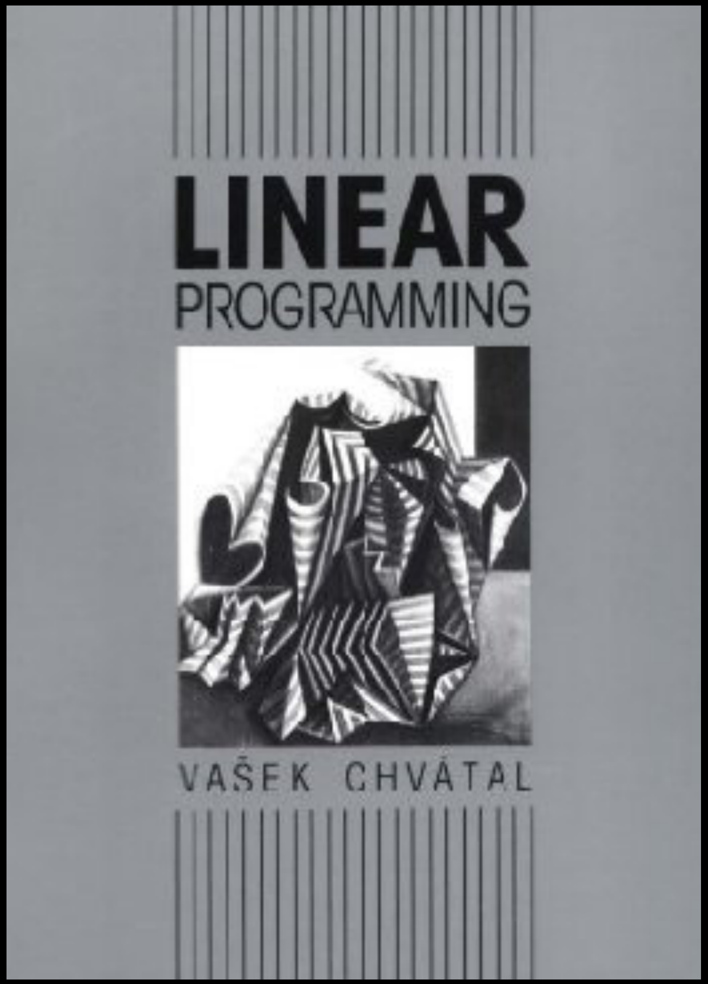


Lecture slides by Kevin Wayne

Linear Programming I

- *a refreshing example*
- *standard form*
- *fundamental questions*
- *geometry*
- *linear algebra*
- *simplex algorithm*

Last updated on 7/25/17 11:09 AM

Linear programming

Linear programming. Optimize a linear function subject to linear inequalities.

$$
\begin{array}{llrcll}
\text{(P)} & \max & \sum_{j=1}^{n} c_j x_j & & & \\
& \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j & = & b_i & 1 \le i \le m \\
& & x_j & \ge & 0 & 1 \le j \le n
\end{array}
$$

$$
\begin{array}{llrcl}
\text{(P)} & \max & c^T x & & \\
& \text{s.t.} & Ax & = & b \\
& & x & \ge & 0
\end{array}
$$

Linear programming

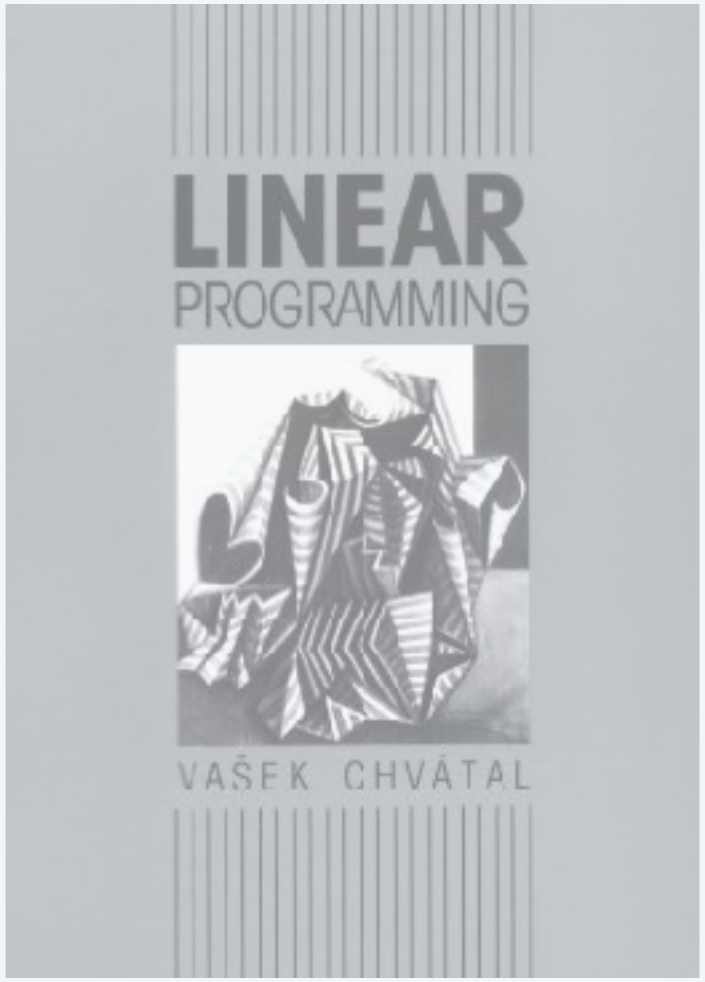
Linear programming. Optimize a linear function subject to linear inequalities.

Generalizes: $Ax = b$, 2-person zero-sum games, shortest path, max flow, assignment problem, matching, multicommodity flow, MST, min weighted arborescence, …

Why significant?

- Design poly-time algorithms.
- Design approximation algorithms.
- Solve NP-hard problems using branch-and-cut.

Ranked among most important scientific advances of 20th century.



Linear Programming I

- *a refreshing example*
- *standard form*
- *fundamental questions*
- *geometry*
- *linear algebra*
- *simplex algorithm*

Reference: *The Allocation of Resources by Linear Programming,* Scientific American, by Bob Bland

Brewery problem

Small brewery produces ale and beer.

- Production limited by scarce resources: corn, hops, barley malt.
- Recipes for ale and beer require different proportions of resources.

Beverage	Corn (pounds)	Hops (ounces)	Malt (pounds)	Profit ($)
Ale (barrel)	5	4	35	13
Beer (barrel)	15	4	20	23
constraint	480	160	1190	

How can brewer maximize profits?

- Devote all resources to ale: 34 barrels of ale ⇒ \$442
- Devote all resources to beer: 32 barrels of beer ⇒ \$736
- 7.5 barrels of ale, 29.5 barrels of beer ⇒ \$776
- 12 barrels of ale, 28 barrels of beer ⇒ \$800

