451: Linear Programming

G. Miller, K. Sutner
Carnegie Mellon University
2020/11/05

Suppose we have 5 kinds of pizza toppings: pepperoni, ham, anchovies, mushrooms, eggplant that can be combined in arbitrary amounts. We measure the amount as $x_1, \ldots, x_5$, the variables of the problem (all non-negative reals). Each topping has an associated cost c_j.

Each topping contains a certain amount of key nutrients (for Domino's that would be sugar, salt, fat, ...).

Write a_{ij} for the content of nutrient i in topping j.

Lastly, assume that there is a minimal daily allowance b_i for each key nutrient.

For a reasonable diet we want

Healthy $\sum_j a_{ij} x_j \geq b_i$.

Cheap minimize $\sum_j c_i x_j$.

Note that naturally $x_1, \ldots, x_5 \geq 0$.

For example, we might have data

a_{ij}					b_i
3	4	1	0	0	5
4	5	1	1	0	15
0	0	1	2	3	20

$c = (4, 3, 5, 2, 1)$

The optimal solution here is $(0, 3, 0, 0, 20/3)$ with a cost of $47/3 \approx 15.67$.

This is one of the most important algorithmic problems, period. There are dozens of implementations, 100's books, 1000's papers, dozens of companies, 2 Nobel prizes (Kantorovich and Koopmans).

How do we solve this type of problem efficiently?

- Simplex algorithm (Dantzig 1947)
- Khachian ellipsoid method (1979)
- Karmarkar's interior point method (1984)
- Vaidya (1987/9)

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

Linear Programming (LP)[1] is a minimization/maximization problem where the objective function is linear and the constraints are linear inequalities (in either direction) or equalities.

We let n be the number of variables and m the number of constraints. The data is given as an $m \times n$ real matrix A, $m \le n$, a m-component real vector b and an n-component real vector c.

One has to find a real vector $x \in \mathbb{R}^n$ so that we

$$\text{minimize} \quad z = c^T x = c \circ x$$

subject to the constraints

$$\begin{aligned} Ax &\le b \\ x &\ge 0 \end{aligned}$$

This is sometimes called a LP problem in canonical form.

[1] Programming here simply means planning.

Inequalities are quite a bit more complicated than equalities in many ways so the special case when we only have the latter is of considerable interest.

$$\text{minimize} \quad z = c \circ x$$

subject to the constraints

$$Ax = b$$
$$x \geq 0$$

This an LP in standard form.

In our discussion of the simplex algorithm we will use standard form.

It matters little what particular version of the problem we tackle, there are ways to transform instances from one form to another in linear time.

Unconstrained Variables Split the unconstrained variable into a positive and negative part $x = x^+ - x^-$ where $0 \leq x^+, x^-$.

Inequalities $a \leq b$ iff $-a \geq -b$ iff $a + z = b$ for some $z \geq 0$
z is a slack variable, note the implicit existential quantification.

Equalities $a = b$ iff $a \leq b \leq a$.

Min/Max Use $-c$ versus c.

But note that these maneuvers can increase the dimensionality or number of constraints of the problem (which may cause computational difficulties).

Exercise

Figure out in detail how to convert a mixed LP into standard form.

To appreciate the power of LP, note that one can easily express flow problems as a LP.

Variables: x_e: flow along an edge e.

Constraints: $0 \leq x_e \leq c(e)$.

Conservation: $\sum_{e=ux} x_e = \sum_{e=xu} x_e$.

Maximize $\sum_{e=sx} x_e - \sum_{e=xs} x_e$

In terms of trying to find good algorithms for LP, this is not a good sign.

Let $\mathbb{F}$ be a field, think $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ or $\mathbb{F}_{p^k}$.

Definition

A vector space over a field $\mathbb{F}$ is a two-sorted structure $\langle V, +, \cdot, \mathbf{0} \rangle$ where

- $\langle V, +, \mathbf{0} \rangle$ is an Abelian group,
- $\cdot : \mathbb{F} \times V \to V$ is scalar multiplication subject to
  - $a \cdot (x + y) = a \cdot x + a \cdot y$,
  - $(a + b) \cdot x = a \cdot x + b \cdot x$,
  - $(ab) \cdot x = a \cdot (b \cdot x)$,
  - $1 \cdot x = x$.

