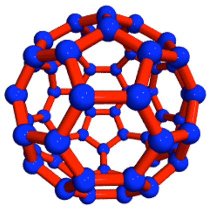
Linear Programming



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

Linear Programming

see ORF 307

What is it?
- Quintessential tool for optimal allocation of scarce resources, among a number of competing activities.
- Powerful and general problem-solving method that encompasses:
  - shortest path, network flow, MST, matching
  - Ax = b, 2-person zero sum games

Why significant?
- Widely applicable.
- Fast commercial solvers: CPLEX, OSL.
- Powerful modeling languages: AMPL, GAMS.
- Ranked among most important scientific advances of 20th century.
- Dominates world of industry.

Ex: Delta claims saving $100 million per year using LP

Applications

Agriculture. Diet problem.
Computer science. Compiler register allocation, data mining.
Electrical engineering. VLSI design, optimal clocking.
Energy. Blending petroleum products.
Economics. Equilibrium theory, two-person zero-sum games.
Environment. Water quality management.
Finance. Portfolio optimization.
Logistics. Supply-chain management.
Management. Hotel yield management.
Marketing. Direct mail advertising.
Manufacturing. Production line balancing, cutting stock.
Medicine. Radioactive seed placement in cancer treatment.
Operations research. Airline crew assignment, vehicle routing.
Physics. Ground states of 3-D Ising spin glasses.
Plasma physics. Optimal stellarator design.
Telecommunication. Network design, Internet routing.
Sports. Scheduling ACC basketball, handicapping horse races.

Brewery Problem: A Toy LP Example

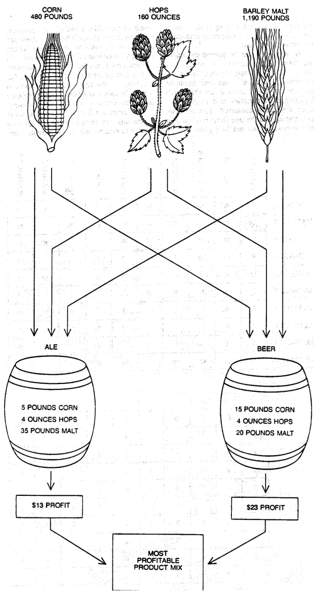
Small brewery produces ale and beer.
- Production limited by scarce resources: corn, hops, barley malt.
- Recipes for ale and beer require different proportions of resources.

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

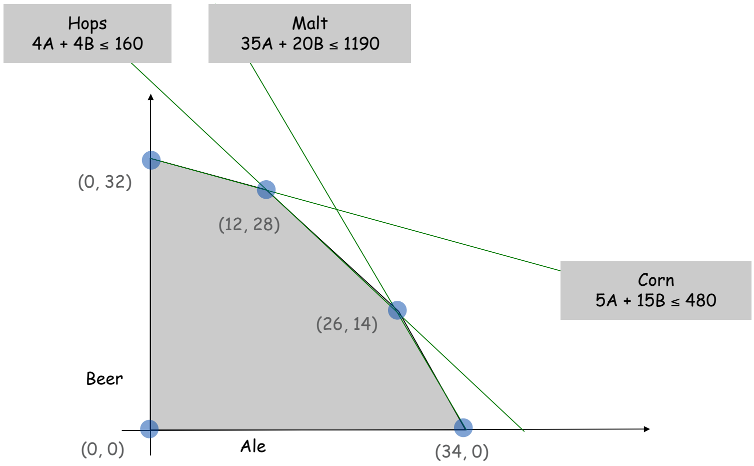
How can brewer maximize profits?
- Devote all resources to ale: 34 barrels of ale ⇒ $442.
- Devote all resources to beer: 32 barrels of beer ⇒ $736.
- 7.5 barrels of ale, 29.5 barrels of beer ⇒ $776.
- 12 barrels of ale, 28 barrels of beer ⇒ $800.

