Lecture 8

Linear programming

Kin Cheong Sou
Department of Mathematical Sciences
Chalmers University of Technology and Göteborg University
November 17, 2013

CHALMERS | GÖTEBORGS UNIVERSITET

Linear programs (LP)

Consider a linear program (LP):

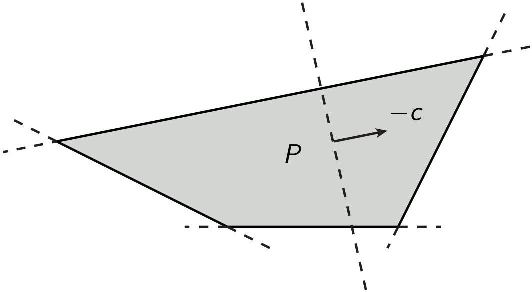
$$\begin{aligned} z^* = \text{infimum} \quad & c^T x, \\ \text{subject to} \quad & x \in P, \end{aligned}$$

where P is a polyhedron (i.e., $P = \{x \mid Ax \leq b\}$).

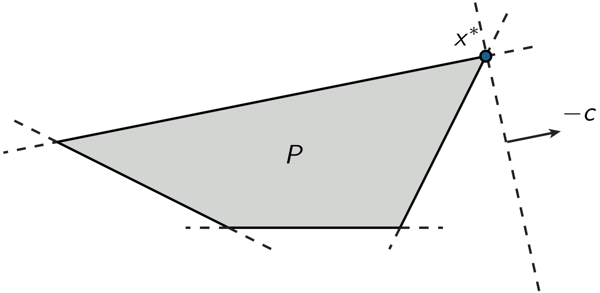
- $A \in \mathbb{R}^{m \times n}$ is a given matrix, and b is a given vector,
- Inequalities interpreted entry-wise (i.e., $(Ax)_i \leq (b)_i, \; i = 1, \ldots, m$),
- Minimize a linear function, over a polyhedron (linear constraints).

Linear programs (LP)

$$
\begin{aligned}
z^* = \text{infimum} \quad & c^T x, \\
\text{subject to} \quad & x \in P,
\end{aligned}
$$



$$
\begin{aligned}
z^* = \text{infimum} \quad & c^T x, \\
\text{subject to} \quad & x \in P,
\end{aligned}
$$



Polyhedra in different forms

Inequality constraints $Ax \leq b$ (i.e., $x \in P$) might look restrictive, but in fact more general:

- $x \geq \mathbf{0}^n \iff -I^n x \leq \mathbf{0}^n$,
- $Ax \geq b \iff -Ax \leq -b$,
- $Ax = b \iff Ax \leq b \quad \text{and} \quad -Ax \leq -b$.

In particular, we often consider **polyhedron in standard form**:

$$P = \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq \mathbf{0}^n\}.$$

P is a polyhedron, since $P = \{x \in \mathbb{R}^n \mid \tilde{A}x \leq \tilde{b}\}$ for some $\tilde{A}$ and $\tilde{b}$.

Standard form linear programs

We say that a LP is written in **standard form** if

$$\begin{aligned} z^* = \text{infimum} \quad & c^T x, \\ \text{subject to} \quad & Ax = b, \\ & x \geq \mathbf{0}. \end{aligned}$$

- Meaning that $P = \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq \mathbf{0}\}$.
- Only considering nonnegative variables and equality constraints.
- But standard form LP can in fact model all LP's.

Rewriting to standard form

- For example, if $P = \{x \in \mathbb{R}^n \mid Ax \leq b,\ x \geq \mathbf{0}\}$, we can add **slack variables**, s, in order to write P on standard form.

$$\begin{aligned} Ax &\leq b, \\ x &\geq \mathbf{0}^n \end{aligned} \iff \begin{aligned} Ax + I^m s &= b, \\ x &\geq \mathbf{0}^n \\ s &\geq \mathbf{0}^m \end{aligned} \iff \begin{aligned} [A\ I^m]\, v &= b, \\ v &\geq \mathbf{0}^{n+m} \end{aligned}$$

- If some variable x_j is free of sign, substitute it everywhere by

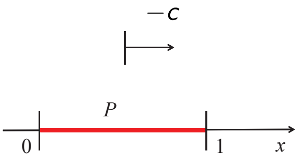
$$x_j = x_j^+ - x_j^-$$

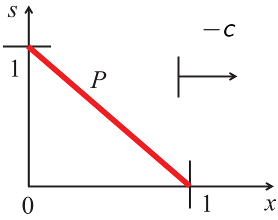
where $x_j^+, x_j^- \geq 0$

- We also assume that $b \geq \mathbf{0}^m$. If some element of b is negative, multiply that constraint by -1.