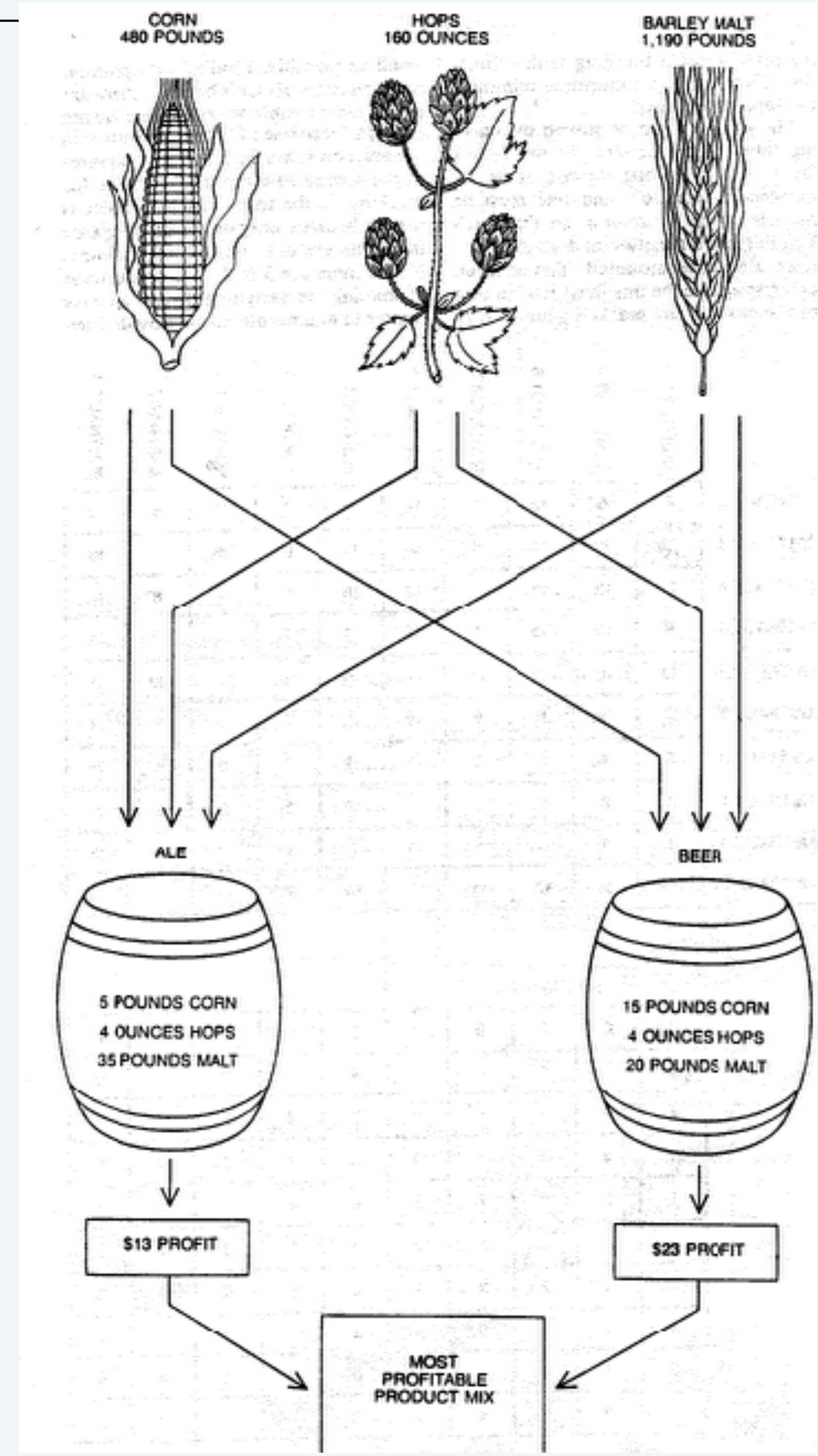
Brewery problem

objective function

	Ale		Beer			
max	$13A$	+	$23B$			Profit
s. t.	$5A$	+	$15B$	$\le$	480	Corn
	$4A$	+	$4B$	$\le$	160	Hops
	$35A$	+	$20B$	$\le$	1190	Malt
	A	,	B	$\ge$	0	

constraint

decision variable



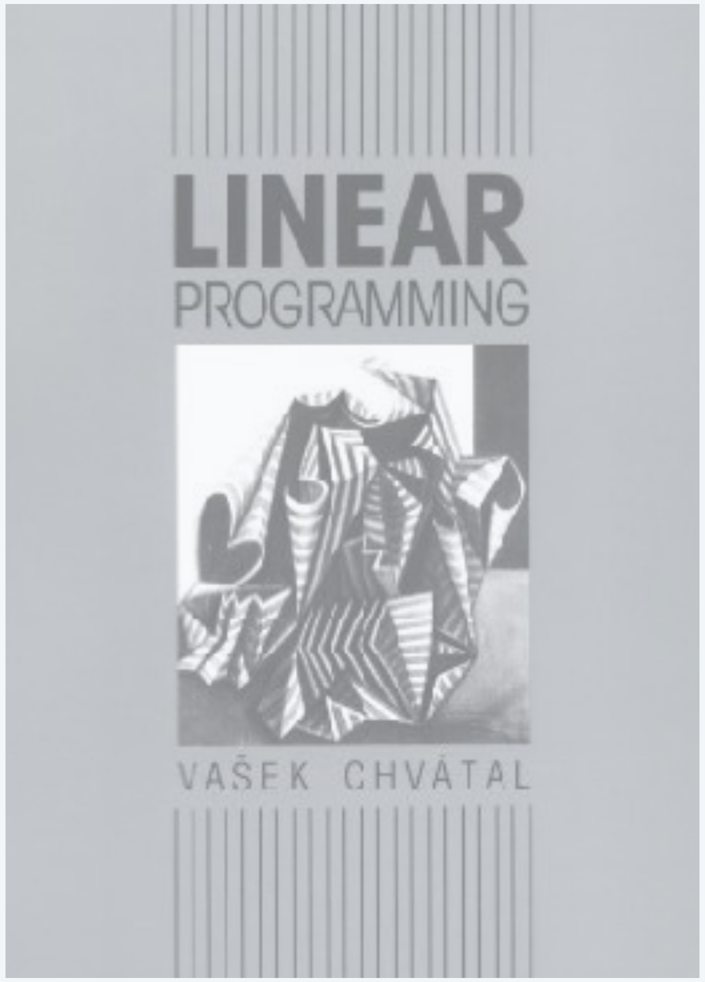
Reference

SCIENTIFIC AMERICAN JUNE 1981

The Allocation of Resources by Linear Programming

Abstract, crystal-like structures in many geometrical dimensions can help to solve problems in planning and management. A new algorithm has set upper limits on the complexity of such problems

By Robert G. Bland



LINEAR PROGRAMMING I

- *a refreshing example*
- *standard form*
- *fundamental questions*
- *geometry*
- *linear algebra*
- *simplex algorithm*

Standard form of a linear program

"Standard form" of an LP.

- Input: real numbers a_{ij}, c_j, b_i.
- Output: real numbers x_j.
- n = # decision variables, m = # constraints.
- Maximize linear objective function subject to linear inequalities.

$$
\begin{array}{llrcll}
\text{(P)} & \max & \sum_{j=1}^{n} c_j x_j & & & \\
 & \text{s. t.} & \sum_{j=1}^{n} a_{ij} x_j & = & b_i & 1 \le i \le m \\
 & & x_j & \ge & 0 & 1 \le j \le n
\end{array}
$$

$$
\begin{array}{llrcl}
\text{(P)} & \max & c^T x & & \\
 & \text{s. t.} & Ax & = & b \\
 & & x & \ge & 0
\end{array}
$$

Linear. No x^2, $x\,y$, $\arccos(x)$, etc.

Programming. Planning (term predates computer programming).

Brewery problem: converting to standard form

Original input.

$$
\begin{array}{rrcrcr}
\max & 13A & + & 23B & & \\
\text{s. t.} & 5A & + & 15B & \leq & 480 \\
 & 4A & + & 4B & \leq & 160 \\
 & 35A & + & 20B & \leq & 1190 \\
 & A & , & B & \geq & 0
\end{array}
$$

Standard form.

- Add slack variable for each inequality.
- Now a 5-dimensional problem.

$$
\begin{array}{rrcrcccccccr}
\max & 13A & + & 23B & & & & & & & & \\
\text{s. t.} & 5A & + & 15B & + & S_C & & & & & = & 480 \\
 & 4A & + & 4B & & & + & S_H & & & = & 160 \\
 & 35A & + & 20B & & & & & + & S_M & = & 1190 \\
 & A & , & B & , & S_C & , & S_H & , & S_M & \geq & 0
\end{array}
$$

Equivalent forms

Easy to convert variants to standard form.

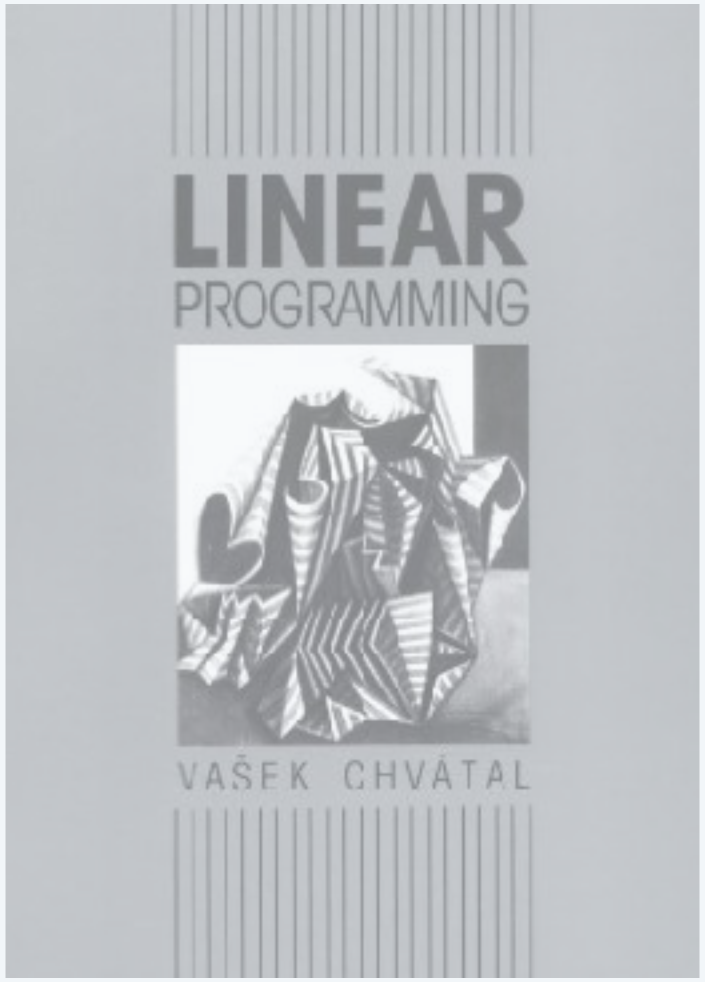
$$
\begin{array}{llrcl}
\text{(P)} & \max & c^T x & & \\
 & \text{s. t.} & Ax & = & b \\
 & & x & \geq & 0
\end{array}
$$

Less than to equality. $x + 2y - 3z \leq 17 \Rightarrow x + 2y - 3z + s = 17, s \geq 0$

Greater than to equality. $x + 2y - 3z \geq 17 \Rightarrow x + 2y - 3z - s = 17, s \geq 0$

Min to max. $\min \; x + 2y - 3z \Rightarrow \max \; -x - 2y + 3z$

Unrestricted to nonnegative. x unrestricted $\Rightarrow x = x^+ - x^-, \; x^+ \geq 0, x^- \geq 0$



LINEAR PROGRAMMING I

- *a refreshing example*
- *standard form*
- *fundamental questions*
- *geometry*
- *linear algebra*
- *simplex algorithm*

Fundamental questions

LP. For $A \in \Re^{m \times n}$, $b \in \Re^{m}$, $c \in \Re^{n}$, $\alpha \in \Re$, does there exist $x \in \Re^{n}$ such that: $Ax = b,\ x \geq 0,\ c^{\mathrm{T}} x \geq \alpha$?

