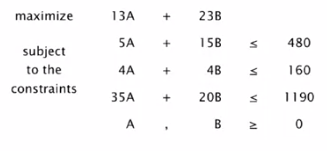
Linear Programming

Linear programming? Problem-solving model for optimal allocation of scarce resources among a number of competing activities that encompasses:

* Shortest paths, maxflow, MST, matching, assignment
* Ax = b, 2-person zero-sum games



Why significant?

* Fast commercial solvers available
* Widely applicable problem-solving model; Delta airlines claims that LP saves $100 million/ year
* Key subroutine for integer programming solvers

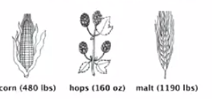
Applications:

* Environment: water quality management
* Finance: portfolio optimizations
* Logistics: supply chain management
* Marketing: direct mail advertising
* Telecommunication: network design, Internet routin

Toy LP example: Brewer’s problem

Small brewery produces ale and beer

* Production limited by scares resources: corn, hops, barley malt



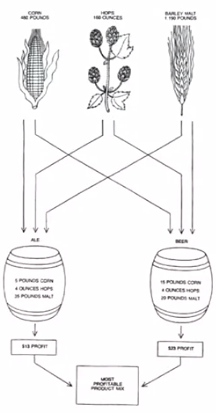
* Recipes for ale and beer require different proportions of resources



Brewer’s problem: choose product mix to maximize profits



Linear programming formulation

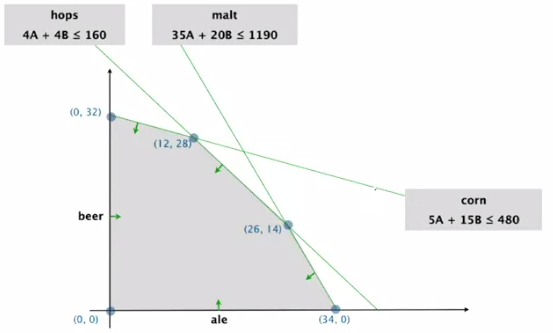


* Let A be the number of barrels of ale
* Let B be the number of barrels of beer

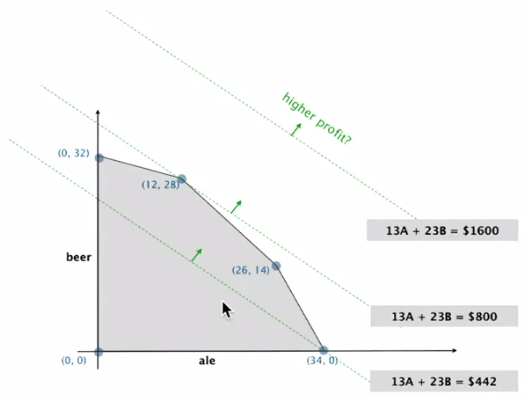


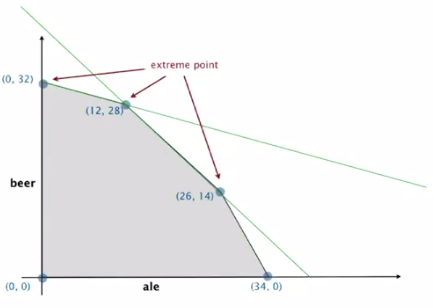
Feasible region

Inequalities define halfplanes; feasible region is a convex polygon



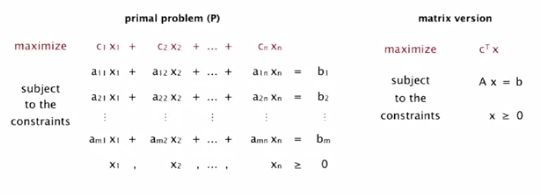
Objective function



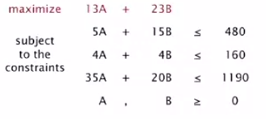
Geometry: optimal solution occurs at an extreme point (intersection of 2 constraints in 2d) 

Goal: maximize linear objective function of n nonnegative variables subject to m linear equations   
(linear means no x2, xy, arccos(x), etc.)

