Linear Programming

“program” as in “exercise program” or “spending program”, not “C program”

optimization problem with linear inequalities

variables $x_1 \cdots x_d$ in d-dimensions.

$$\max \quad c_1 x_1 + c_2 x_2 + \cdots + c_d x_d$$

$$\text{s.t.} \quad \begin{aligned} a_{11} x_1 + a_{12} x_2 + \cdots + a_{1d} x_d &\le b_1 \\ &\vdots \\ a_{n1} x_1 + \cdots + a_{nd} x_d &\le b_n \end{aligned}$$

i.e. $\max \ cx$

$Ax \le b$

c $1 \times d$ vector

x $d \times 1$ vector

A $n \times d$ matrix

b $n \times 1$

An application: planning menus.

	apple	brocolli		milk
d foods			...	
	1	2		d
each with cost	c_1	c_2	...	c_d

	protein	vitamin D		
n nutrients	1	2	...	n
each with daily requirement	b_1	b_2	...	b_n

a_{ij} – amount of nutrient i in food j

Buy food to meet daily requirements, min cost

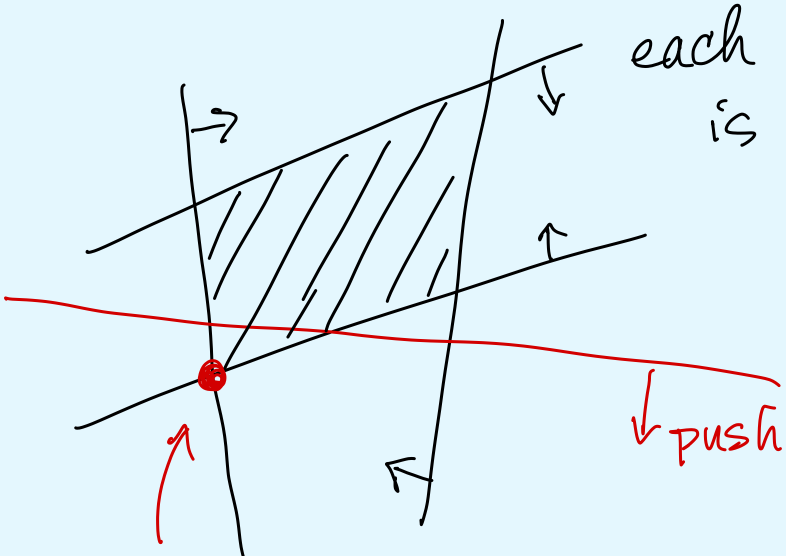
$$\min cx$$
$$Ax \geq b$$

variables $x_1 \cdots x_d$

x_j = amount of food j to buy.

picture in 2D

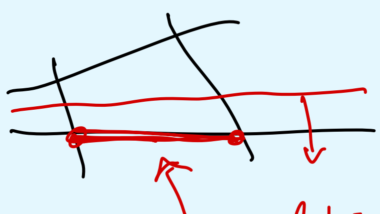
each constraint $a_1 x_1 + a_2 x_2 \le b$
is a half-space



push as far as possible

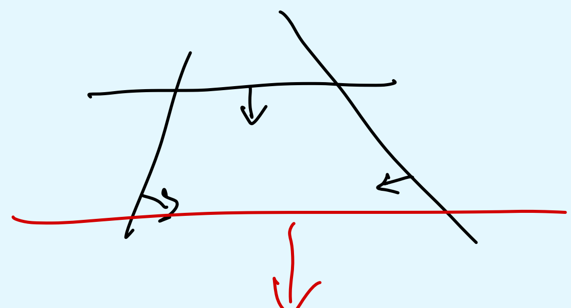
optimum solution.

opt. solution is at a vertex
except



multiple opt.

use tie-breaking to avoid this



unbounded

Straightforward algorithm:

try **all** vertices, see which gives max

From last day: this is the dual problem to Convex Hull and can be solved by same algorithms