In this context, the elements of V are vectors, the elements of $\mathbb{F}$ are scalars.

Note that the multiplicative subgroup of $\mathbb{F}$ acts on V on the left (the 0 in $\mathbb{F}$ is an annihilator, $0 \cdot x = 0$).

Given a vector space, the only terms we can build in this language (at least after expanding out) look like

$$c_1 v_1 + c_2 v_2 + \ldots + c_k v_k$$

where $c_i \in \mathbb{F}$ and $v_i \in V$. This is a linear combination of the v_i.

Interesting special cases:

$U \subseteq V$ is linearly independent if for all $v_i \in U$: $c_1 v_1 + c_2 v_2 + \ldots + c_k v_k = 0$ implies $c_i = 0$ for all i.

$U \subseteq V$ is spanning if for all $v \in V$ there are $v_i \in U$ and $c_i \in \mathbb{F}$ such that $c_1 v_1 + c_2 v_2 + \ldots + c_k v_k = v$.

$U \subseteq V$ is a basis for V if it is both linearly independent and spanning.

Theorem (AC)

Every vector space has a basis. Moreover, all bases have the same cardinality.

The dimension of V is the cardinality of any basis of V.

We are mostly interested in finite-dimensional spaces over $\mathbb{R}$. Suppose we have an (ordered) basis $\mathcal{B} = (v_1, v_2, \ldots, v_n)$. Then every vector v can be written uniquely as

$$v = c_1 v_1 + c_2 v_2 + \ldots + c_n v_n$$

So we might as well write $v = (c_1, c_2, \ldots, c_n)$, the usual coordinate notation.

We have $v + v' = (c_i + c_i')$ and $a \cdot v = (ac_i)$.

Strictly speaking, we need to keep track of the corresponding basis, we really should write something like

$$[v]_{\mathcal{B}} = (c_1, c_2, \ldots, c_n) \in \mathbb{F}^n$$

to display the corresponding basis.

If only one basis is in play, this is slight overkill and usually omitted.

But when dealing with multiple bases it can be confusing to leave off the reference. It is entirely standard, though.

We are mostly interested in Euclidean spaces $\mathbb{R}^n$.

We can define an inner product, a norm, and a distance that all conform to our natural geometric intuition:

$$x \circ y = \sum x_i y_i$$
$$\|x\| = \sqrt{x \circ x}$$
$$\text{dist}(x, y) = \|x - y\|$$

$$\|x + y\|^2 = \|x\|^2 + \|y\|^2 + 2\, x \circ y$$
$$x \circ y = \|x\| \, \|y\| \, \cos(\theta)$$

In terms of matrices we have $x \circ y = x^T y$.

Suppose U and V are two vector spaces over $\mathbb{F}$. A function $f : U \to V$ is linear (or a vector space homomorphism) if

- $f(x + y) = f(x) + f(y)$,
- $f(cx) = cf(x)$.

Now suppose U is n-dimensional and V is m-dimensional. Fix a basis $\mathcal{B}$ in U and $\mathcal{B}'$ in V. Then a linear map f can be represented by an $m \times n$ matrix A over $\mathbb{F}$:

$$[f(x)]_{\mathcal{B}'} = A \cdot [x]_{\mathcal{B}}$$

Again, we should write something like $A = {}_{\mathcal{B}'}[f]_{\mathcal{B}}$ but we omit the pesky subscripts when possible.

For a matrix $A \in \mathbb{F}^{m \times n}$ write $A_i^{\rightarrow}$ for the ith row in A, and $A_j^{\downarrow}$ for the jth column; similarly $A_I^{\rightarrow}$ and $A_J^{\downarrow}$ denote submatrices for $I \subseteq [m]$, $J \subseteq [n]$.

The rank of a matrix $A \in \mathbb{F}^{m \times n}$ is the dimension of the vector space spanned by all the rows of A. Why rows rather than columns?

Lemma

The rank is also the dimension of the space spanned by the columns of A.