* Input: real numbers aij, cj, bi
* Output: real numbers xj

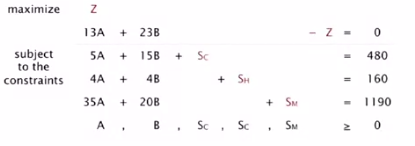


Original form of Brewer’s problem:



Standard form of Brewer’s problem:

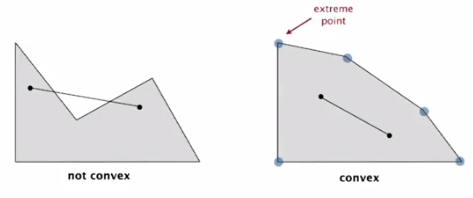
* Add variable Z and equation corresponding to objective function
* Add slack variable to convert each inequality to an equality
* Now a 6-dimensional problem



Geometry: Inequalities define **halfspaces** feasible region is a convex polyhedron

A set is convex if for any two points a and b in the set, so is ½ (a + b)

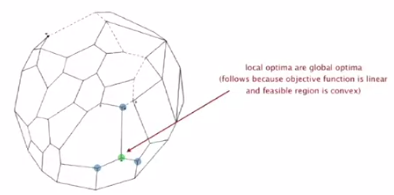
An extreme point of a set is a point in the set that can’t be written as ½ (a + b), where a and b are two distinct points in the set



Don’t always trust your intuitions for higher dimensions (than 2D)

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

* Number of extreme points to consider is finite
* But number of extreme points can be **exponential**



Greedy property: Extreme point optimal iff no better adjacent extreme point

Simplex algorithm

* Developed by George Dantzig shortly after WWII in response to logistical problems, including Berlin airlift
* Ranked as one of the top 10 scientific algorithms of the 20th century

Generic algorithm

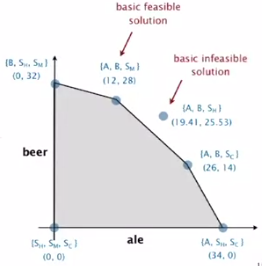
* Start at some extreme point
* Pivot from one extreme point to an adjacent one (never decreasing objective function)
* Repeat until optimal

How to implement? Linear algebra

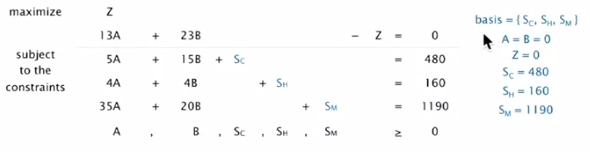
Basis: a basis is a subset of m of the n variables

Basic feasible solution (BFS)

* Set n – m nonbasic variables top 0, solve for remaining m variables
* Solve m equations in m unknowns
* If unique and feasible -> BFS]
* BFS <-> extreme point



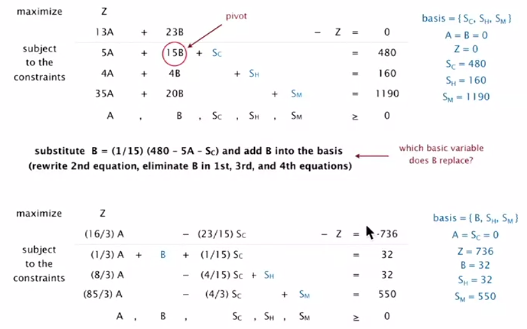
Simplex algorithm initialization



Initialize feasible solution

* Start with slack variables { SC, SH, SM } (one basic variable per row) as the basis
* Set non-basic variables A and B to 0
* 3 equations in 3 unknowns yields SC = 480, SH = 160, SM = 1190 (no algebra needed)

PIVOT 1:



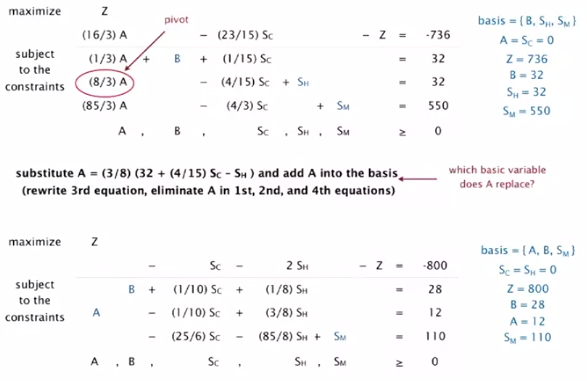
Why pivot on column 2 (corresponding to variable B)?