Q. Is LP in **NP**?

Q. Is LP in co-**NP**?

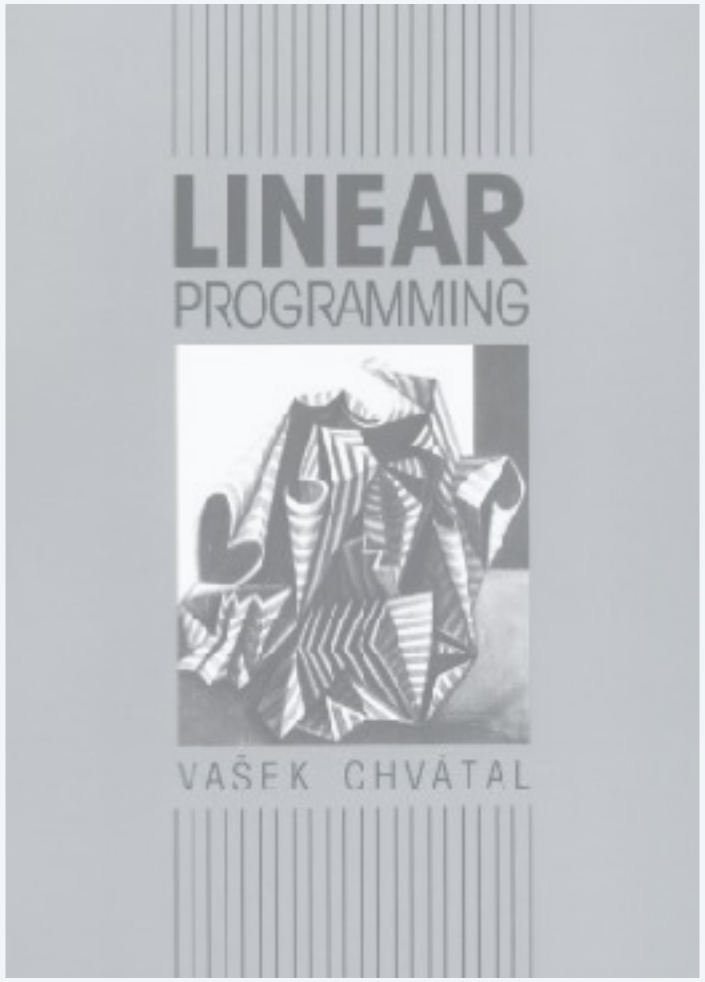
Q. Is LP in **P**?

Q. Is LP in $\mathbf{P}_{\Re}$?

Blum–Shub–Smale model

Input size.

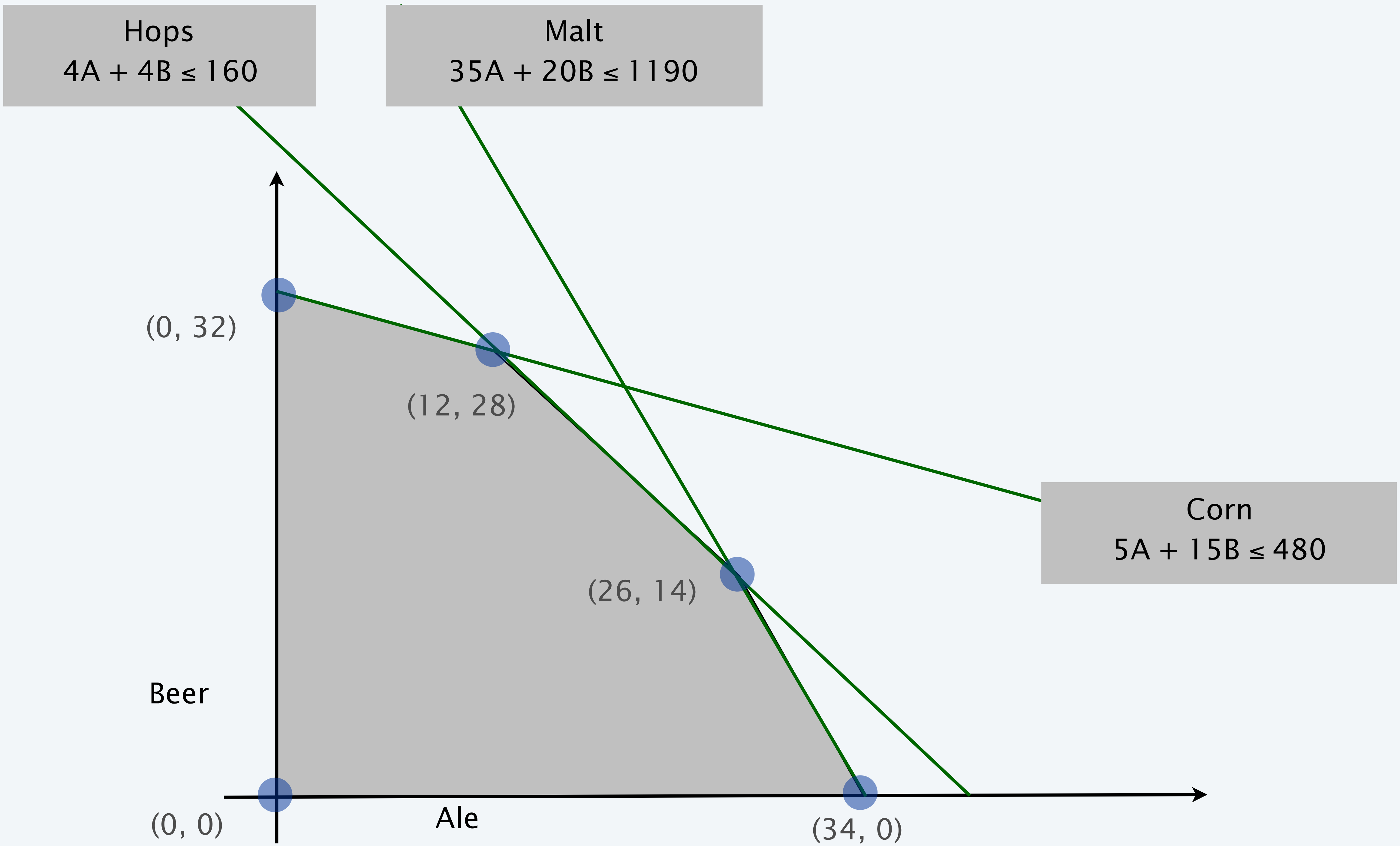
- n = number of variables.
- m = number of constraints.
- L = number of bits to encode input.



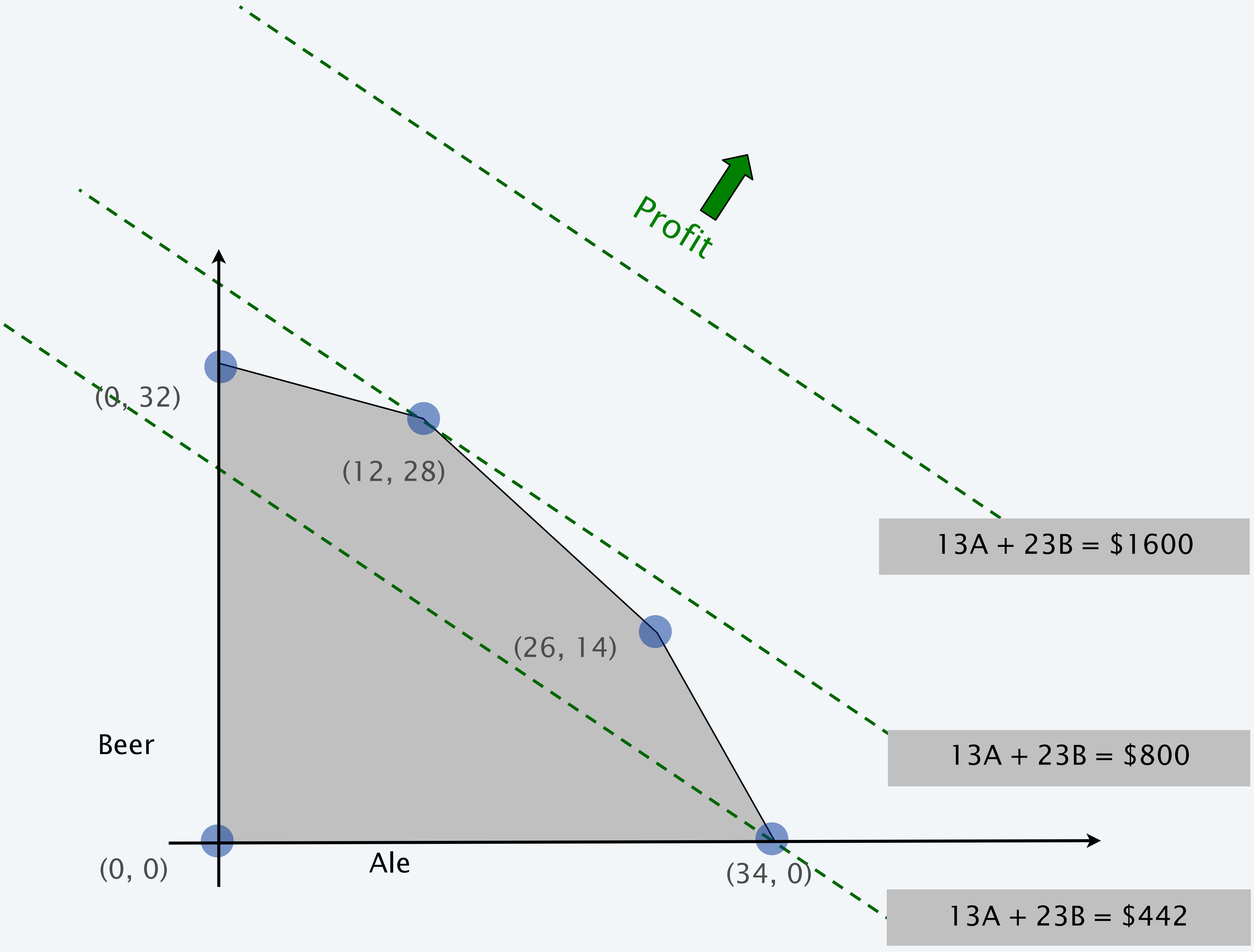
LINEAR PROGRAMMING I

- *a refreshing example*
- *standard form*
- *fundamental questions*
- *geometry*
- *linear algebra*
- *simplex algorithm*