A has full rank if its rank is $\min(n, m)$. In this case we have

- $m > n$ f is injective (a monomorphism),
- $n = m$ the matrix is invertible (non-singular), the function is a bijection (an isomorphism),
- $m < n$ f is surjective (an epimorphism)

$R \subseteq \mathbb{R}^n$ is convex if for all $x, y \in R$, the convex combination $\lambda x + (1 - \lambda) y$ is also in R, $\lambda \in [0, 1]$.

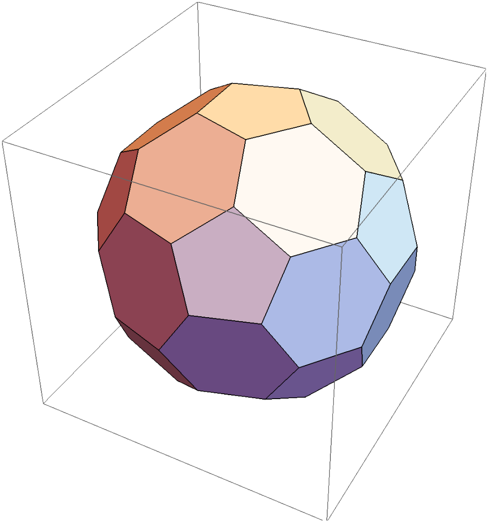
$R \subseteq \mathbb{R}^n$ is a convex polytope if it is the intersection of finitely many half-spaces in $\mathbb{R}^n$.

We are interested in polytopes of the form

$$\{ x \in \mathbb{R}^n \mid Ax \leq b \}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, $m \leq n$; in particular when the region is non-empty and bounded.

Polytopes in 2D are called polygons, and in 3D, polyhedra.



A point x in a polytope R is a vertex (or extreme point) if

$$\forall\, x \neq u, v \in R, \lambda \in [0,1]\, \big(x \neq \lambda u + (1-\lambda)v\big)$$

Equivalently,

$$\neg\exists\, d \neq 0\, \big(x \pm d \in R\big)$$

Lemma

If R is a bounded, convex polytope, then R is the convex hull of its vertices.

Exercise

Determine how many vertices are needed to write an arbitrary point as a convex combination.

For an LP in standard form we may assume that matrix A has full rank m, otherwise we can remove redundant equations. So we are dealing with the polytope

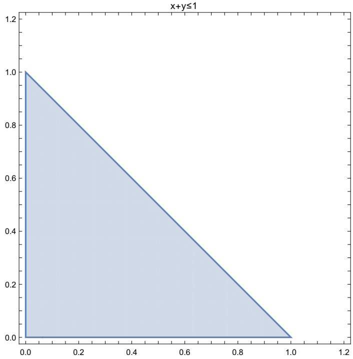
$$\{ x \in \mathbb{R}^n \mid Ax = b \}$$

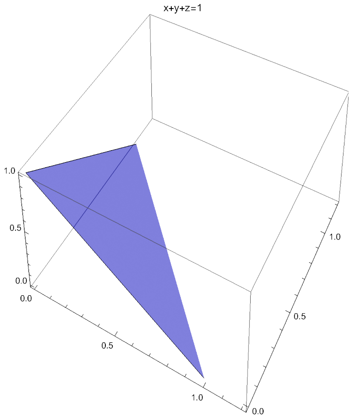
and its intersection with the first orthant $\mathbb{R}^n_{\geq 0}$.

This is called the set of feasible solutions or the simplex. Note that this is a high-dimensional object, it is not a priori clear how to compute feasible solutions, much less optimal ones.

For geometric intuition, it may be preferable to look at canonical form:

$$\{ x \in \mathbb{R}^n \mid Ax \leq b, 0 \leq x \}$$





The slack variable increases the dimension to 3.

As we have already seen in 2D-LP, there are several potential issues to contend with.

- The set of feasible solutions is empty.
- The set of feasible solutions is unbounded.
- There are several optimal solutions.

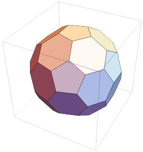
For the time being, we will focus on the the critical task: we have to develop machinery to perform "geometric operations" relating to the simplex, using only arithmetic (so everything can be executed on a RAM).

In a nutshell, this algorithm is an iterative procedure that moves from vertex to vertex on the simplex S, decreasing the objective function, until a minimum is reached.

