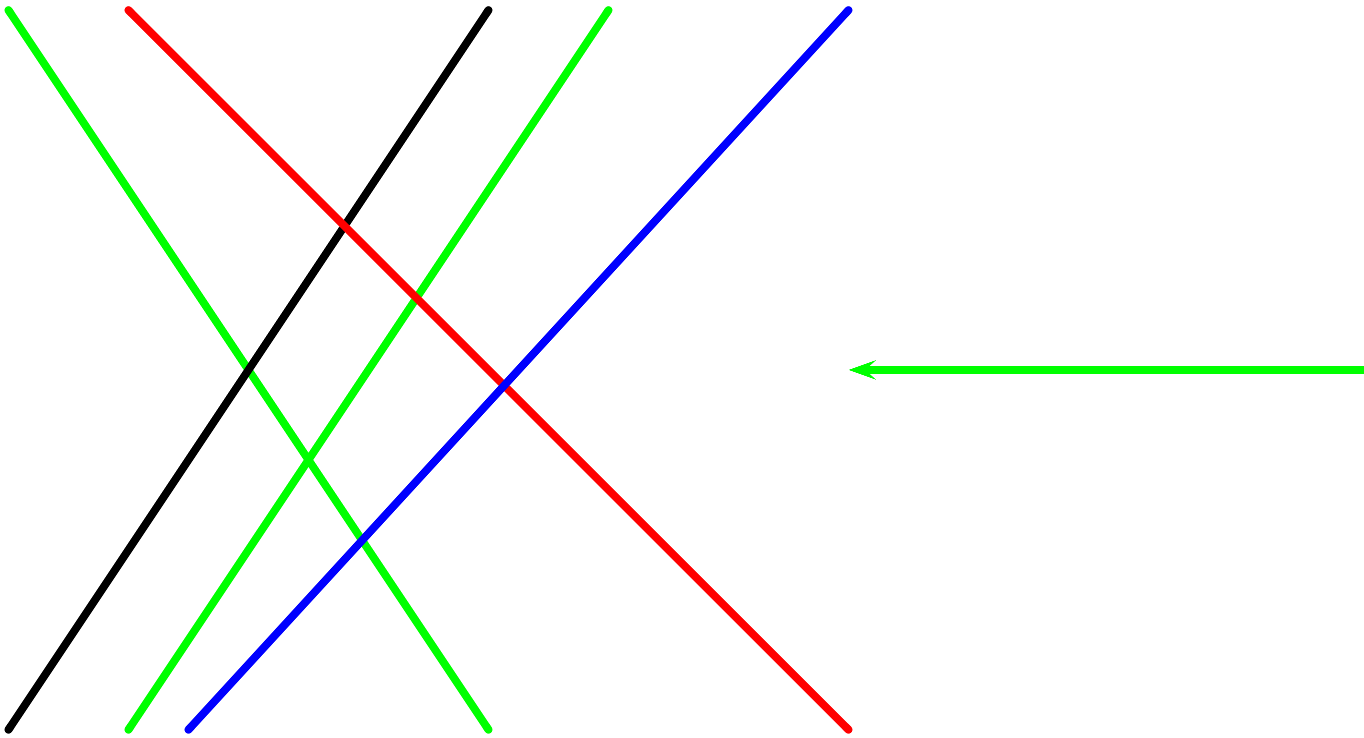
key: R = green   V = red   G = blue



Example

$r = 2$, "small enough" $= 1$

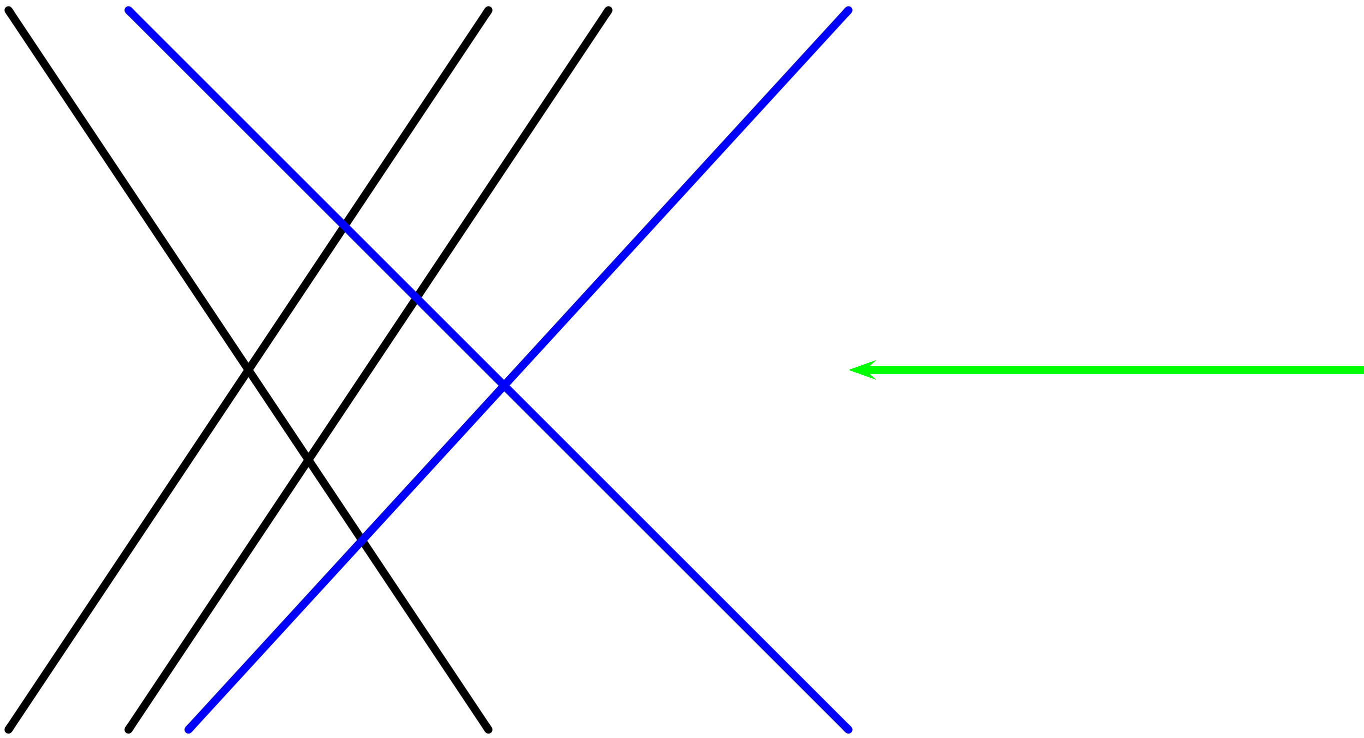
key: R = green V = red G = blue



Example

$r = 2$, "small enough" $= 1$

key: R = green V = red G = blue



Analysis

Bounding the number of iterations until a succe

- number of violated constraints is nd/r

$$E[|V|] = n\,\mathsf{Pr}(constraint\ is\ violated) \leq n(d/(r+$$

$$\boxed{\text{“small enough”} := 2nd/r}$$

- number of iterations until a successful one is 2

$$\mathsf{Pr}(|V| \geq 2E[|V|]) < 1/2$$

Algorithm: SolveLP1

SolveLP1(H)

$G := \emptyset$

repeat

$R :=$ random subset of H of size r

$B :=$ SolveLP1($G \cup R$)

$V :=$ set of constraints violated by $v(B$

if $|V| \leq 2nd/r$, then $G := G \cup V$

until $V = \emptyset$

Analysis

Bounding the number of successful iterati

If no basis constraint is violated,
then the solution must be the optimal solutio

In every successful iteration,
a basis constraint is added to G!

➡ number of successful iterations is d

Analysis

Running time

- number of constraints in recursive call is at most