Rewriting to standard form, example

$$\begin{array}{ll} \text{minimize} & -2x \\ \text{subject to} & x \leq 1 \\ & x \geq 0 \end{array} \qquad \begin{array}{ll} \text{minimize} & -2x \\ \text{subject to} & x + s = 1 \\ & x, s \geq 0 \end{array}$$

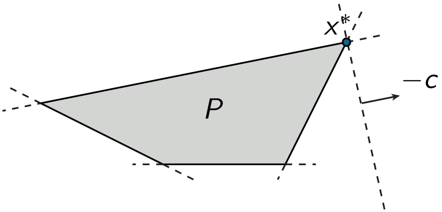




Equivalent LP's, but different polyhedra!

Optimality, a geometric motivation

Minimizing $c^T x$ over $P \implies$ an optimal solution at a **vertex** of P (the unique optimal solution over P minimizing $c^T x$ for some c):

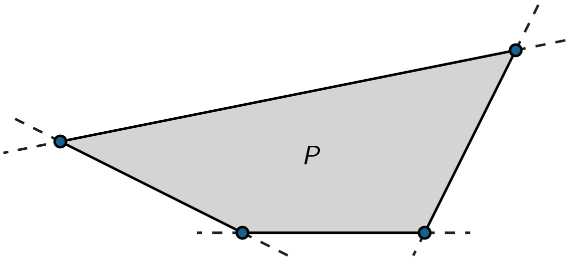


If P has a vertex and if LP has an optimal solution,
then at least one optimal solution is a vertex.

A point x is a vertex of $P \iff x$ is an **extreme point** of P.

Extreme points

An **extreme point** of a convex set S is a point that cannot be written as a convex combination of two other points in S.



Extreme points solution candidates

We focus on extreme points when searching for optimal solutions, but

- When does polyhedron $\{x \in \mathbb{R}^n \mid Ax \leq b\}$ have extreme points?
- Extreme points are geometrical objects... hard to put in algorithm

Work with LP in **standard form** (i.e., $P = \{x \mid Ax = b, x \geq \mathbf{0}\}$) because

A nonempty polyhedron in standard form always has an extreme point

Instead of geometric objects such as extreme points, we work with their algebraic equivalence – **basic feasible solution** in standard form LP.

- We will show, indeed, **basic feasible solutions are extreme points and vice versa**.

Basic feasible solution (BFS)

Standard form polyhedron $P = \{x \mid Ax = b, x \geq \mathbf{0}\}$, $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = m$

A point $\bar{x}$ is a **basic feasible solution** (BFS) if

- $\bar{x} \geq \mathbf{0}$, and $\bar{x}$ is a **basic solution**.

A point $\bar{x}$ is a **basic solution** if

- $A\bar{x} = b$, and
- the columns of A corresponding to non-zero components of $\bar{x}$ are linearly independent (and extendable to a basis of $\mathbb{R}^m$). (Recall that: $A\bar{x} = \sum_{j=1}^n a_j \bar{x}_j$, where a_j is column j of A.)

Basic and non-basic variables

For any BFS $\bar{x}$, we can reorder the variables according to

$$\bar{x} = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, \quad A = (B, N), \quad c = \begin{pmatrix} c_B \\ c_N \end{pmatrix},$$

such that

- $B \in \mathbb{R}^{m \times m}$, $\operatorname{rank}(B) = m$.
- $x_N = \mathbf{0}^{n-m}$.
- $x_B = B^{-1}b$ (as a consequence of $A\bar{x} = Bx_B + Nx_N = b$).