Brewery problem: feasible region

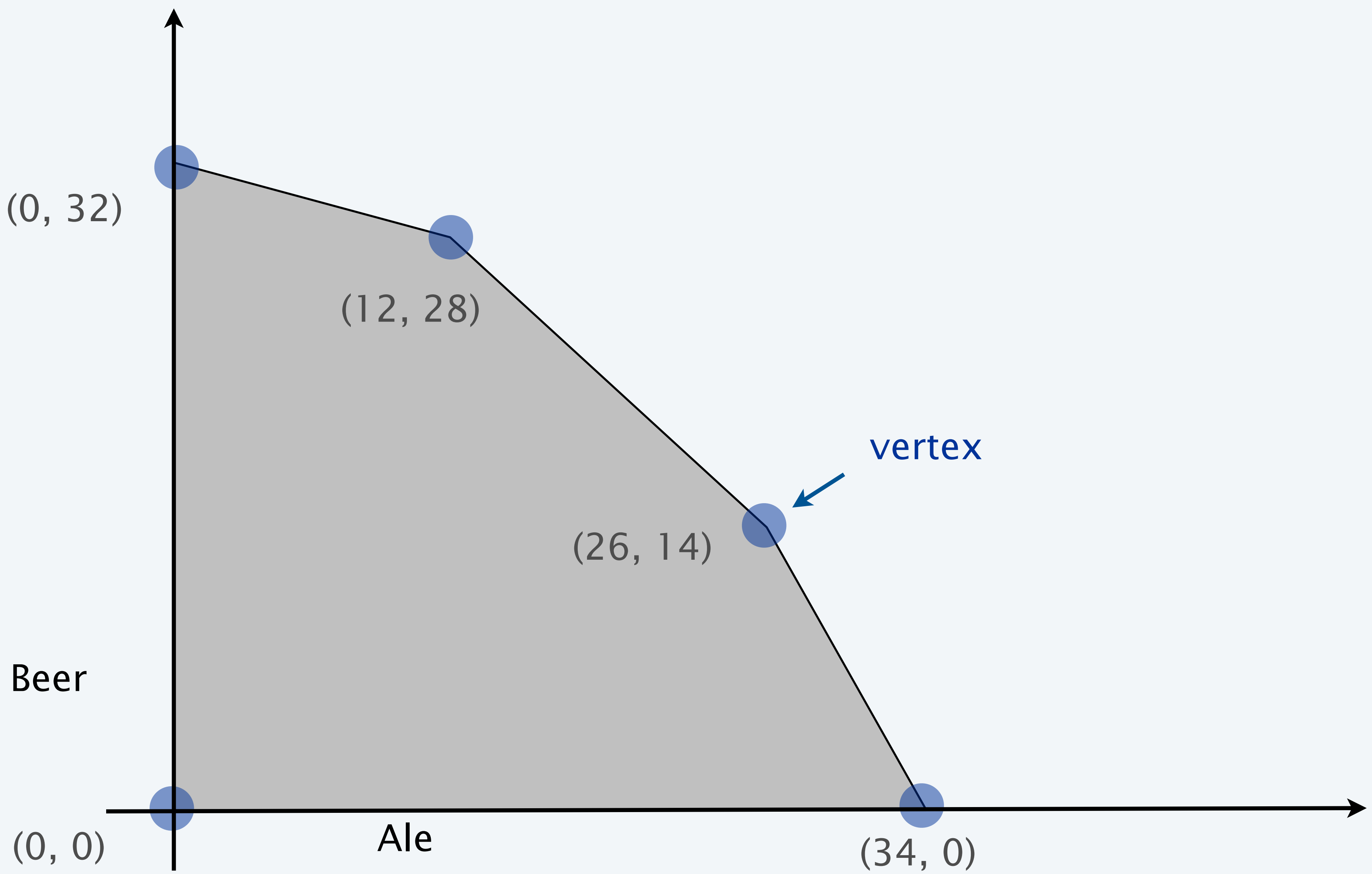


Brewery problem: objective function



Brewery problem: geometry

Brewery problem observation. Regardless of objective function coefficients, an optimal solution occurs at a vertex.



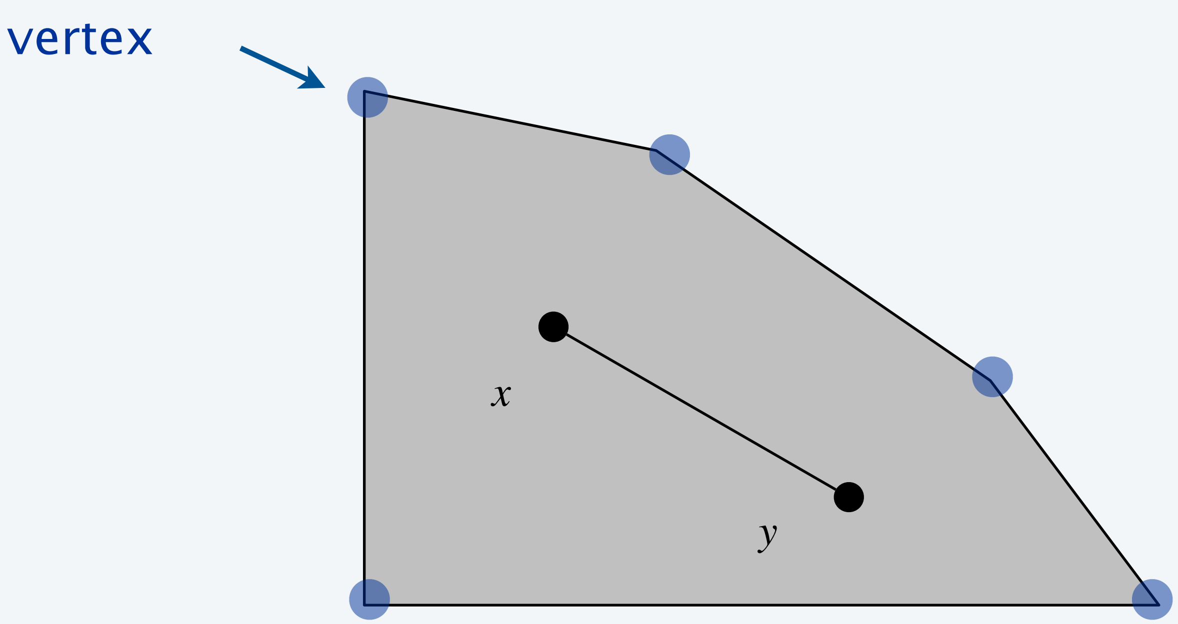
Convexity

Convex set. If two points x and y are in the set, then so is $\lambda x + (1 - \lambda) y$ for $0 \leq \lambda \leq 1$.

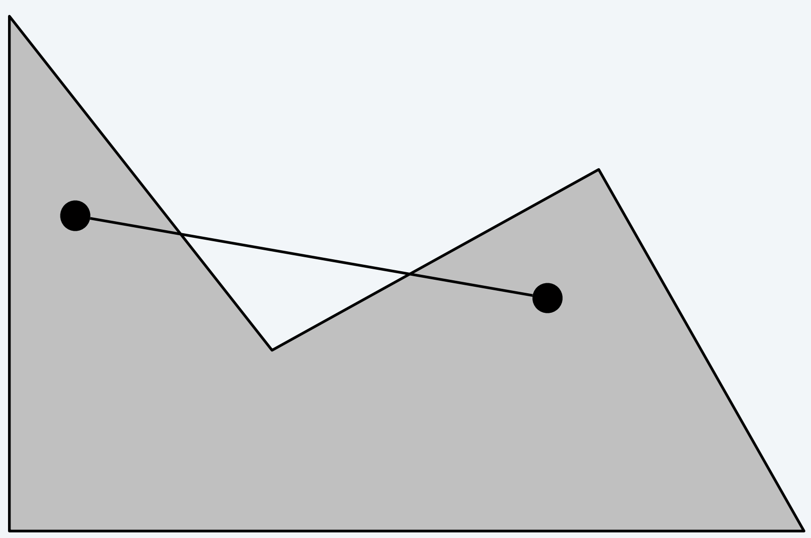
convex combination

not a vertex iff $\exists\, d \neq 0$ s.t. $x \pm d$ in set

Vertex. A point x in the set that can't be written as a strict convex combination of two distinct points in the set.