$$r + |G| \le r + d|V| \le r + 2nd^2/r$$

- running time is

$$T_1(n) \le 2dT_1(r + 2nd^2/r) + O(d^2n)$$

$$\boxed{r{:=}d\sqrt{2n}}$$

- base case is $n = O(d^2)$

$$n \le d\sqrt{2n} \Longrightarrow n \le 2d^2$$

Algorithm: SolveLP1

SolveLP1(H)

if $n \leq 2d^2$, then return solution

$G := \emptyset$

repeat

$R :=$ random subset of H of size $d\sqrt{2n}$

$B :=$ SolveLP1($G \cup R$)

$V :=$ set of constraints violated by $v(B)$

if $|V| \leq \sqrt{2n}$, then $G := G \cup V$

until $V = \emptyset$

Analysis with Seidel

SolveLP1(H)

if $n \leq 2d^2$, then return Seidel(H)

$G := \emptyset$

repeat

$R :=$ random subset of H of size $d\sqrt{2n}$

$B :=$ Seidel($G \cup R$)

$V :=$ set of constraints violated by $v(B)$

if $|V| \leq \sqrt{2n}$, then $G := G \cup V$

until $V = \emptyset$

Analysis with Seidel

Plug in Seidel

$$S(n, d) = O(d!n)$$

➡ running time is $O(d^2 n + d^2 d! \sqrt{n})$

$$\begin{aligned} T(n) &= 2dS(d\sqrt{n}, d) + O(d^2 n) \\ &= O(d^2 n + d^2 d! \sqrt{n}) \end{aligned}$$

Improving the Algorithm

Reduce sample size

Instead of forcing the violated constraints,
include them in random sample with higher proba