$O(n \log n)$ in 2D, 3D
$O(n^{\lfloor d-1/2 \rfloor})$ for $d \geq 4$

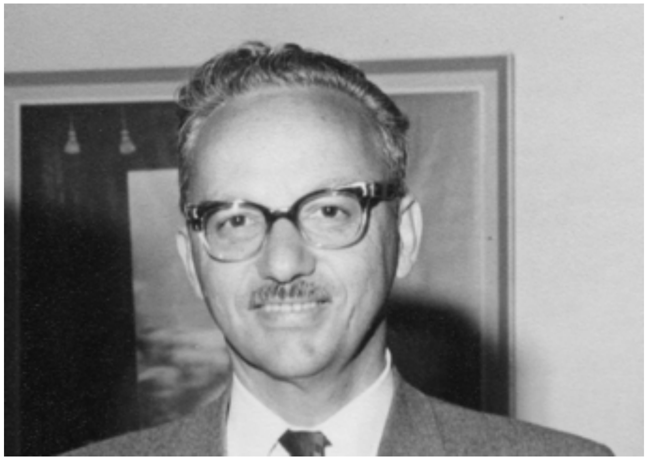
But we don't really want all the vertices, so we can do better.

History

early 40’s, 50’s

George Dantzig

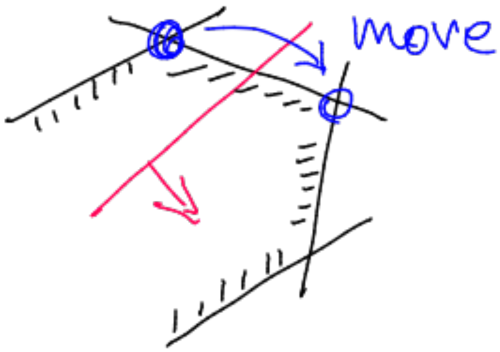
- simplex method in the ’40’s



W https://en.wikipedia.org/wiki/George_Dantzig

Simplex Method

- geometrically — walk from one vertex of the feasible region to an adjacent one
- Simplex pivot rule
  - which inequality to remove
  - which one to add



a great intro to linear programming:

Understanding and **using linear programming**
J Matousek, B Gärtner - 2007

https://ocul-wtl.primo.exlibrisgroup.com/permalink/01OCUL_WTL/5ob3ju/alma9953153109505162

History

OPEN: is there a pivot rule that gives a polynomial time algorithm?

But the simplex algorithm is very good in practice.

Related question:

Given initial vertex s and final vertex t (on a convex polyhedron), how many edges on the shortest path from s to t?

diameter of the polyhedron = worst case over all s and t

Hirsch conjecture. W https://en.wikipedia.org/wiki/Hirsch_conjecture
The diameter of a convex polyhedron is $\leq n - d$
where n = number of inequalities, d = dimension

disproved in 2012, $d = 43$.

But there could still be a polynomial (or even linear) bound.

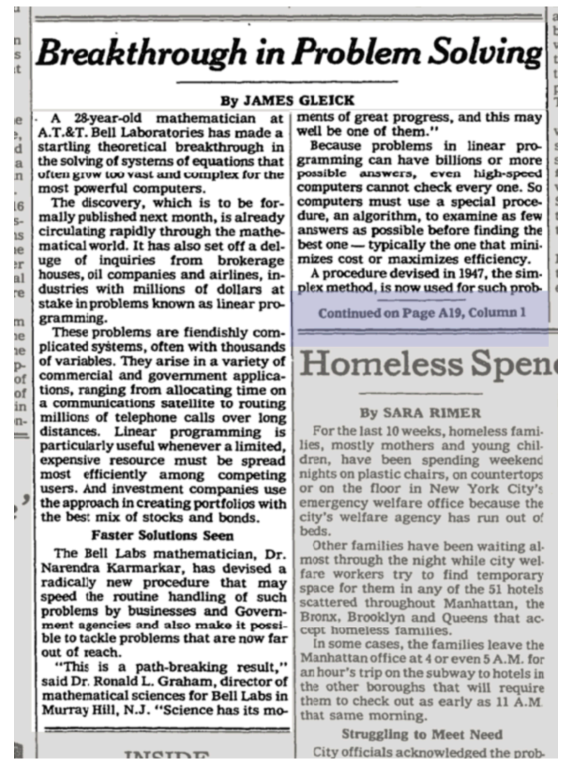
History

Polynomial time algorithms for Linear Programming:

’80 — Katchian, ellipsoid method

’84 — Karmarkar, interior point method

front page NYT 1984



Breakthrough in Problem Solving

By JAMES GLEICK

A 28-year-old mathematician at A.T.&T. Bell Laboratories has made a startling theoretical breakthrough in the solving of systems of equations that often grow too vast and complex for the most powerful computers.

The discovery, which is to be formally published next month, is already circulating rapidly through the mathematical world. It has also set off a deluge of inquiries from brokerage houses, oil companies and airlines, industries with millions of dollars at stake in problems known as linear programming.

These problems are fiendishly complicated systems, often with thousands of variables. They arise in a variety of commercial and government applications, ranging from allocating time on a communications satellite to routing millions of telephone calls over long distances. Linear programming is particularly useful whenever a limited, expensive resource must be spread most efficiently among competing users. And investment companies use the approach in creating portfolios with the best mix of stocks and bonds.

Faster Solutions Seen

The Bell Labs mathematician, Dr. Narendra Karmarkar, has devised a radically new procedure that may speed the routine handling of such problems by businesses and Government agencies and also make it possible to tackle problems that are now far out of reach.

“This is a path-breaking result,” said Dr. Ronald L. Graham, director of mathematical sciences for Bell Labs in Murray Hill, N.J. “Science has its moments of great progress, and this may well be one of them.”

Because problems in linear programming can have billions or more possible answers, even high-speed computers cannot check every one. So computers must use a special procedure, an algorithm, to examine as few answers as possible before finding the best one — typically the one that minimizes cost or maximizes efficiency.

A procedure devised in 1947, the simplex method, is now used for such prob-

Continued on Page A19, Column 1

Homeless Spen

By SARA RIMER

For the last 10 weeks, homeless families, mostly mothers and young children, have been spending weekend nights on plastic chairs, on countertops or on the floor in New York City's emergency welfare office because the city's welfare agency has run out of beds.

Other families have been waiting almost through the night while city welfare workers try to find temporary space for them in any of the 51 hotels scattered throughout Manhattan, the Bronx, Brooklyn and Queens that accept homeless families.

In some cases, the families leave the Manhattan office at 4 or even 5 A.M. for an hour's trip on the subway to hotels in the other boroughs that will require them to check out as early as 11 A.M. that same morning.

Struggling to Meet Need

City officials acknowledged the prob-

these operate on the bit representations of the numbers

OPEN: an LP algorithm that uses number of arithmetic operations polynomial in *n* and *d* , “strongly polynomial time”

“smoothed analysis” explains the good behaviour of the simplex method

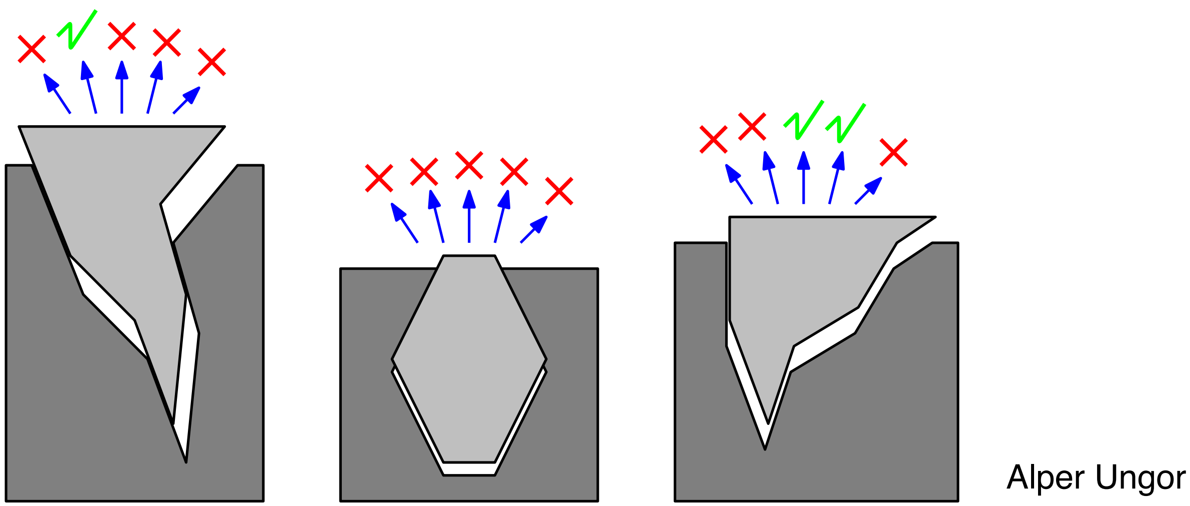
W https://en.wikipedia.org/wiki/Smoothed_analysis