* It’s objective function coefficient is positive  
  (each unit increase in B from 0 increases object value by $23)
* Pivoting on column 1 (corresponding to A) also okay

Why pivot on row 2?

* Preserves feasibility by ensuring RHS >= 0
* Minimum ratio rule: min { 480/15. 160/4. 1190/20 }

PIVOT 2:



When to stop pivoting? When no objective function is positive

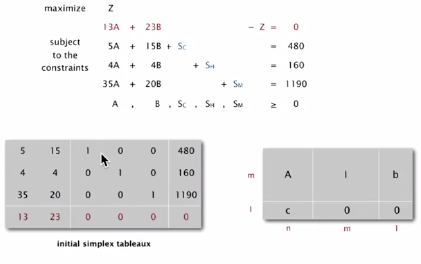
Why is resulting solution optimal?  
Any particular solution satisfies current system of equations

* In particular: Z = 800 – SC – 2 SH
* Thus, optimal objective value Z\* <= 800, since SC, SH >= 0
* Current BFS has value 800 -> optimal

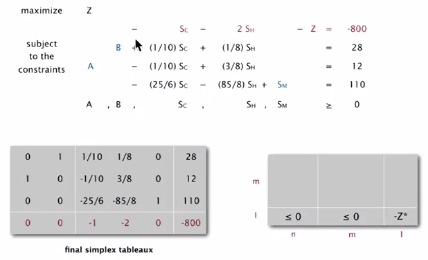
Simplex Implementations

Simplex tableau

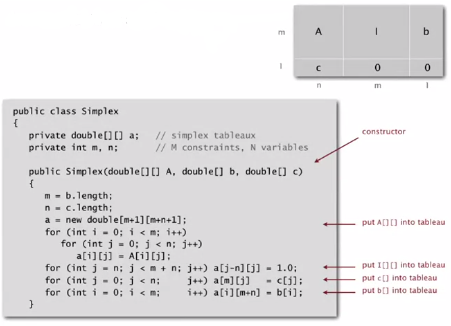
Encode standard form LP in a single Java 2D array



Simplex algorithm transforms initial 2D array into solution



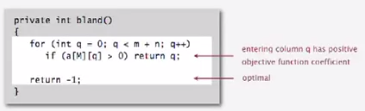
Simplex implementation in Java



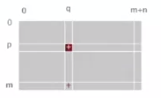
Simplex algorithm: Bland’s rule

Find entering **column** q using Bland’s rule:

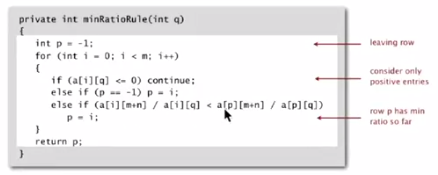
Index of first **column** whose objective function coefficient is positive



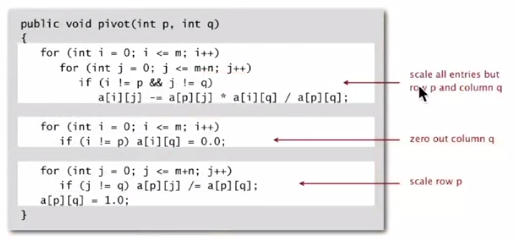
Min-ratio rule



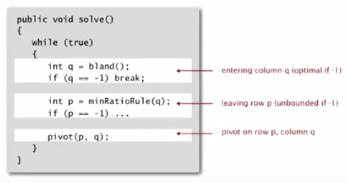
Find leaving row p using min ratio rule  
(Bland’s rule: if a tie, choose first such row)



Pivot implementation



Execute simplex algorithm

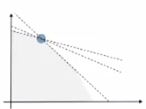


**Property**: in typical practical applications, simplex algorithm terminates after at most 2(m + n) pivots

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

* No pivot rule is known that is guaranteed to be polynomial
* Most pivot rules are known to be exponential (or worse) in worst case