We call

- x_B the **basic variables**. If $x_B \not> \mathbf{0}$ then BFS $\bar{x}$ is called **degenerate**.
- x_N the **non-basic variables**.
- B the **basis matrix**. Each BFS is associated with at least one basis.

BFS = extreme point

Theorem
Assume $\text{rank}(A) = m$. A point $\bar{x}$ is an extreme point of the set $\{x \in \mathbb{R}^n \mid Ax = b,\ x \geq \mathbf{0}\}$ if and only if it is a basic feasible solution.

Proof: We show it on blackboard, or consult Theorem 8.7 in text.

Thus,

- "vertex = extreme point = basic feasible solution (BFS)".
- So, we focus optimal solution search in BFS's (extreme points). Now let's formally show that the restriction is justified!

Representation theorem, standard form

- $P = \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq \mathbf{0}\}$ (i.e., polyhedron in standard form)
- $V = \{v^1, \dots, v^k\}$ be the extreme points of P
- $C = \{x \in \mathbb{R}^n \mid Ax = \mathbf{0},\ x \geq \mathbf{0}\}$
- $D = \{d^1, \dots, d^r\}$ be the extreme directions of C

Representation Theorem (standard form polyhedron)
Every point $x \in P$ is the sum of a convex combination of points in V and a non-negative linear combination of points in D, i.e.

$$x = \sum_{i=1}^{k} \alpha_i v^i + \sum_{j=1}^{r} \beta_j d^j,$$

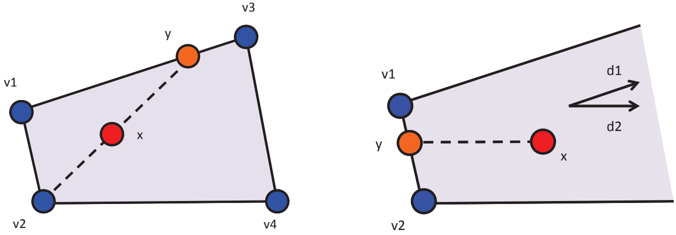
where $\alpha_1, \dots, \alpha_k \geq 0$, $\sum_{i=1}^{k} \alpha_i = 1$ and $\beta_1, \dots, \beta_r \geq 0$

Proof: See text Theorem 8.9 (In the proof, Th. 8.9 should be Th. 3.26).

Illustration of representation theorem

Representation theorem provides “inner representation” of polyhedron.

- (a) x is convex combo. of v^2 and y, and y is convex combo. of v^1 and v^3 $\implies$ x is convex combo. of v^1, v^2 and v^3.
- (b) x is convex combo. of v^1 and v^2, plus $\beta_2 d^2$.



(a) Bounded case (b) Unbounded case

Optimality of extreme points

Now we can present the theorem regarding optimality of extreme points

Theorem

Consider the standard form LP problem

$$\begin{aligned} z^* = \text{infimum} \quad & z = c^T x, \\ \text{subject to} \quad & x \in P, \end{aligned}$$

This problem has a finite optimal solution if and only if P is nonempty and z is bounded on P, meaning that $c^T d^j \geq \mathbf{0}$ for all $d^j \in D$

Moreover, if the problem has a finite optimal solution, then there exists an optimal among the extreme points.

Proof: We show it on blackboard, or see Theorem 8.10 in text.

Summary so far

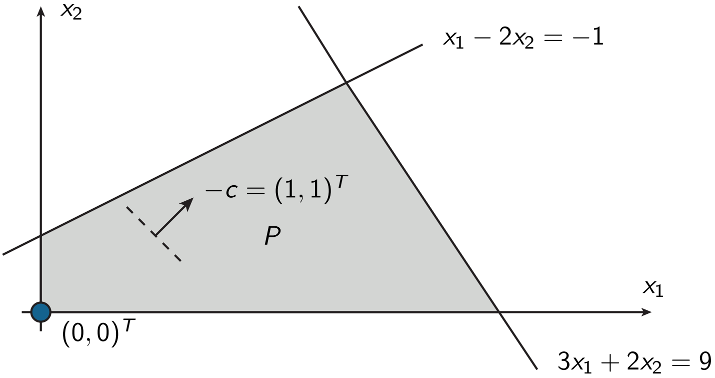
So far, we have seen

- All linear programs can be written in standard form.
- Vertex $=$ extreme point $=$ basic feasible solution (BFS).
- If a standard form LP has finite optimal solution, then it has an optimal BFS.