The simplex algorithm is NP-mighty d https://doi.org/10.1145/3280847
Y Disser, M Skutella - ACM Transactions on Algorithms (TALG), 2018 - dl.acm.org
We show that the Simplex Method, the Network Simplex Method—both with Dantzig's original pivot rule—and the Successive Shortest Path Algorithm are NP-mighty. That is, each of these algorithms can be used to solve, with polynomial overhead, any problem in NP implicitly during the algorithm's execution. This result casts a more favorable light on these algorithms' exponential worst-case running times. Furthermore, as a consequence of our approach, we obtain several novel hardness results. For example, for a given input to the …

Linear programming in small (fixed) dimension *d*

Application: Casting (from [CGAA]). Make a 3D object in a mold

picture in 2D



Pour liquid into a mold, harden, and then remove by straight line motion in some direction. Find a direction that works.

For a given top face, this can be expressed as linear programming.
Try all top faces.

Linear programming in small (fixed) dimension

Megiddo 1984, algorithm with runtime O(n)

but the dependence on d is bad $O(2^{2^d} n)$

Seidel 1991, randomized algorithm with expected runtime O(n)

dependence on d $O(d!\, n)$

comparing 2^{2^d} vs $d!$

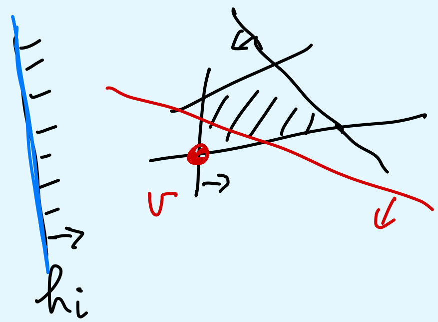
take logs 2^d $d \log d$

Randomized Incremental Algorithm in 2D, Seidel

Idea: add the halfplanes one by one in random order, updating the optimum solution vertex *v* each time

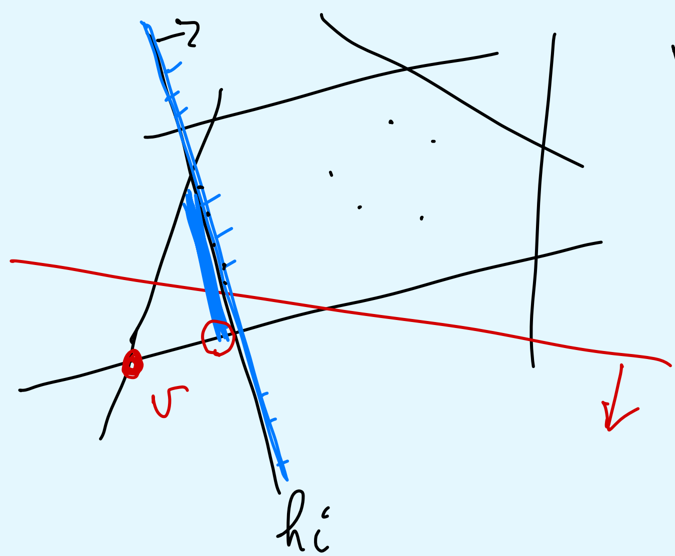
To add h_i. Two cases:

1



$v \in h_i$ — no update to v.