So we need to find a vertex of the feasible region. Then we iterate:

- Consider all immediate neighbors of the current vertex.
- If none of them provides a better value for the objective function, stop.
- Otherwise pick one that does, and repeat.

Convention:
We assume an LP in standard form $Ax = b$ where A has full rank $m \leq n$.



Lemma

If there is an optimal solution, then at least one vertex is an optimal solution.

But bear in mind that any convex combination of optimal solutions is again optimal, so a whole edge, face, ... of the polytope may also be optimal.

Intuitively, for the proof, we start from any optimal solution x, and march in a non-increasing direction until we reach the boundary of the feasible region S. Repeat till we reach a vertex.

For a vector x write $\mathrm{sp}(x) = \{ j \in [n] \mid x_j \neq 0 \}$ for the support of x.

Suppose x is not a vertex and let $d \neq 0$ with $x \pm d \in S$; hence $Ad = 0$. Note that sp$(x) \subseteq$ sp(d) since $x \pm d \geq 0$.

Wlog $c \circ d \leq 0$.

Case 1: $d_j < 0$ for some j

Increase λ until some component of $x + \lambda d$ hits 0 from above; call this λ'. Then $x' = x + \lambda' d$ is feasible and has (at least) one more 0 component than x. Furthermore, $c \circ x' \leq c \circ x$. Rinse and repeat.

Case 2: $d \geq 0$.

This time, $x + \lambda d$ is feasible for all $\lambda \geq 0$.

But $c \circ d < 0$, otherwise we are essentially back in case 1. Hence $\lim c \circ (x + \lambda d) = -\infty$ as $\lambda \to \infty$. But then there is no optimal solution.

$\square$

There is a simple way to characterize the vertices of the polytope S in terms of the column vectors of A.

Lemma

x is a vertex of R iff the vectors $A_j^{\downarrow}$, $j \in \mathsf{sp}(x)$, are linearly independent.

Given $J \subseteq [n]$, write $A_J = A_J^{\downarrow}$ and $A_{\overline{J}}$ for the remaining part.
Similarly write x_J and $x_{\overline{J}}$ for a vector x.

Proof.

First suppose the $A_j^{\downarrow}$, $j \in J = \mathsf{sp}(x)$, are linearly dependent. Then $A_J\, d_J = 0$ for some $d \neq 0$ and $\mathsf{sp}(d) \subseteq \mathsf{sp}(x)$.

But then $A\, d = 0$ and $x + \lambda d \in R$ for sufficiently small λ. Thus x is not a vertex.

For the opposite direction assume x is not a vertex.

As before, there is some $d \neq 0$ such that $x \pm d \in S$ and we have $Ad = 0$, $J' = \mathsf{sp}(d) \subseteq \mathsf{sp}(x) = J$.

But then $A_{J'}$ has linearly dependent columns, and is a submatrix of A_J, done.

□

By adding a few independent columns if necessary, we get to following result.

Corollary

The vertices of R are exactly of the form

$$x_J = A_J^{-1} b \qquad x_{\overline{J}} = 0$$

where $J \subseteq [n]$, $|J| = m$, A_J non-singular.

For intuition, think about the simple case $J = [m]$. We can rewrite the system as

$$A\,x = \left(\; A_J \;\middle|\; A_{\overline{J}} \;\right) \binom{x_J}{x_{\overline{J}}}$$

where the blocks are $m \times m$ and $m \times (n-m)$, respectively (the basic and non-basic part of A). To concoct a vertex of the simplex, we can set $x_J = A_J^{-1}\, b$ and $x_{\overline{J}} = 0$.

In a sense, this particular form is really general: we can just reorder the basis. Coordinates are an artifact, anyway.

Alas, in a real algorithm we can't just say "up to isomorphism . . ."

Again: suppose that $A_J \in \mathbb{R}^{m \times m}$ is non-singular. We obtain the corresponding vertex $x = \mathsf{bfs}(J)$ of the simplex by

$$x_J = A_J^{-1} b$$

$$x_{\overline{J}} = 0$$

This is called the basic solution associated with J.

If, in addition, $x \geq 0$, then x is a basic feasible solution (bfs) satisfying all the constraints in the LP.

Of course, we also have to deal with the objective function.