Brewery Problem

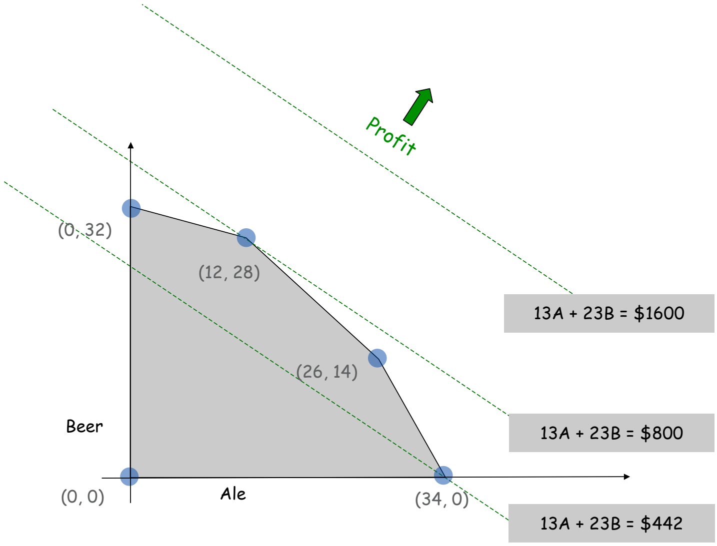
$$\begin{array}{lrcrccrl}
 & \overset{\text{Ale}}{\downarrow} & & \overset{\text{Beer}}{\downarrow} & & & & \\
\max & 13A & + & 23B & & & & \leftarrow \text{Profit} \\
\text{s. t.} & 5A & + & 15B & \le & 480 & & \leftarrow \text{Corn} \\
 & 4A & + & 4B & \le & 160 & & \leftarrow \text{Hops} \\
 & 35A & + & 20B & \le & 1190 & & \leftarrow \text{Malt} \\
 & A & , & B & \ge & 0 & &
\end{array}$$



Brewery Problem: Feasible Region

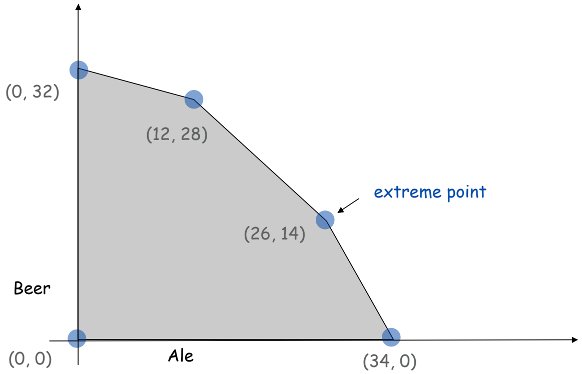


Brewery Problem: Objective Function



Brewery Problem: Geometry

Brewery problem observation. Regardless of objective function coefficients, an optimal solution occurs at an extreme point.



Standard Form LP

"Standard form" LP.

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

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

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

Linear. No x^2, xy, arccos(x), etc.

Programming. Planning (term predates computer programming).

Brewery Problem: Converting to Standard Form

Original input.

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

Standard form.

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

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

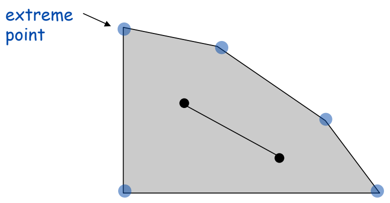
Geometry

Geometry.

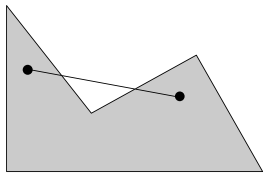
- Inequality: halfplane (2D), hyperplane (kD).
- Bounded feasible region: convex polygon (2D), convex polytope (kD).

Convex set. If two points a and b are in the set, then so is $\frac{1}{2}(a + b)$.

Extreme point. A point in the set that can't be written as $\frac{1}{2}(a + b)$, where a and b are two distinct points in the set.



convex



not convex

Geometry

Extreme point property. If there exists an optimal solution to (P), then there exists one that is an extreme point.

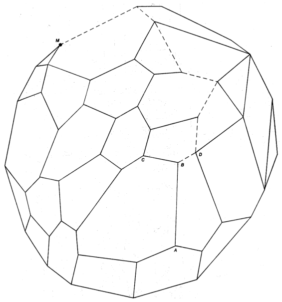
Consequence. Only need to consider finitely many possible solutions.

Challenge. Number of extreme points can be exponential!

n-dimensional hypercube

Greedy property. Extreme point is optimal iff no neighboring extreme point is better.

local optima are global optima



Simplex Algorithm

Simplex algorithm. [George Dantzig, 1947]

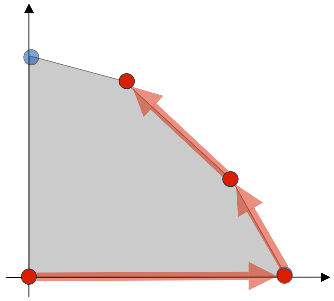
- Developed shortly after WWII in response to logistical problems, including Berlin airlift.
- One of greatest and most successful algorithms of all time.

Generic algorithm.

- Start at some extreme point.
- Pivot from one extreme point to a neighboring one.
- Repeat until optimal.

never decreasing objective function

How to implement? Linear algebra.