2



$v \notin h_i$

New opt. vertex will lie on the line l_i of h_i

So solve a 1-dim. LP problem along line l_i

We reduced to 1D Linear Programming.
What is 1D Linear Programming?

max x (the constant is irrelevant)

subject to

inequalities like $x \le 2$

$-1 \le x$

$\vdots$

$x \le 5$

(soln is $x = 2$)

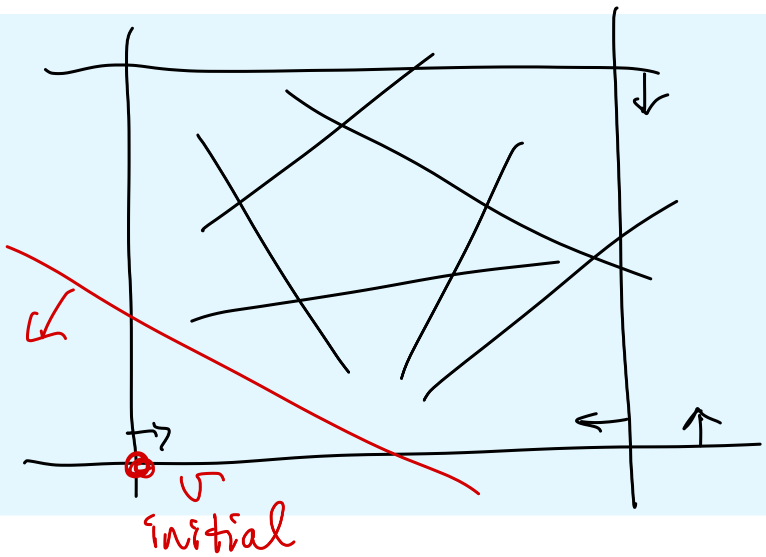
feasible region is intersection of intervals = one interval

easy to do $O(n)$

Some issues:

What is the initial vertex (when there are no halfplanes)?

What if the LP is “unbounded”, (e.g. max x, $x \geq 0$)



Add a large box.

Intialize vertex v to optimum vertex of box

If final v on boundary of box then original LP was unbounded.

The method also needs the optimum to be unique — handle this by asking for optimum, then (to break ties) the lexicographically largest
(i.e. max x_1, then max x_2, . .)

Algorithm LP_2 (H , c) $H_n = \{h_1, \ldots, h_n\}$, a set of halfplanes, c = objective

1. add large box; initialize v to optimum vertex of box (wrt c)
2. take random order $h_1, \ldots, h_n$
3. for $i = 1 \ldots n$ # add h_i
4. suppose h_i is $a_1 x_1 + a_2 x_2 \leq b$
5. if $v \notin h_i$ (i.e. $a_1 v_1 + a_2 v_2 > b$) then
6. # solve the problem restricted to the line $a_1 x_1 + a_2 x_2 = b$
7. $\{h'_1, \ldots, h'_{i-1}\}$, c' := use the equation to eliminate one variable from $\{h_1, \ldots, h_{i-1}\}$, c
8. $v := LP_1(\{h'_1, \ldots, h'_{i-1}\}, c')$

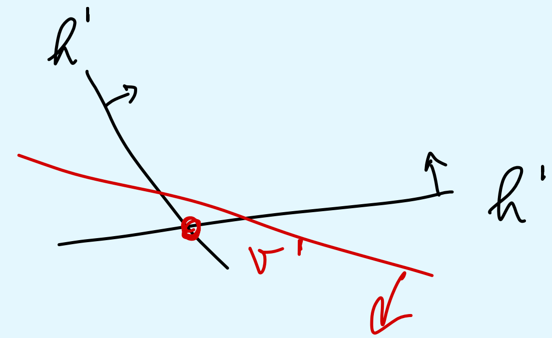
Worst case run time: $O(n^2)$ line 7, 8 each take $O(n)$
recall d is constant

Exercise: Show that the worst case can happen.