We already know that it suffices to consider bfs only (though it's far from clear that this is a good strategy computationally). This would suggest an approach along the lines of

- Somehow get our hands on some bfs.
- Then repeatedly find a better bfs, until we reach an optimal one.

Let's ignore the first part for the time being and focus on how to improve a bfs.

We need to modify a basis J to a new basis J' such that

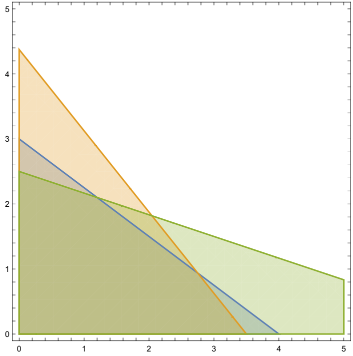
$$c \circ \mathsf{bfs}(J') < c \circ \mathsf{bfs}(J).$$

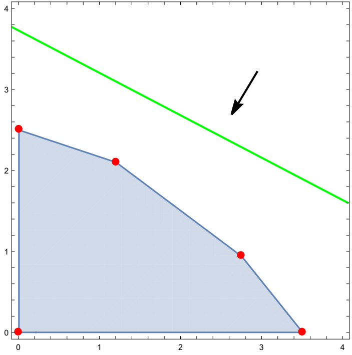
until we get to an optimal solution.

Question: What changes to the basis should we consider?

Luckily, we can restrict ourselves to an operation called pivoting where one column leaves the basis and a new one enters it. This makes perfect sense geometrically, since it corresponds to moving from a bfs vertex to an *adjacent* bfs vertex.

Problem: We need to determine a column $l \in J$ to leave, and a column $k \notin J$ to enter the new basis.





The original problem has $m = 3$, $n = 2$ in canonical form.

We add 3 slack variables to get standard form. Note that the objective function ignores these. Also, it's easy to get a starting point: $J = \{3, 4, 5\}$. Here is a little tableaux that summarizes the data:

-1	-2	0	0	0	0
3	4	1	0	0	12
10	8	0	1	0	35
2	6	0	0	1	15

Note that $A_{\{3,4,5\}}$ is the identity matrix, so "computing" the inverse is trivial.

Also, in this case, any matrix A_J for $|J| = 3$ has rank 3.

Of course, not all of them satisfy the non-negativity constraints, e.g. bfs$(1, 4, 5) = (4, 0, 0, -5, 7)$.

The good bfs correspond to the 5 vertices of the simplex.

J	x_J	$c \circ x$
$\{1,2,4\}$	$\left(\frac{6}{5}, \frac{21}{10}, \frac{31}{5}\right)$	-5.4
$\{1,2,5\}$	$\left(\frac{11}{4}, \frac{15}{16}, \frac{31}{8}\right)$	-4.625
$\{1,3,5\}$	$\left(\frac{7}{2}, \frac{3}{2}, 8\right)$	-3.5
$\{2,3,4\}$	$\left(\frac{5}{2}, 2, 15\right)$	-5
$\{3,4,5\}$	$(12, 35, 15)$	0

To get to the optimal bfs we could do

$$\{3,4,5\} \rightsquigarrow \{1,3,5\} \rightsquigarrow \{1,2,5\} \rightsquigarrow \{1,2,4\}$$

or

$$\{3,4,5\} \rightsquigarrow \{2,3,4\} \rightsquigarrow \{1,2,4\}$$

A tableaux[2] is is a nice way to organize the constraints and objective function into a single table. We want to maintain $A_J = I$

$$
\begin{array}{ccccccc}
c_J \circ x_J & + & c_{\overline{J}} \circ x_{\overline{J}} & = & -z & & \min z \\
A_J \circ x_J & + & A_{\overline{J}} \circ x_{\overline{J}} & = & b & & \\
x_J & , & x_{\overline{J}} & \geq & 0 & &
\end{array}
$$

After a few steps, J and $\overline{J}$ will be interleaved, and the unit vectors in A_J will not be in the natural order, so the table will look less pretty.

Since the non-negativity constraints never change we might as well drop them to avoid visual clutter.

Note the $-z$ in the top right corner, this makes it easier to perform the necessary row operations on the tableaux.

[2] Traditionally, our last column is placed on the left, "column 0."