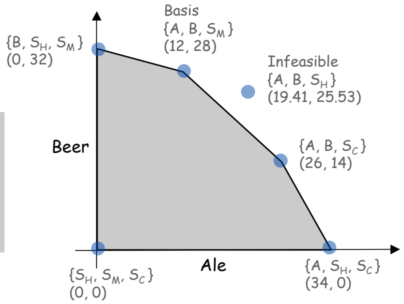
Simplex Algorithm: Basis

Basis. Subset of m of the n variables.

Basic feasible solution (BFS). Set n - m nonbasic variables to 0, solve for remaining m variables.

- Solve m equations in m unknowns.
- If unique and feasible solution ⇒ BFS.
- BFS ⇔ extreme point.

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



Simplex Algorithm: Initialization

$$
\begin{array}{rcrcrcrcrcrcr}
\multicolumn{13}{l}{\max Z \text{ subject to}} \\
13A & + & 23B & & & & & & & - & Z & = & 0 \\
\hline
5A & + & 15B & + & S_C & & & & & & & = & 480 \\
4A & + & 4B & & & + & S_H & & & & & = & 160 \\
35A & + & 20B & & & & & + & S_M & & & = & 1190 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

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

$$
\begin{array}{rcrcrcrcrcrcr}
\multicolumn{13}{l}{\max Z \text{ subject to}} \\
13A & + & 23B & & & & & & & - & Z & = & 0 \\
\hline
5A & + & 15B & + & S_C & & & & & & & = & 480 \\
4A & + & 4B & & & + & S_H & & & & & = & 160 \\
35A & + & 20B & & & & & + & S_M & & & = & 1190 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

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)

$$
\begin{array}{rcrcrcrcrcrcr}
\multicolumn{13}{l}{\max Z \text{ subject to}} \\
\frac{16}{3}A & & & - & \frac{23}{15}S_C & & & & & - & Z & = & -736 \\
\hline
\frac{1}{3}A & + & B & + & \frac{1}{15}S_C & & & & & & & = & 32 \\
\frac{8}{3}A & & & - & \frac{4}{15}S_C & + & S_H & & & & & = & 32 \\
\frac{85}{3}A & & & - & \frac{4}{3}S_C & & & + & S_M & & & = & 550 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
$$

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

Simplex Algorithm: Pivot 1

$$
\begin{array}{l}
\text{max } Z \text{ subject to} \\
\begin{array}{rcrcrcrcrcrcr}
13A & + & 23B & & & & & & & - & Z & = & 0 \\ \hline
5A & + & \boxed{15B} & + & S_C & & & & & & & = & 480 \\
4A & + & 4B & & & + & S_H & & & & & = & 160 \\
35A & + & 20B & & & & & + & S_M & & & = & 1190 \\
A & , & B & , & S_C & , & S_H & , & S_M & & & \geq & 0
\end{array}
\end{array}
$$

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

Why pivot on column 2?

- Each unit increase in B increases objective value by $23.
- Pivoting on column 1 also OK.

Why pivot on row 2?

- Preserves feasibility by ensuring RHS ≥ 0.
- Minimum ratio rule: min { 480/15, 160/4, 1190/20 }.

Simplex Algorithm: Pivot 2

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

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

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

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

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 non-positive.

Q. Why is resulting solution optimal?
A. Any feasible solution satisfies system of equations in tableaux.

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

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

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

Simplex Algorithm: Bare Bones Implementation

Construct the simplex tableaux.

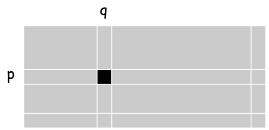
$$
\begin{array}{c|c|c|c}
m & A & I & b \\ \hline
1 & c & 0 & 0 \\
 & n & n & 1
\end{array}
$$

```
public class Simplex {
   private double[][] a;   // simplex tableaux
   private int M, N;

   public Simplex(double[][] A, double[] b, double[] c) {
      M = b.length;
      N = c.length;
      a = new double[M+1][M+N+1];
      for (int i = 0; i < M; i++)
         for (int j = 0; j < N; j++)
            a[i][j] = A[i][j];
      for (int j = N; j < M + N; j++) a[j-N][j] = 1.0;
      for (int j = 0; j < N;     j++) a[M][j]   = c[j];
      for (int i = 0; i < M;     i++) a[i][M+N] = b[i];
   }
```

Simplex Algorithm: Bare Bones Implementation

Pivot on element (p, q).