convex



not convex

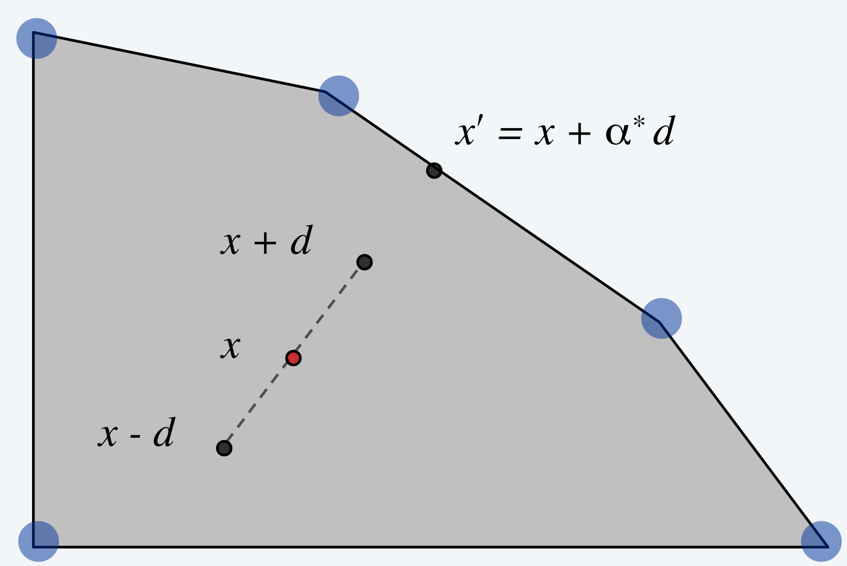
Observation. LP feasible region is a convex set.

Purificaiton

Theorem. If there exists an optimal solution to (P), then there exists one that is a vertex.

$$\begin{array}{llrcl} \text{(P)} & \max & c^T x & & \\ & \text{s. t.} & Ax & = & b \\ & & x & \geq & 0 \end{array}$$

Intuition. If x is not a vertex, move in a non-decreasing direction until you reach a boundary. Repeat.



Purificaiton

Theorem. If there exists an optimal solution to (P), then there exists one that is a vertex.

Pf.

- Suppose x is an optimal solution that is not a vertex.
- There exist direction $d \neq 0$ such that $x \pm d \in P$.
- $A\,d = 0$ because $A(x \pm d) = b$.
- Assume $c^{\mathrm{T}} d \leq 0$ (by taking either d or $-d$).
- Consider $x + \lambda\, d,\ \lambda > 0$:

Case 1. [there exists j such that $d_j < 0$]

- Increase λ to λ^* until first new component of $x + \lambda\, d$ hits 0.
- $x + \lambda^* d$ is feasible since $A(x + \lambda^* d) = Ax = b$ and $x + \lambda^* y \geq 0$.
- $x + \lambda^* d$ has one more zero component than x.
- $c^{\mathrm{T}} x' = c^{\mathrm{T}}(x + \lambda^* d) = c^{\mathrm{T}} x + \lambda^* c^{\mathrm{T}} d \leq c^{\mathrm{T}} x$.

$d_k = 0$ whenever $x_k = 0$ because $x \pm d \in P$

Purificaiton

Theorem. If there exists an optimal solution to (P), then there exists one that is a vertex.

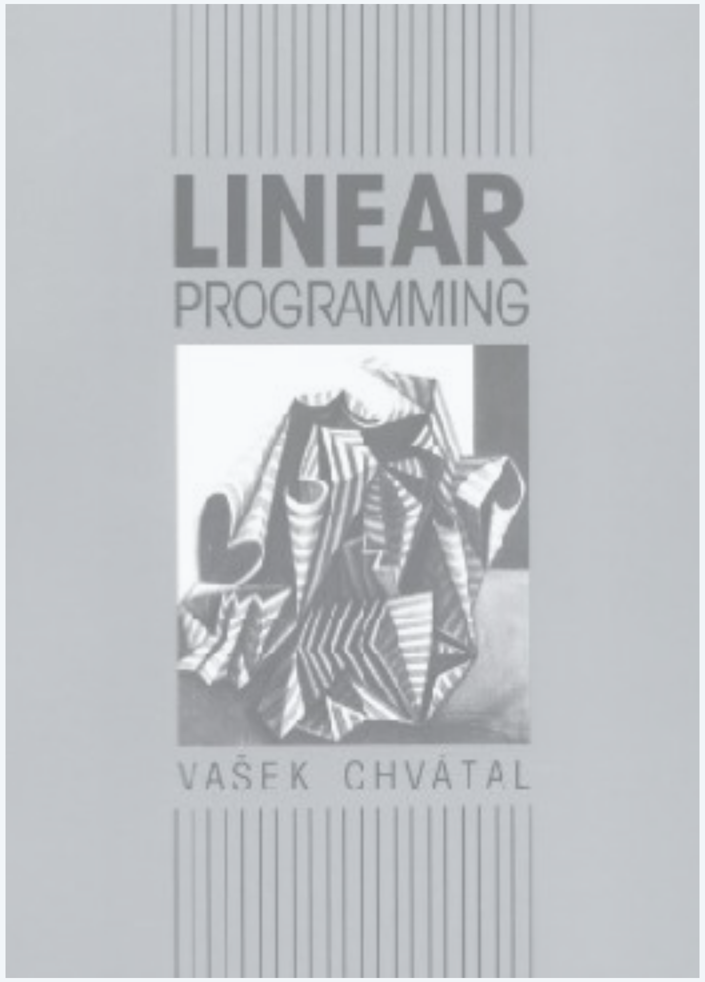
Pf.

- Suppose x is an optimal solution that is not a vertex.
- There exist direction $d \neq 0$ such that $x \pm d \in P$.
- $A\,d = 0$ because $A(x \pm d) = b$.
- Assume $c^T d \leq 0$ (by taking either d or $-d$).
- Consider $x + \lambda\, d,\ \lambda > 0$:

Case 2. [$d_j \geq 0$ for all j]

- $x + \lambda d$ is feasible for all $\lambda \geq 0$ since $A(x + \lambda d) = b$ and $x + \lambda d \geq x \geq 0$.
- As $\lambda \to \infty$, $c^T(x + \lambda d) \to \infty$ because $c^T d < 0$. ▪

if $c^T d = 0$, choose d so that case 1 applies

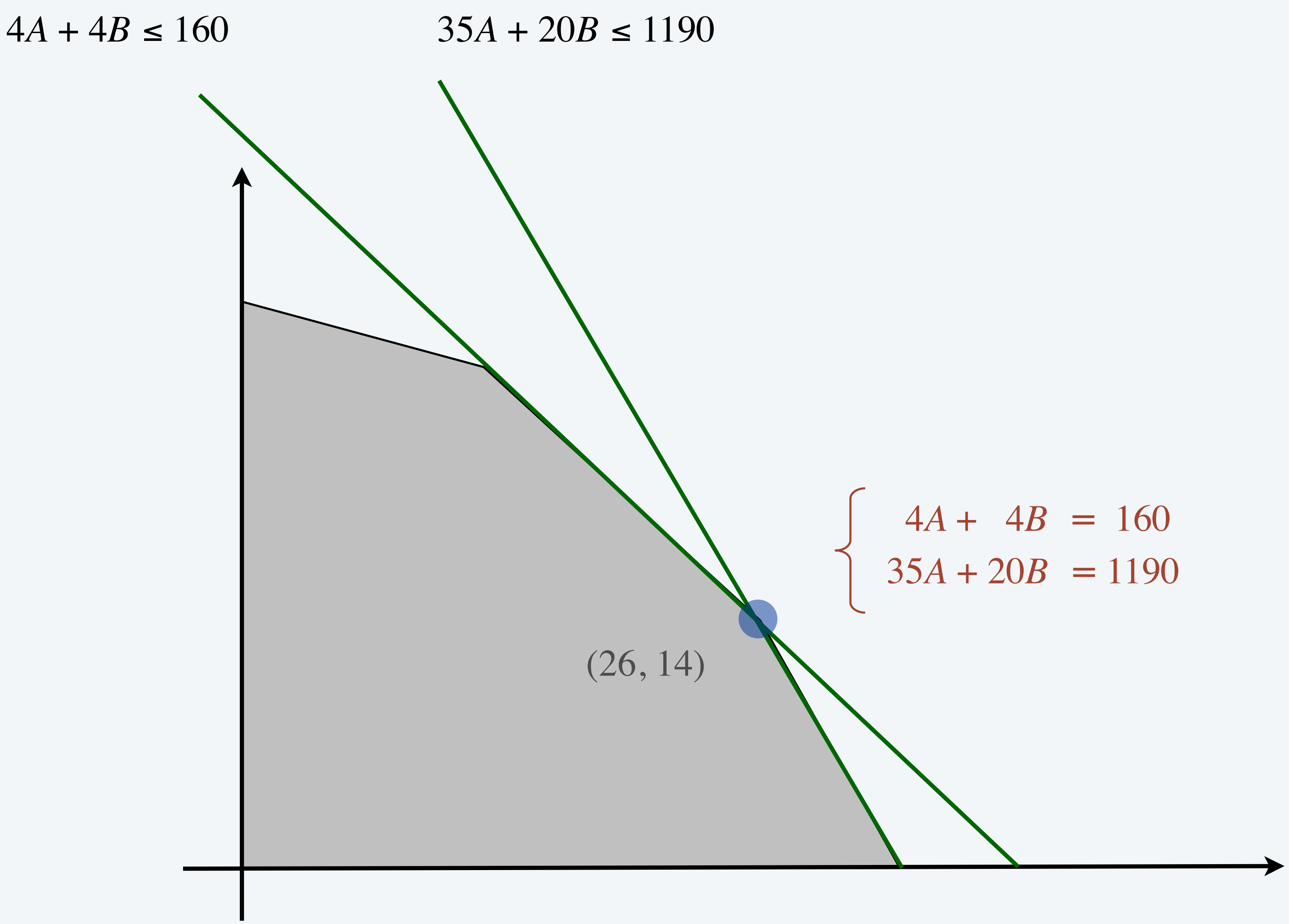


Linear Programming I

- *a refreshing example*
- *standard form*
- *fundamental questions*
- *geometry*
- *linear algebra*
- *simplex algorithm*

Intuition

Intuition. A vertex in $\Re^m$ is uniquely specified by m linearly independent equations.