Expected runtime Prove expected run time is $O(n)$

use backwards analysis

After we add h_i. Suppose opt. is vertex v' at intersection of h', h''



We already $h_1 \cdots h_i$

So $h', h'' \in \{h_1 \cdots h_i\}$

How much work for h_i was it case 1 (no work) or case 2 (call LP_1)

Case 2 iff h_i is h' or h''.

$$\text{Prob}\{h_i = h' \text{ or } h_i = h''\} = \frac{2}{i}$$

* because h_i is equally likely to be any of the i inequalities inserted so far.

Expected total work $\displaystyle\sum_{i=1}^{n} \frac{2}{i} \underbrace{O(i)}_{\text{work for } LP_1 \text{ on } h_1 \cdots h_i} = O(n)$ $\boxtimes$

Note: degeneracy OK.

In higher dimensions

$\frac{2}{i}$ becomes $\frac{d}{i}$ because it takes *d* hyperplanes to specify a vertex

run time recurrence: $T_d(n) = T_d(n-1) + \frac{d}{n} O(T_{d-1}(n))$

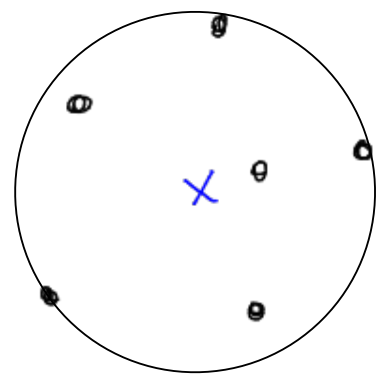
solution is: $T_d(n) = O(d!\, n)$

Smallest enclosing disc

Not a linear programming problem, but amenable to the same approach

Given points $p_1, \ldots, p_n \in \mathbb{R}^d$

find the smallest enclosing sphere.



smallest enclosing disc in 2D

This is a facility location problem — the center of the disc minimizes the maximum distance to all points.

Natural formulation gives quadratic programming.

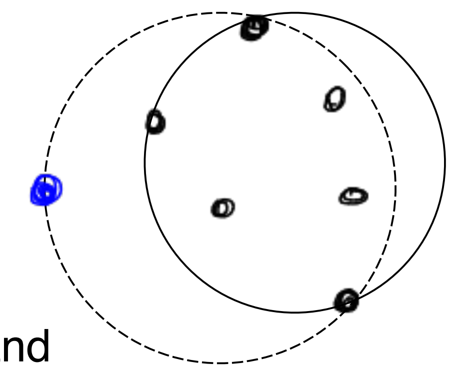
Megiddo's approach gives O(n) but bad constant.

Randomized incremental approach, Welzl, 1991.

note that the smallest disc will go through 3 points

Smallest enclosing disc. Randomized incremental algorithm.
Suppose we have the solution for $n-1$ points.

new point p



FACT: updated disc goes through p

New problem: min disc enclosing P and going through p

W(P, R) — find smallest disc enclosing points P with points R on the boundary.
$|R| \leq 3$. Initially P is the whole set of points and $R = \emptyset$

if $|R| = 3$ or $P = \emptyset$ — easy

$D := W(P - \{p\}, R)$ — p chosen at random

if $p \in D$ return D

else return $W(P - \{p\}, R \cup \{p\})$

Expected run time O(n) (no details)

Summary

- brief intro to linear programming
- linear programming in fixed dimension — randomized algorithm with expected run time O(*n*)
- smallest enclosing disc

References

- [CGAA] Chapter 4
- [Zurich] Appendix E, F, G
- Seidel's paper

Small-dimensional linear programming and convex hulls made easy
R Seidel - Discrete & Computational Geometry, 1991 - Springer
https://doi.org/10.1007/BF02574699

- general Linear Programming

Understanding and **using linear programming**
J Matousek, B Gärtner - 2007
https://ocul-wtl.primo.exlibrisgroup.com/permalink/01OCUL_WTL/5ob3ju/alma9953153109505162