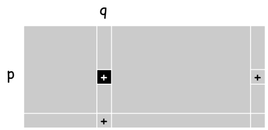
```
public void pivot(int p, int q) {
   for (int i = 0; i <= M; i++)
      for (int j = 0; j <= M + N; j++)
         if (i != p && j != q)
            a[i][j] -= a[p][j] * a[i][q] / a[p][q];

   for (int i = 0; i <= M; i++)
      if (i != p) a[i][q] = 0.0;               zero out column q

   for (int j = 0; j <= M + N; j++)
      if (j != q) a[p][j] /= a[p][q];          scale row p
   a[p][q] = 1.0;
}
```

Simplex Algorithm: Bare Bones Implementation

Simplex algorithm.



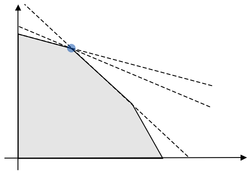
```
public void solve() {
   while (true) {
      int p, q;
      for (q = 0; q < M + N; q++)          find entering variable q
         if (a[M][q] > 0) break;           (positive objective function coefficient)
      if (q >= M + N) break;

      for (p = 0; p < M; p++)              find row p according to min ratio rule
         if (a[p][q] > 0) break;
      for (int i = p+1; i < M; i++)
         if (a[i][q] > 0)
            if (a[i][M+N] / a[i][q] < a[p][M+N] / a[p][q])
               p = i;                      min ratio test
      pivot(p, q);
   }
}
```

Simplex Algorithm: Degeneracy

Degeneracy. New basis, same extreme point.

"stalling" is common in practice



Cycling. Get stuck by cycling through different bases that all correspond to same extreme point.

- Doesn't occur in the wild.
- Bland's least index rule guarantees finite # of pivots.

Simplex Algorithm: Running Time

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

- No polynomial pivot rule known.
- Most pivot rules known to be exponential (or worse) in worst-case.

Pivoting rules. Carefully balance the cost of finding an entering variable with the number of pivots needed.

Simplex Algorithm: Implementation Issues

Implementation issues.

- Avoid stalling.
- Choosing the pivot.
- Numerical stability.
- Maintaining sparsity.
- Detecting infeasiblity
- Detecting unboundedness.
- Preprocessing to reduce problem size.

requires fancy data structures

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

LP Solvers

AMPL. [Fourer, Gay, Kernighan] An algebraic modeling language.
CPLEX solver. Industrial strength solver.

separate data from model

```
set PROD := beer ale;
set INGR := corn hops malt;

param: profit :=
ale  13
beer 23;

param: supply :=
corn  480
hops  160
malt 1190;

param amt: ale beer :=
corn        5  15
hops        4   4
malt       35  20;
```

beer.dat

beer.mod

```
set INGR;
set PROD;
param profit {PROD};
param supply {INGR};
param amt {INGR, PROD};
var x {PROD} >= 0;

maximize total_profit:
   sum {j in PROD} x[j] * profit[j];

subject to constraints {i in INGR}:
   sum {j in PROD} amt[i,j] * x[j] <= supply[i];
```

```
[cos226:tucson] ~> ampl
AMPL Version 20010215 (SunOS 5.7)
ampl: model beer.mod;
ampl: data beer.dat;
ampl: solve;
CPLEX 7.1.0: optimal solution; objective 800
ampl: display x;
x [*] :=  ale 12   beer 28;
```

LP Duality: Economic Interpretation

Brewer's problem. Find optimal mix of beer and ale to maximize profits.

$$
\begin{array}{llrcrcr}
\text{(P)} & \max & 13A & + & 23B & & \\
 & \text{s. t.} & 5A & + & 15B & \le & 480 \\
 & & 4A & + & 4B & \le & 160 \\
 & & 35A & + & 20B & \le & 1190 \\
 & & A & , & B & \ge & 0
\end{array}
$$

A* = 12
B* = 28
OPT = 800

Entrepreneur's problem. Buy resources from brewer at min cost.

- C, H, M = unit price for corn, hops, malt.
- Brewer won't agree to sell resources if 5C + 4H + 35M < 13.

$$
\begin{array}{llrcrcrcr}
\text{(D)} & \min & 480C & + & 160H & + & 1190M & & \\
 & \text{s. t.} & 5C & + & 4H & + & 35M & \ge & 13 \\
 & & 15C & + & 4H & + & 20M & \ge & 23 \\
 & & C & , & H & , & M & \ge & 0
\end{array}
$$

C* = 1
H* = 2
M* = 0
OPT = 800

LP Duality

Primal and dual LPs. Given real numbers a_{ij}, b_i, c_j, find real numbers x_j, y_i that optimize (P) and (D).