Algorithm: SolveLP2

H is a multiset

SolveLP2(H)

repeat

R := random subset of H of size r

B := SolveLP2(R)

V := multiset of constraints violated by

if $|V|$ is small enough, then $H := H \cup V$

until $V = \emptyset$

Analysis

Bounding the number of iterations until a succe

- number of violated constraints is $|H|d/r$

“small enough” $:= 2|H|d/r$

- number of iterations until a successful one is 2

Algorithm: SolveLP2

H is a multiset

SolveLP2(H)

repeat

R := random subset of H of size r

B := SolveLP2(R)

V := multiset of constraints violated by v

if $|V| \leq 2|H|d/r$, then $H := H \cup V$

until $V = \emptyset$

Analysis

Bounding the number of successful iterati

Fix a basis B of H.

- in each iteration, multiplicity of a constraint in B
- after kd iterations, some constraint in B has mult

at least 2^k

- in each iteration, size of H increases by $2|H|d/r$
- after kd iterations, $|H|$ is at most $n\exp(2d^2/r)^k$

$$|H| \leq n(1+2d/r)^{kd} < n\exp(2kd^2/r)$$

$$\boxed{2^k < n\exp(2d^2/r)^k}$$

Analysis

Bounding the number of successful iterati

$$\boxed{2^k < n \exp(2d^2/r)^k}$$

- algorithm will terminate when inequality is violate
- minimize r subject to $2 \geq \exp(2d^2/r)$

$$\boxed{r = O(d^2)}$$

- minimize k subject to $2^k \geq n e^{k/2}$

$$\boxed{k = O(\log n)}$$

Algorithm: SolveLP2

H is a multiset

SolveLP2(H)

repeat

R := random subset of H of size $4d^2$

B := SolveLP2(R)

V := multiset of constraints violated by

if $|V| \leq |H|/2d$, then $H := H \cup V$

until $V = \emptyset$

Running Time

- running time is

$$T_2(n) \le O(d \log n T'(d^2) + d^2 n \log n)$$

- base case is $n = O(d^2)$

Algorithm: SolveLP2

H is a multiset

SolveLP2(H)

repeat

R := random subset of H of size $4d^2$

B := solution for R

V := multiset of constraints violated by v

if $|V| \leq |H|/2d$, then $H := H \cup V$

until $V = \emptyset$

Analysis with SolveLP1 and Seid

SolveLP1(H)

if $n \leq 2d^2$, then return SolveLP2(H)

$G := \emptyset$

repeat

$R :=$ random subset of H of size $d\sqrt{2n}$

$B :=$ SolveLP2($G \cup R$)

$V :=$ set of constraints violated by $v(B$

if $|V| \leq \sqrt{2n}$, then $G := G \cup V$

until $V = \emptyset$

Analysis with SolveLP1 and Seid

H is a multiset

SolveLP2(H)

repeat

$R :=$ random subset of H of size $4d^2$

$B :=$ Seidel(R)

$V :=$ multiset of constraints violated by

if $|V| \leq |H|/2d$, then $H := H \cup V$

until $V = \emptyset$

Analysis with SolveLP1 and Seid

Plug in Seidel and plug into SolveLP1

$$S(n,d) = O(d!n)$$

➡ running time is $O(d!d^3 \log n + d^3\sqrt{n}\log n + d^2 n)$

$$\begin{aligned}
T(n) &= O(dT_2(d\sqrt{n}) + d^2 n) \\
&= O(d(d \log n S(d^2, d) + d^2\sqrt{n}\log n) + d \\
&= O(d!d^3 \log n + d^3\sqrt{n}\log n + d^2 n)
\end{aligned}$$

Solving Small LPs

Seidel(H)

if $|H| = d$, then return H

else

Pick a random constraint $c \in H$

$B :=$ Seidel($H - \{c\}$)

if $v(B)$ does not violate c, then return

else project constraints onto c and solv

running time is $O(d!n)$

$$S(n, d) = S(n-1, d) + (d/n)S(n-1, d-1) + O$$

Algorithm: BasisLP

Modify Seidel!

BasisLP(H,B)

Pick a random constraint in $c \in H - B$

B := BasisLP($H - \{c\}$,B)

if $v(B)$ does not violate c, then return B

else return BasisLP(H,Basis($B \cup \{c\}$))

Analysis Idea

Define a parameter that captures the quality of th

- hint quality k is (roughly) $d - |B_{hint} \cap B_{real}|$

$$T(n,k) \leq T(n-1,k) + \frac{1}{n-d}\sum_{i=1}^{k}(1 + T(n, k -$$

- running time is $O\left(nd\exp\left(\sqrt{d\ln(n+1)}\right)\right)$

Conclusion

Use BasisLP as base of recursion!

➡ running time is $O(d^2 n + e^{O(\sqrt{d \log d})})$