We finally have rationale to search only the BFS's to solve a standard form LP. This is the main characteristic of the **simplex algorithm**.

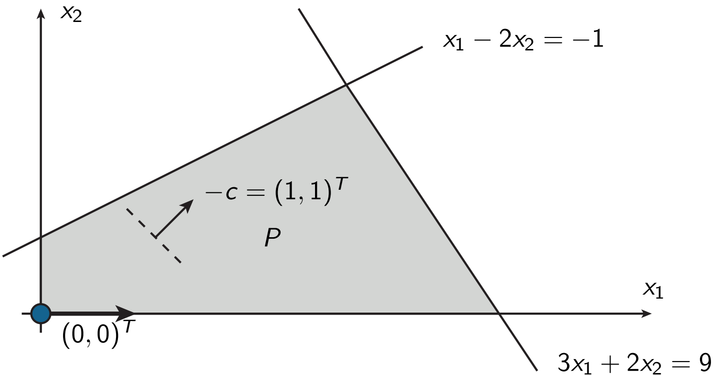
Graphic illustration

Start at a BFS, in this case $(0,0)^T$.



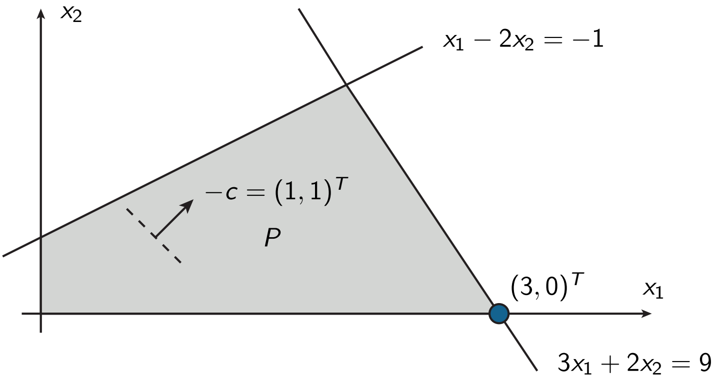
Graphic illustration

Find a feasible descent direction towards an adjacent BFS.



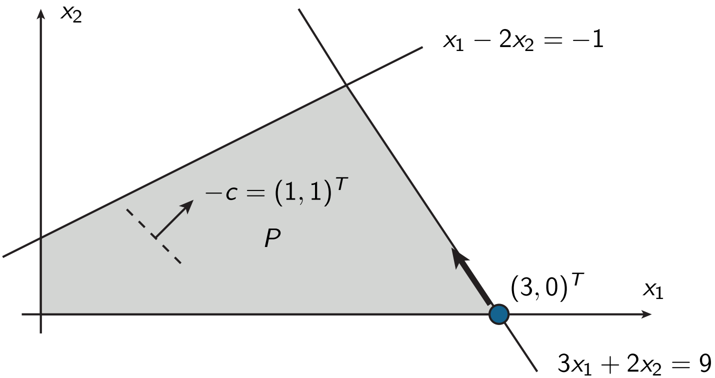
Graphic illustration

Move along the search direction until a new BFS is found.



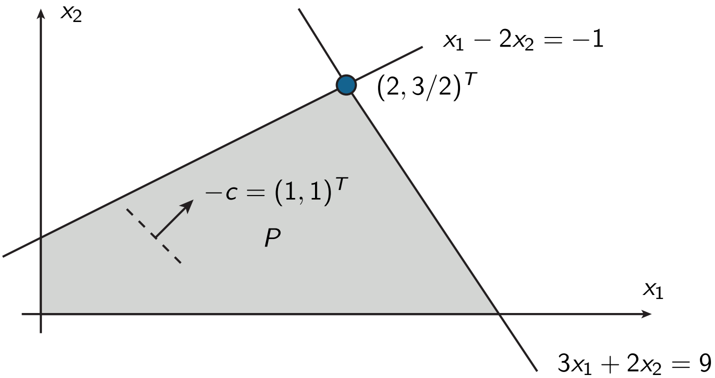
Graphic illustration

Find a new feasible descent direction at the current BFS.