Basic feasible solution

Theorem. Let $P = \{ x : Ax = b, x \geq 0 \}$. For $x \in P$, define $B = \{ j : x_j > 0 \}$. Then, x is a vertex iff A_B has linearly independent columns.

Notation. Let B = set of column indices. Define A_B to be the subset of columns of A indexed by B.

Ex.

$$A = \begin{bmatrix} 2 & 1 & 3 & 0 \\ 7 & 3 & 2 & 1 \\ 0 & 0 & 0 & 5 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 16 \\ 0 \end{bmatrix}$$

$$x = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad B = \{1, 3\}, \quad A_B = \begin{bmatrix} 2 & 3 \\ 7 & 2 \\ 0 & 0 \end{bmatrix}$$

Basic feasible solution

Theorem. Let $P = \{ x : Ax = b, x \geq 0 \}$. For $x \in P$, define $B = \{ j : x_j > 0 \}$. Then, x is a vertex iff A_B has linearly independent columns.

Pf. $\Leftarrow$

- Assume x is not a vertex.
- There exist direction $d \neq 0$ such that $x \pm d \in P$.
- $A\,d = 0$ because $A(x \pm d) = b$.
- Define $B' = \{ j : d_j \neq 0 \}$.
- $A_{B'}$ has linearly dependent columns since $d \neq 0$.
- Moreover, $d_j = 0$ whenever $x_j = 0$ because $x \pm d \geq 0$.
- Thus $B' \subseteq B$, so $A_{B'}$ is a submatrix of A_B.
- Therefore, A_B has linearly dependent columns.

Basic feasible solution

Theorem. Let $P = \{ x : Ax = b, x \geq 0 \}$. For $x \in P$, define $B = \{ j : x_j > 0 \}$. Then, x is a vertex iff A_B has linearly independent columns.

Pf. $\Rightarrow$

- Assume A_B has linearly dependent columns.
- There exist $d \neq 0$ such that $A_B\, d = 0$.
- Extend d to $\Re^n$ by adding 0 components.
- Now, $A\, d = 0$ and $d_j = 0$ whenever $x_j = 0$.
- For sufficiently small λ, $x \pm \lambda\, d \in P \Rightarrow x$ is not a vertex. ▪

Basic feasible solution

Theorem. Given $P = \{ x : Ax = b, x \geq 0 \}$, x is a vertex iff there exists $B \subseteq \{ 1, \ldots, n \}$ such $| B | = m$ and:

- A_B is nonsingular.
- $x_B = A_B^{-1} b \geq 0$.
- $x_N = 0$.

basic feasible solution

Pf. Augment A_B with linearly independent columns (if needed). ▪

$$A = \begin{bmatrix} 2 & 1 & 3 & 0 \\ 7 & 3 & 2 & 1 \\ 0 & 0 & 0 & 5 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 16 \\ 0 \end{bmatrix}$$

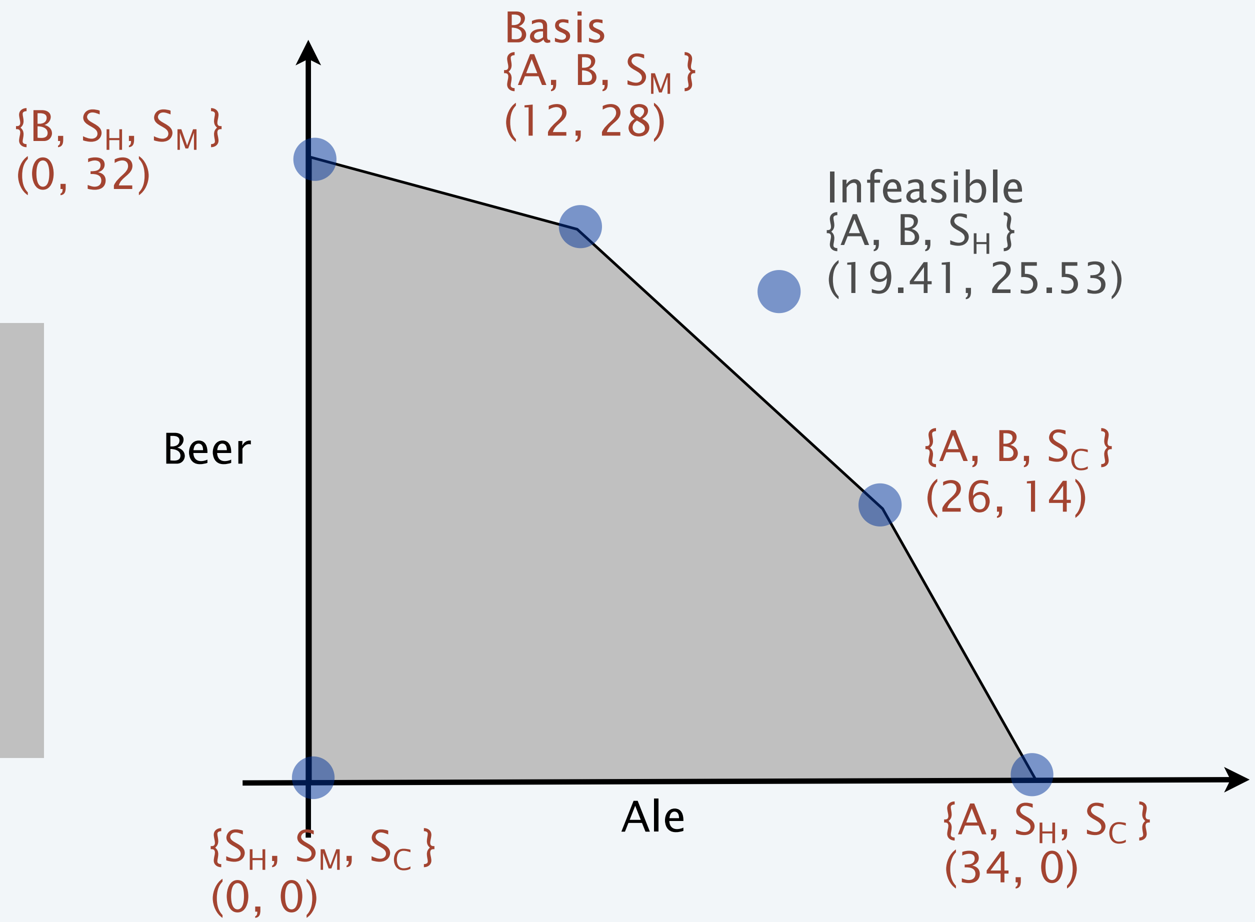
$$x = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad B = \{ 1, 3, 4 \}, \quad A_B = \begin{bmatrix} 2 & 3 & 0 \\ 7 & 2 & 1 \\ 0 & 0 & 5 \end{bmatrix}$$

Assumption. $A \in \Re^{m \times n}$ has full row rank.

Basic feasible solution: example

Basic feasible solutions.

$$
\begin{array}{lrcrcrcrcrcl}
\max & 13A & + & 23B & & & & & & & & \\
\text{s. t.} & 5A & + & 15B & + & S_C & & & & & = & 480 \\
 & 4A & + & 4B & & & + & S_H & & & = & 160 \\
 & 35A & + & 20B & & & & & + & S_M & = & 1190 \\
 & A & , & B & , & S_C & , & S_H & , & S_M & \geq & 0
\end{array}
$$



Fundamental questions

LP. For $A \in \Re^{m \times n}$, $b \in \Re^m$, $c \in \Re^n$, $\alpha \in \Re$, does there exist $x \in \Re^n$ such that: $Ax = b$, $x \geq 0$, $c^T x \geq \alpha$?

Q. Is LP in NP?

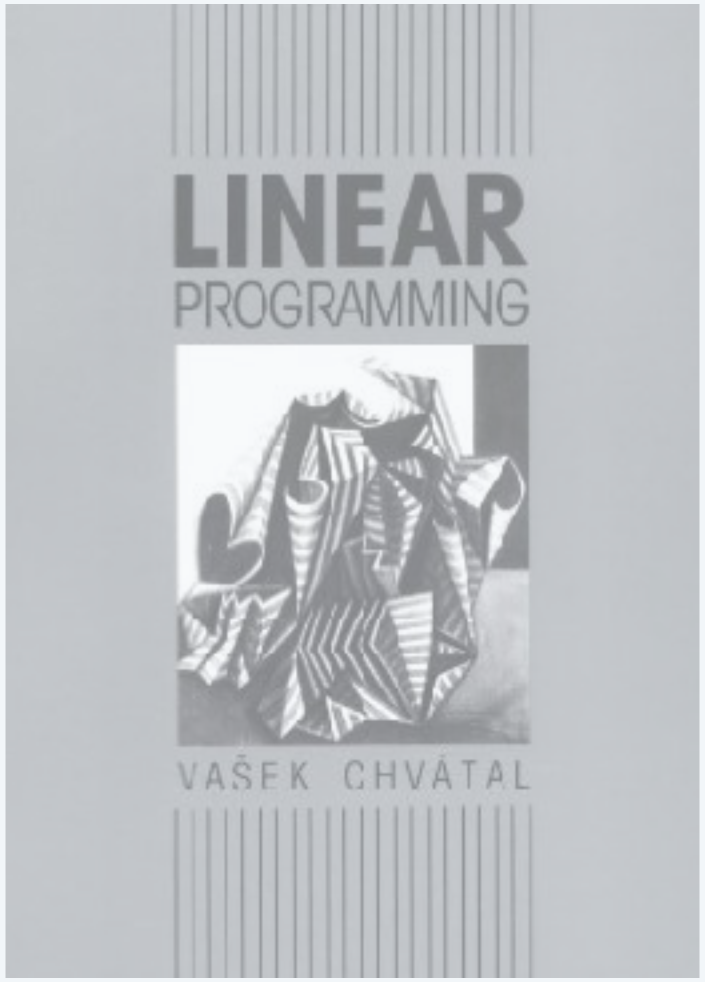
A. Yes.