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

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

Duality Theorem. [Gale-Kuhn-Tucker 1951, Dantzig-von Neumann 1947]
If (P) and (D) have feasible solutions, then max = min.

LP Duality: Sensitivity Analysis

Q. How much should brewer be willing to pay (marginal price) for additional supplies of scarce resources?
A. Corn $1, hops $2, malt $0.

Q. How do I compute marginal prices (dual variables)?
A. Simplex solves primal and dual simultaneously.

↖ objective row of final simplex tableaux provides optimal dual solution!

Q. New product "light beer" is proposed. It requires 2 corn, 5 hops, 24 malt. How much profit must be obtained from light beer to justify diverting resources from production of beer and ale?
A. Breakeven: 2 ($1) + 5 ($2) + 24 ($0) = $12 / barrel.

History

1939. Production, planning. [Kantorovich]
1947. Simplex algorithm. [Dantzig]
1950. Applications in many fields.
1975. Nobel prize in Economics. [Kantorovich and Koopmans]
1979. Ellipsoid algorithm. [Khachian]
1984. Projective scaling algorithm. [Karmarkar]
1990. Interior point methods.
200x. Approximation algorithms, large scale optimization.

Simplex vs. Interior Point Methods

interior point faster when polyhedron smooth like disco ball

simplex faster when polyhedron spiky like quartz crystal

Ultimate Problem Solving Model

Ultimate problem-solving model?

- Shortest path.
- Maximum flow.
- Assignment problem.
- Min cost flow.
- Multicommodity flow.
- Linear programming.
- Semidefinite programming.
- Nash equilibrium.
- . . .
- TSP (or any NP-complete problem)

} tractable

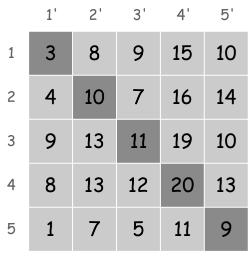
↖ complexity unknown

↖ intractable (conjectured)

Does P = NP? No universal problem-solving model exists unless P = NP.

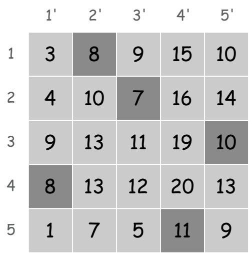
Assignment Problem

Assignment problem. Assign n jobs to n machines to minimize total cost, where c_{ij} = cost of assignment job j to machine i.



	1'	2'	3'	4'	5'
1	3	8	9	15	10
2	4	10	7	16	14
3	9	13	11	19	10
4	8	13	12	20	13
5	1	7	5	11	9

cost = 3 + 10 + 11 + 20 + 9 =53



	1'	2'	3'	4'	5'
1	3	8	9	15	10
2	4	10	7	16	14
3	9	13	11	19	10
4	8	13	12	20	13
5	1	7	5	11	9

cost = 8 + 7 + 20 + 8 + 11 = 44

Assignment Problem: Applications

Natural applications.

- Match jobs to machines.
- Match personnel to tasks.
- Match PU students to writing seminars.

Non-obvious applications.

- Vehicle routing.
- Signal processing.
- Virtual output queueing.
- Multiple object tracking.
- Approximate string matching.
- Enhance accuracy of solving linear systems of equations.

Assignment Problem: LP Formulation

$$\begin{array}{llcll} \min & \sum_{1 \le i \le n} \sum_{1 \le j \le n} c_{ij}\, x_{ij} & & & \\ \text{s. t.} & \sum_{1 \le j \le n} x_{ij} & = & 1 & 1 \le i \le n \\ & \sum_{1 \le i \le n} x_{ij} & = & 1 & 1 \le j \le n \\ & x_{ij} & \ge & 0 & 1 \le i,\, j \le n \end{array}$$

Interpretation: if x_{ij} = 1, then assign job j to machine i

Theorem. [Birkhoff 1946, von Neumann 1953] All extreme points of the above polyhedron are {0-1}-valued.

Corollary. Can solve assignment problem by solving LP since LP algorithms return an optimal solution that is an extreme point.

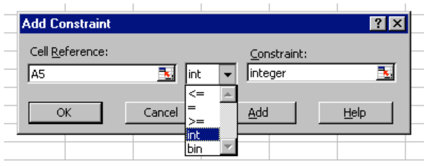
Perspective

LP is near the deep waters of NP-completeness.

- Solvable in polynomial time.
- Known for ≈ 25 years.

Integer linear programming.

- LP with integrality requirement.
- NP-hard.



Add Constraint

Cell Reference: A5

int

Constraint: integer

An unsuspecting MBA student transitions from tractable LP to intractable ILP in a single mouse click.