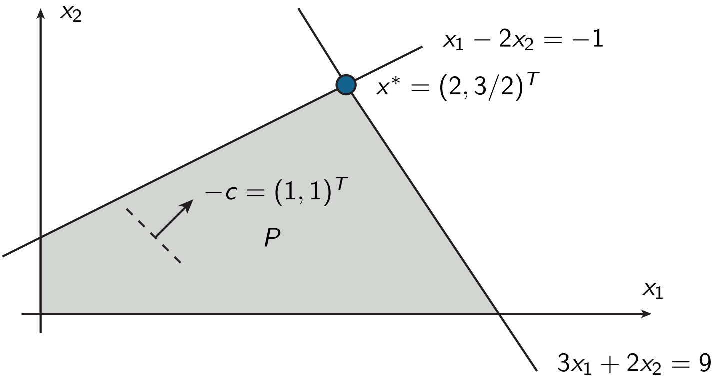
Graphic illustration

Move along the search direction.



Graphic illustration

If no feasible descent directions exist, the current BFS is declared optimal.



Developing the simplex algorithm

To develop the simplex algorithm, we translate geometric intuition into algebraic manipulations. We need to...

- Find a feasible descent direction at any BFS.
- Determine the step size to move along a feasible descent direction.
- Certify optimality at an optimal BFS.

Basic feasible solution, recap

A BFS $\bar{x}$ satisfies $A\bar{x} = b$, $\bar{x} \geq \mathbf{0}$, can reorder the variables such that

$$\bar{x} = \begin{pmatrix} x_B \\ x_N \end{pmatrix}, \quad A = (B, N), \quad c = \begin{pmatrix} c_B \\ c_N \end{pmatrix},$$

such that

- **Basis matrix** $B \in \mathbb{R}^{m \times m}$, $\text{rank}(B) = m$.
- **Non-basic variables** $x_N = \mathbf{0}^{n-m}$.
- **Basic variables** $x_B \geq \mathbf{0}$, $x_B = B^{-1}b$ (as a consequence of $A\bar{x} = b$).

We call a BFS $\bar{x}$ **degenerate** if $x_B \not> \mathbf{0}$. $\bar{x}$ **non-degenerate** if $x_B > \mathbf{0}$.

Basic feasible solution, example

Consider standard form LP

$$\begin{array}{rcccccccccc}
\text{minimize} & 2x_1 & + & 0\cdot x_2 & + & 0\cdot x_3 & + & 0\cdot x_4 & & \\
\text{subject to} & x_1 & + & x_2 & + & x_3 & + & x_4 & = & 2 \\
 & 2x_1 & + & 0\cdot x_2 & + & 3x_3 & + & 4x_4 & = & 2 \\
 & x_1, & & x_2, & & x_3, & & x_4 \geq 0 & &
\end{array}$$

An example BFS has basic variables $x_B = (x_1, x_2)$ and nonbasic variables $x_N = (x_3, x_4)$. $A = (B, N)$ with

$$B = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}, \quad N = \begin{pmatrix} 1 & 1 \\ 3 & 4 \end{pmatrix}, \quad x_B = B^{-1}b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad x_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

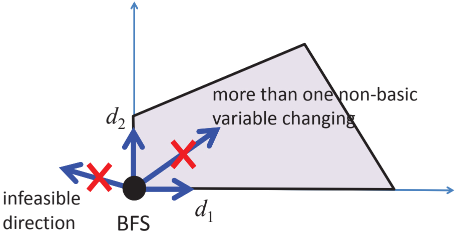
- At BFS $\bar{x} = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ with $A = (B, N)$; iterate update $\bar{x} \leftarrow \bar{x} + \theta d$ for step-size $\theta \geq 0$ with search direction $d = \begin{pmatrix} d_B \\ d_N \end{pmatrix}$.
- **Update only one non-basic variable**: Let $d_N = e_j$, for $j = 1, \ldots, n-m$ (e_j is the j-th unit vector in $\mathbb{R}^{n-m}$).
- d_B is not arbitrary – it is decided by feasibility of $\bar{x} + \theta d$:
  - $A(\bar{x} + \theta d) = b \implies Ad = 0 \implies d_B = -B^{-1}N_j$
  - search directions become $d_j = \begin{pmatrix} -B^{-1}N_j \\ e_j \end{pmatrix}$, $j = 1, \ldots, n-m$