Original tableaux with slack variables, $J = \{3, 4, 5\}$:

-1	-2	0	0	0	0
3	4	1	0	0	12
10	8	0	1	0	35
2	6	0	0	1	15

After pivoting with $\ell = 5$ and $k = 2$, $J = \{2, 3, 4\}$:

$-1/3$	0	0	0	$1/3$	5
$5/3$	0	1	0	$-2/3$	2
$22/3$	0	0	1	$-4/3$	15
$1/3$	1	0	0	$1/6$	$5/2$

After pivoting with $\ell = 3$ and $k = 1$, $J = \{1, 2, 4\}$:

0	0	$1/5$	0	$1/5$	$27/5$
1	0	$3/5$	0	$-2/5$	$6/5$
0	0	$-22/5$	1	$8/5$	$31/5$
0	1	$-1/5$	0	$3/10$	$21/10$

Step 1:

−1	−2	0	0	0	0
3	4	1	0	0	12
10	8	0	1	0	35
1/3	1	0	0	1/6	5/2

Step 2:

−1	−2	0	0	0	0
3	4	1	0	0	12
22/3	0	0	1	−4/3	15
1/3	1	0	0	1/6	5/2

Step 3:

−1	−2	0	0	0	0
5/3	0	1	0	−2/3	2
22/3	0	0	1	−4/3	15
1/3	1	0	0	1/6	5/2

Step 4:

−1/3	0	0	0	1/3	5
5/3	0	1	0	−2/3	2
22/3	0	0	1	−4/3	15
1/3	1	0	0	1/6	5/2

Suppose we have a $x = \text{bfs}(J)$ and we want to swap in column k. Since J is a basis there must be some coefficient vector ξ such that

$$A_J x_J = b$$
$$A_J \xi_J = A_k^{\downarrow}$$

We want to increase x_k to some $\theta > 0$ st

$$\theta A_k^{\downarrow} + A_J x'_J = b$$

whence

$$x'_J = x_J - \theta\xi$$

Because of our non-negativity constraints we get for $\xi_j > 0$

$$\theta \leq x_j/\xi_j$$

So we will set $\theta = \min\big(x_j/\xi_j \mid \xi_j > 0 \big)$. If the minimum occurs at ℓ, column ℓ leaves the basis (min ratio).

- There is no $j \in J$ such that $\xi_j > 0$.
  Then we can choose $\theta \geq 0$ arbitrarily.

- We have $\theta = 0$.
  In this case we get $x' = x$ but we have changed the basis and we are dealing with a degenerate solution.

A bfs is degenerate if $|\text{sp}(x)| < m$. This happens in particular when the minimization produces more than one ℓ.

We will talk about this case next time.

We have a method to choose the leaving column ℓ given the entering column k. How should one pick $k \notin J$?

We want to make progress, so $c \circ x' < c \circ x$ where

$$c \circ x' = \theta c_k + c_J \circ (x - \theta \xi)$$

$$c \circ x = c_J x$$

$$\Delta = \theta(c_k - c_J \xi)$$

Thus we want $c_k - c_J \xi$ to be negative.

Here is another look at the objective function.

By minor abuse of notation, we can occasionally think of v_J and $v_{\overline{J}}$ as vectors in $\mathbb{R}^n$: just fill the missing positions with 0's. By splitting a vector as $v = v_J + v_{\overline{J}}$ we have

$$c \circ v = c_J \circ v_J + c_{\overline{J}} \circ v_{\overline{J}}$$

For a general solution $Av = b$ we get

$$v_J = A_J^{-1}\big(b - A_{\overline{J}} v_{\overline{J}}\big)$$

Substituting, we obtain

$$\begin{aligned} c \circ v &= c_J \circ A_J^{-1} b + (c_{\overline{J}} - c_J^T A_J^{-1} A_{\overline{J}}) \circ v_{\overline{J}} \\ &= z_0 + p \circ v_{\overline{J}} \end{aligned}$$

Note that in

$$c \circ v = z_0 + p \circ v_{\overline{J}}$$

the first term z_0 depends only on A_J but not on v. If v is the bfs for J it is the current cost. In the second term, $p = p(J)$ is called the relative cost vector.