Degeneracy: New basis, same extreme point (“stalling” is common in practice)



Cycling: get stuck by cycling through different bases that all correspond to the same extreme point

* Doesn’t occur in the wild
* Bland’s rule guarantees finite number of pivots (choose lowest valid index for entering and leaving columns)

Implementation issues

* Avoiding stalling (requires artful engineering)
* Maintain sparsity (requires fancy data structures)
* Numerical stability (requires advanced math)
* Detect infeasibility (run “phase i” simplex algorithm)
* Detect unboundedness (no leaving row)

*Best practice: don’t implement it yourself*

**Basic implementations** already available in many programming environments

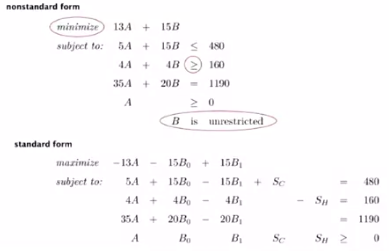
**Industrial strength solvers:** routinely solve LPs with millions of variables

**Modeling languages:** simplify task of modeling problem as LP

Linear programming reductions

Reductions to standard form

* Minimization problem: replace min 13A + 15B with max – 13A – 15B
* >= constraints: Replace 4A + 4B >= 160 with 4A + 4B – SH = 160, SH >= 0
* Unrestricted variables: Replace B with B = B0 – B1, B0 >= 0, B1 >= 0

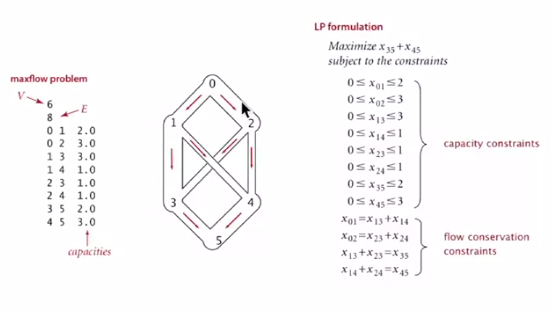


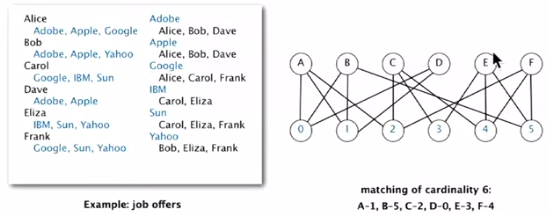
Linear “programming” (1950s term) = reduction to LP (modern term)

* Process of formulating an LP model for a problem
* Solution to LP for a specific problem gives solution to the problem

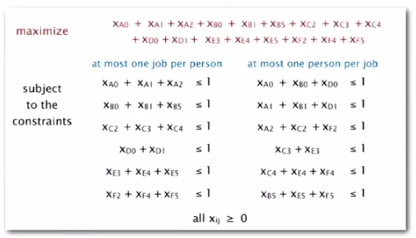
1. Identify **variables**
2. Define **constraints** (inequalities and equations)
3. Define **objective function**
4. Convert to standard form  
   *software usually performs this step automatically*

Examples:

* Shortest paths
* Maxflow:
* Bipartite matching:  
  Maximum cardinality bipartite matching problem
  + Interpretation: Mutual preference constraints
    - People to jobs
    - Students to writing seminars



* + LP formulation: one variable per pair
  + Interpretation: xij = 1 if person i is assigned to job j



* + Theorem: all extreme points of the polyhedron have integer (0 or 1) coordinates
  + Corollary: can solve matching problem by solving LP, but not usually so lucky.
* Assignment problem

If you have an optimization problem (e.g. shortest paths, maxflow, matching, etc.), 2 approaches:

* Approach 1: Use a specialized algorithm to solve it
  + Algorithms 4/e
  + Vast literature on algorithms
* Approach 2: Use linear programming
  + Many problems are easily modeled as LPs
  + Commercial solvers can solve those LPs
  + Might be slower than specialized solution (but you may not care or not have one)

Good idea to learn to use an LP solver