- Number of vertices $\leq C(n, m) = \binom{n}{m} \leq n^m$.
- Cramer's rule $\Rightarrow$ can check a vertex in poly-time.

Cramer's rule. For $B \in \Re^{n \times n}$ invertible, $b \in \Re^n$, the solution to $Bx = b$ is given by:

$$x_i = \frac{\det(B_i)}{\det(B)}$$

replace ith column of B with b



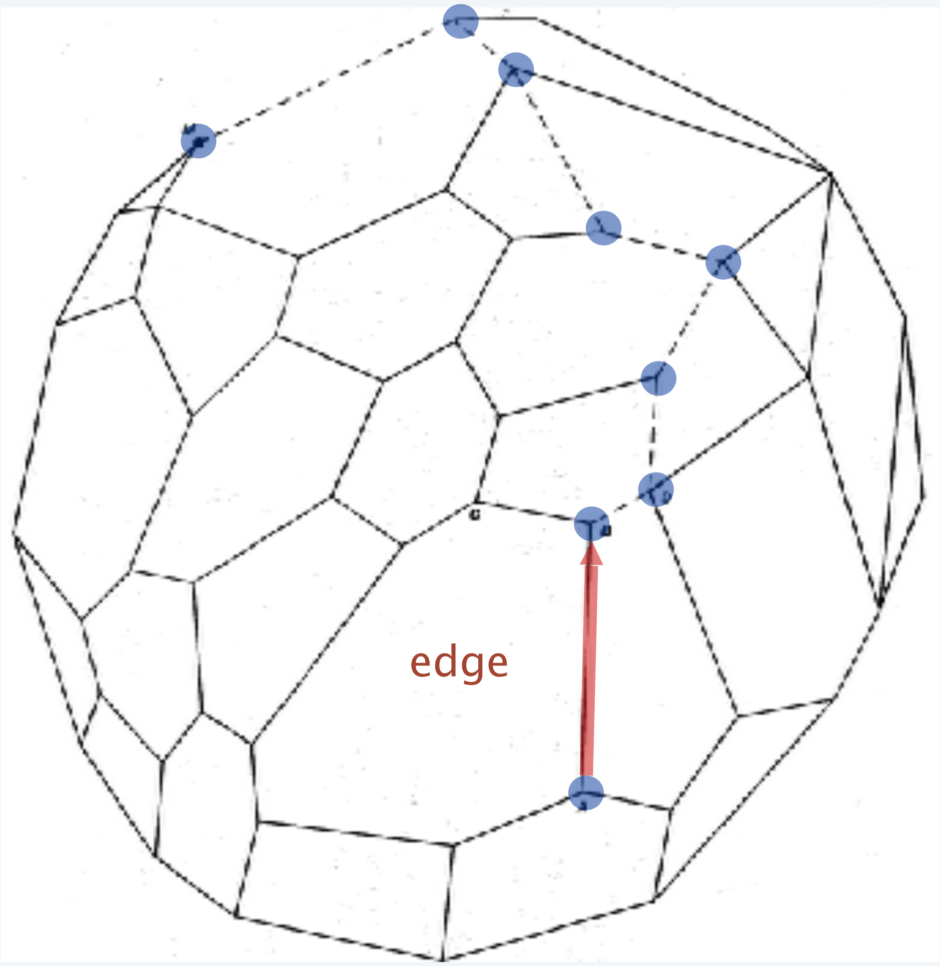
LINEAR PROGRAMMING I

- a refreshing example
- standard form
- fundamental questions
- geometry
- linear algebra
- simplex algorithm

Simplex algorithm: intuition

Simplex algorithm. [George Dantzig 1947] Move from BFS to adjacent BFS, without decreasing objective function.

replace one basic variable with another



Greedy property. BFS optimal iff no adjacent BFS is better.

Challenge. Number of BFS can be exponential!

Simplex algorithm: initialization

max Z subject to

$$
\begin{array}{rcrcccccccrcr}
13A & + & 23B & & & & & & & - & Z & = & 0 \\
\hline
5A & + & 15B & + & S_C & & & & & & & = & 480 \\
4A & + & 4B & & & + & S_H & & & & & = & 160 \\
35A & + & 20B & & & & & + & S_M & & & = & 1190 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

Basis = $\{S_C, S_H, S_M\}$
$A = B = 0$
$Z = 0$
$S_C = 480$
$S_H = 160$
$S_M = 1190$

Simplex algorithm: pivot 1

max Z subject to

$$
\begin{array}{rcrcrcrcrcrcl}
13A & + & 23B & & & & & & & - & Z & = & 0 \\
\hline
5A & + & 15B & + & S_C & & & & & & & = & 480 \\
4A & + & 4B & & & + & S_H & & & & & = & 160 \\
35A & + & 20B & & & & & + & S_M & & & = & 1190 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

Basis = $\{S_C, S_H, S_M\}$
$A = B = 0$
$Z = 0$
$S_C = 480$
$S_H = 160$
$S_M = 1190$

Substitute: $B = 1/15\ (480 - 5A - S_C)$

max Z subject to

$$
\begin{array}{rcrcrcrcrcrcl}
\frac{16}{3}A & & & - & \frac{23}{15}S_C & & & & & - & Z & = & -736 \\
\hline
\frac{1}{3}A & + & B & + & \frac{1}{15}S_C & & & & & & & = & 32 \\
\frac{8}{3}A & & & - & \frac{4}{15}S_C & + & S_H & & & & & = & 32 \\
\frac{85}{3}A & & & - & \frac{4}{3}S_C & & & + & S_M & & & = & 550 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

Basis = $\{B, S_H, S_M\}$
$A = S_C = 0$
$Z = 736$
$B = 32$
$S_H = 32$
$S_M = 550$

Simplex algorithm: pivot 1

max Z subject to

$$
\begin{array}{rcrcrcrcrcrcl}
13A & + & 23B & & & & & & & - & Z & = & 0 \\
\hline
5A & + & 15B & + & S_C & & & & & & & = & 480 \\
4A & + & 4B & & & + & S_H & & & & & = & 160 \\
35A & + & 20B & & & & & + & S_M & & & = & 1190 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

Basis = $\{S_C, S_H, S_M\}$
$A = B = 0$
$Z = 0$
$S_C = 480$
$S_H = 160$
$S_M = 1190$

Q. Why pivot on column 2 (or 1)?

A. Each unit increase in B increases objective value by \$23.

Q. Why pivot on row 2?

A. Preserves feasibility by ensuring RHS ≥ 0.

min ratio rule: $\min \{ 480/15, \ 160/4, \ 1190/20 \}$

Simplex algorithm: pivot 2

max Z subject to

$$
\begin{array}{rcrcrcrcrcrcr}
\frac{16}{3} A & & & - & \frac{23}{15} S_C & & & & & - & Z & = & -736 \\
\hline
\frac{1}{3} A & + & B & + & \frac{1}{15} S_C & & & & & & & = & 32 \\
\frac{8}{3} A & & & - & \frac{4}{15} S_C & + & S_H & & & & & = & 32 \\
\frac{85}{3} A & & & - & \frac{4}{3} S_C & & & + & S_M & & & = & 550 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

Basis = $\{B, S_H, S_M\}$
$A = S_C = 0$
$Z = 736$
$B = 32$
$S_H = 32$
$S_M = 550$

Substitute: $A = 3/8\ (32 + 4/15\ S_C - S_H)$

max Z subject to

$$
\begin{array}{rcrcrcrcrcrcr}
 & & & - & S_C & - & 2\ S_H & & & - & Z & = & -800 \\
\hline
 & & B & + & \frac{1}{10} S_C & + & \frac{1}{8} S_H & & & & & = & 28 \\
A & & & - & \frac{1}{10} S_C & + & \frac{3}{8} S_H & & & & & = & 12 \\
 & & & - & \frac{25}{6} S_C & - & \frac{85}{8} S_H & + & S_M & & & = & 110 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

Basis = $\{A, B, S_M\}$
$S_C = S_H = 0$
$Z = 800$
$B = 28$
$A = 12$
$S_M = 110$

Simplex algorithm: optimality

Q. When to stop pivoting?

A. When all coefficients in top row are nonpositive.

Q. Why is resulting solution optimal?

A. Any feasible solution satisfies system of equations in tableaux.

- In particular: $Z = 800 - S_C - 2\,S_H\ ,\ S_C \geq 0,\ S_H \geq 0.$
- Thus, optimal objective value $Z^* \leq 800$.
- Current BFS has value $800 \Rightarrow$ optimal.