We can compute p by solving $A_J^T \lambda = c_J$ (simplex multipliers) and then exploiting

$$c_J^T A_J^{-1} A_{\overline{J}} = \lambda^T A_J A_J^{-1} A_{\overline{J}} = \lambda^T A_{\overline{J}}$$

In particular when x is a bfs and we try to maximize the difference in cost between x' and x we need to maximize

$$p \circ x'_{\overline{J}}$$

We can use this to guide pivoting.

Here is a version of the old pizza problem, of dimension 3×5:

$$A = \begin{pmatrix} 3 & 4 & 1 & 0 & 0 \\ 3 & 5 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 & 3 \end{pmatrix} \qquad b = \begin{pmatrix} 5 \\ 15 \\ 20 \end{pmatrix} \qquad c = (4, -3, 5, 2, 1)$$

We choose the basis $J = \{2, 3, 4\}$, so

$$A_J = \begin{pmatrix} 4 & 1 & 0 \\ 5 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} \quad A_{\overline{J}} = \begin{pmatrix} 3 & 0 \\ 3 & 0 \\ 0 & 3 \end{pmatrix}$$

The bfs associated to J and its cost are is

$$\begin{aligned} x &= (0, 5/6, 5/3, 55/6, 0) \\ c \circ x &= 145/6 \approx 24.17 \end{aligned}$$

For the relative cost vector we solve $A_J^T\,\lambda = c_J$ which produces

$$\lambda = (43/6, -19/3, 25/6)$$

and therefore

$$p = (3/2, -22/3)$$

The second component of p corresponds to column 5, so we are going to use $k = 5$.

Solving $A_J^T \xi = A_5^{\downarrow}$ we get

$$\xi = (-1/2, 2, 1/2)$$

For the min ratio test we have pairs

j	2	3	4
x_j	$5/6$	$5/3$	$55/6$
ξ_j	$-1/2$	2	$1/2$

Hence we get $\ell = 3$ and

$$\theta = 5/6$$

The bfs for $J' = \{2, 4, 5\}$ is

$$x' = (0, 5/4, 0, 35/4, 5/6)$$
$$c \circ x' = 175/12 \approx 14.58$$

This is the optimal solution.

From the tableaux approach, we are in good shape if the instance is in canonical form with $b \geq 0$.

In general, we can use the following two-phase approach.

First, reduce $(A \mid b)$ to row-echelon form R and let $m' = \mathsf{rk}(R)$. If the last entry in $R_i^{\rightarrow}$ is not 0 for some $i > r$, return "infeasible."

Otherwise, remove all rows $R_i^{\rightarrow}$, $i > r$. By slight abuse of notation, call the result A and b.

Add m slack variables and solve

$$\begin{aligned} \min\ \sum z_i \\ A\,x + I z = b \\ x, z \geq 0 \end{aligned}$$

Note that we know how to start in this case.

Note that the old problem is feasible iff the new problem has optimal value 0.

So if we get some value larger than 0 we return "infeasible."

Otherwise we remove the slack variables, and use the original x variables to start phase II and work on the original problem.

Strictly speaking, this assumes we used simplex to solve the last problem. If the solution is obtained some other way, extra work is needed to get a bfs.

Recall that we need to bound the coefficient θ of the entering variable: we compute

$$\theta = \min\big(\, x_j/\xi_j \mid \xi_j > 0 \,\big)$$

If θ is unbounded, the simplex is unbounded.

If the bound is 0, we are dealing with a degenerate solution: we have two bases that produce the same bfs.

This could lead to non-termination, we could cycle through the same bases without ever making progress, depending on the method we use to pick the next pivot.

Smallest Coefficient

In the objective function, pick the least coefficient.

Largest Decrease

Maximize the decrease in the value of the objective function.

Maximum Gradient

Pick the variable so as to minimize

$$\frac{c \circ (x' - x)}{\|x' - x\|}$$

Bland's Rule

Always pick the eligible variable of least possible index.

It requires a bit of work that this guarantees termination.

Randomize

Pick an eligible variable at random.

Perturbation

Consider $A\,x = b + \varepsilon$ where $\varepsilon_i \gg \varepsilon_{i+1}$ (e.g., $\varepsilon_i = \delta^i$).

Then manipulate rows symbolically and use inequalities on the ε_i.