max Z subject to

$$
\begin{array}{rrrrrrrrrrrrl}
 & & - & S_C & - & 2\,S_H & & & - & Z & = & -800 \\
\hline
 & B & + & \frac{1}{10}\,S_C & + & \frac{1}{8}\,S_H & & & & & = & 28 \\
A & & - & \frac{1}{10}\,S_C & + & \frac{3}{8}\,S_H & & & & & = & 12 \\
 & & - & \frac{25}{6}\,S_C & - & \frac{85}{8}\,S_H & + & S_M & & & = & 110 \\
A\ , & B\ , & & S_C\ , & & S_H\ , & & S_M & & & \geq & 0
\end{array}
$$

Basis = $\{A, B, S_M\}$
$S_C = S_H = 0$
$Z = 800$
$B = 28$
$A = 12$
$S_M = 110$

Simplex tableaux: matrix form

Initial simplex tableaux.

$$\begin{aligned} c_B^T x_B + c_N^T x_N &= Z \\ A_B x_B + A_N x_N &= b \\ x_B \;,\; x_N &\geq 0 \end{aligned}$$

Simplex tableaux corresponding to basis B.

$$\begin{aligned} (c_N^T - c_B^T A_B^{-1} A_N)\, x_N &= Z - c_B^T A_B^{-1} b \\ I x_B + A_B^{-1} A_N x_N &= A_B^{-1} b \\ x_B \;,\; x_N &\geq 0 \end{aligned}$$

← subtract $c_B^T A_B^{-1}$ times constraints

← multiply by A_B^{-1}

basic feasible solution	optimal basis
$x_B = A_B^{-1} b \geq 0$, $x_N = 0$	$c_N^T - c_B^T A_B^{-1} A_N \leq 0$

Simplex algorithm: corner cases

Simplex algorithm. Missing details for corner cases.

Q. What if min ratio test fails?

Q. How to find initial basis?

Q. How to guarantee termination?

Unboundedness

Q. What happens if min ratio test fails?

all coefficients in entering column are nonpositive

max Z subject to

$$
\begin{array}{rrrrrrrrrrl}
 & & & + & 2x_4 & + & 20x_5 & - & Z & = & 2 \\
\hline
x_1 & & & - & 4x_4 & - & 8x_5 & & & = & 3 \\
 & x_2 & & + & 5x_4 & - & 12x_5 & & & = & 4 \\
 & & x_3 & & & & & & & = & 5 \\
x_1, & x_2, & x_3, & & x_4, & & x_5 & & & \ge & 0
\end{array}
$$

A. Unbounded objective function.

$$
Z = 2 + 20x_5 \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 3 + 8x_5 \\ 4 + 12x_5 \\ 5 \\ 0 \\ 0 \end{bmatrix}
$$

Phase I simplex method

Q. How to find initial basis?

$$
\begin{array}{llrcl}
\text{(P)} & \max & c^T x & & \\
& \text{s.t.} & Ax & = & b \\
& & x & \geq & 0
\end{array}
$$

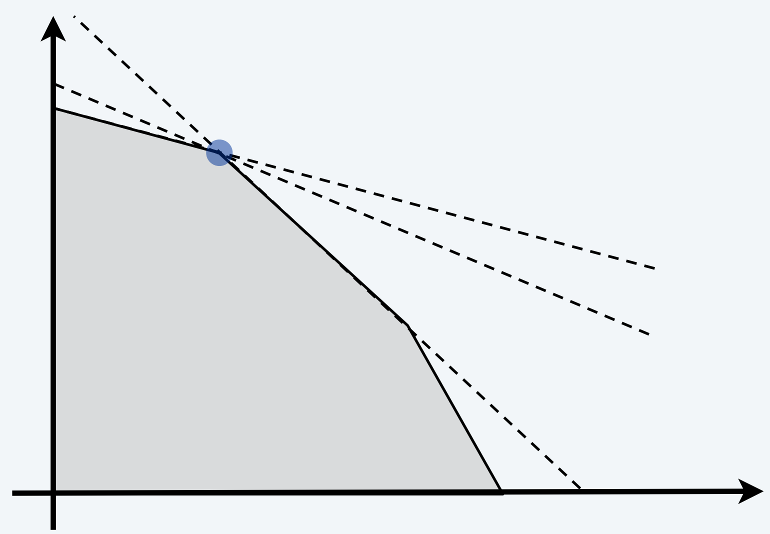
A. Solve (P′), starting from basis consisting of all the z_i variables.

$$
\begin{array}{llrcl}
\text{(P}'\text{)} & \max & \sum_{i=1}^{m} z_i & & \\
& \text{s.t.} & Ax + Iz & = & b \\
& & x, \quad z & \geq & 0
\end{array}
$$

- Case 1: min > 0 $\Rightarrow$ (P) is infeasible.
- Case 2: min $= 0$, basis has no z_i variables $\Rightarrow$ okay to start Phase II.
- Case 3a: min $= 0$, basis has z_i variables. Pivot z_i variables out of basis. If successful, start Phase II; else remove linear dependent rows.

Simplex algorithm: degeneracy

Degeneracy. New basis, same vertex.



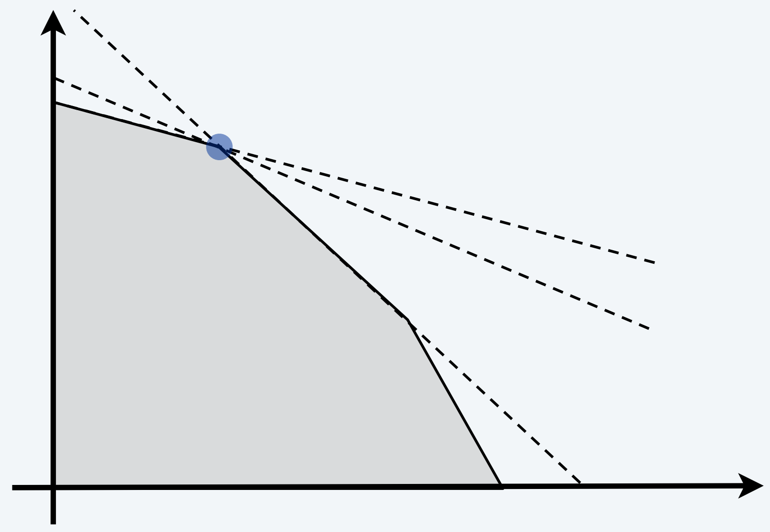
Degenerate pivot. Min ratio = 0.

max Z subject to

$$
\begin{array}{lllllllllllllll}
 & & & & \frac{3}{4}x_4 & - & 20x_5 & + & \frac{1}{2}x_6 & - & 6x_7 & - & Z & = & 0 \\
\hline
x_1 & & & + & \frac{1}{4}x_4 & - & 8x_5 & - & x_6 & + & 9x_7 & & & = & 0 \\
 & x_2 & & + & \frac{1}{2}x_4 & - & 12x_5 & - & \frac{1}{2}x_6 & + & 3x_7 & & & = & 0 \\
 & & x_3 & & & & & + & x_6 & & & & & = & 1 \\
x_1, & x_2, & x_3, & & x_4, & & x_5, & & x_6, & & x_7 & & & \geq & 0
\end{array}
$$

Simplex algorithm: degeneracy

Degeneracy. New basis, same vertex.



Cycling. Infinite loop by cycling through different bases that all correspond to same vertex.

Anti-cycling rules.

- Bland's rule: choose eligible variable with smallest index.
- Random rule: choose eligible variable uniformly at random.
- Lexicographic rule: perturb constraints so nondegenerate.

Lexicographic rule

Intuition. No degeneracy $\Rightarrow$ no cycling.

Perturbed problem.

much much greater, say $\varepsilon_i = \delta^i$ for small δ

$$(P') \quad \max \ c^T x \quad \text{s.t.} \ Ax = b + \varepsilon, \ x \geq 0 \qquad \text{where } \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \text{ such that } \varepsilon_1 \succ \varepsilon_2 \succ \cdots \succ \varepsilon_n$$

Lexicographic rule. Apply perturbation virtually by manipulating ε symbolically:

$$17 + 5\varepsilon_1 + 11\varepsilon_2 + 8\varepsilon_3 \leq 17 + 5\varepsilon_1 + 14\varepsilon_2 + 3\varepsilon_3$$

Lexicographic rule

Intuition. No degeneracy $\Rightarrow$ no cycling.

Perturbed problem.

much much greater, say $\varepsilon_i = \delta^i$ for small δ

$$(P') \quad \max \; c^T x \quad \text{s.t.} \; Ax = b + \varepsilon, \; x \ge 0 \qquad \text{where } \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \text{ such that } \varepsilon_1 \succ \varepsilon_2 \succ \cdots \succ \varepsilon_n$$

Claim. In perturbed problem, $x_B = A_B^{-1}(b + \varepsilon)$ is always nonzero.

Pf. The j^{th} component of x_B is a (nonzero) linear combination of the components of $b + \varepsilon$ $\Rightarrow$ contains at least one of the ε_i terms.

which can't cancel

Corollary. No cycling.

Simplex algorithm: practice

Remarkable property. In practice, simplex algorithm typically terminates after at most $2(m + n)$ pivots.

but no polynomial pivot rule known

Issues.

- Avoid stalling.
- Choose the pivot.
- Maintain sparsity.
- Ensure numerical stability.
- Preprocess to eliminate variables and constraints.

Commercial solvers can solve LPs with millions of variables and tens of thousands of constraints.