e.g., $B = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}$, $\quad N = \begin{pmatrix} 1 & 1 \\ 3 & 4 \end{pmatrix}$, $\quad d_1 = \begin{pmatrix} -3/2 \\ 1/2 \\ 1 \\ 0 \end{pmatrix}$, $\quad d_2 = \begin{pmatrix} -2 \\ 1 \\ 0 \\ 1 \end{pmatrix}$

Search direction, in picture

Consider feasible search directions changing only one non-basic variable.

That is, $d_j = \begin{pmatrix} -B^{-1}N_j \\ e_j \end{pmatrix}, \quad j = 1, \ldots, n-m.$



- We will show d_j indeed goes along the edge of polyhedron.
- Question: Is direction d_j going to decrease the objective value?

Reduced costs

- From $\bar{x}$ to $\bar{x} + \theta d_j$, objective value change is

$$c^T(\bar{x} + \theta d_j - \bar{x}) = \theta \cdot c^T d_j = \theta \cdot (c_B^T, c_N^T) \begin{pmatrix} -B^{-1}N_j \\ e_j \end{pmatrix} := \theta \cdot (\tilde{c}_N)_j$$

$(\tilde{c}_N)_j := (c_N^T - c_B^T B^{-1} N)_j$ is the **reduced cost** for non-basic var $(x_N)_j$

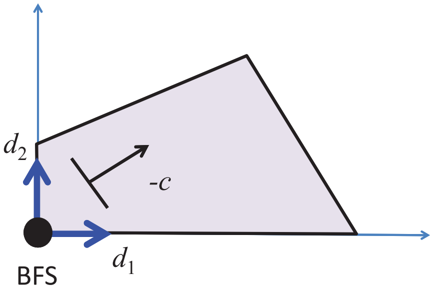
$\tilde{c}_N := (c_N^T - c_B^T B^{-1} N)^T$ are **reduced costs** for non-basic variables

- If $(\tilde{c}_N)_j \geq 0$, d_j does not decrease objective value.
- If $(\tilde{c}_N)_j < 0$, consider update $\bar{x} + \theta d_j$ with θ as large as possible since objective value change is $\theta \cdot (\tilde{c}_N)_j < 0$ as long as $\theta > 0$.

Question: What is the maximum value of θ that we can choose?

Reduced costs, in picture

To see which directions d_j are profitable, we form reduced costs $(\tilde{c}_N)_j$ which are inner products of cost vector c and directions d_j.



Question: Iterate moves along d_j with $(\tilde{c}_N)_j < 0$, but how far?

Update along search direction

At BFS $\bar{x} = (x_B^T, \mathbf{0})^T$, negative reduced cost for $(x_N)_j$ (i.e., $(\tilde{c}_N)_j < 0$).

- Iterate update
  $$\bar{x} + \theta d_j = \begin{pmatrix} x_B \\ \mathbf{0} \end{pmatrix} + \theta \begin{pmatrix} d_B \\ d_N \end{pmatrix} = \begin{pmatrix} x_B - \theta B^{-1} N_j \\ \theta e_j \end{pmatrix}, \ \theta \geq 0$$
- If $B^{-1}N_j \leq 0$ then $\bar{x} + \theta d_j \geq 0$ for all $\theta \geq 0$. Let $\theta \to \infty$, and we conclude that objective value is **unbounded from below**.
- If $B^{-1}N_j \nleq 0$ some entry of $x_B - \theta B^{-1}N_j$ becomes 0 as θ increases. Thus, $\theta \leq \theta^* = \min_{k:(B^{-1}N_j)_k > 0} \frac{(x_B)_k}{(B^{-1}N_j)_k}$, and let i be s.t. $\theta^* = \frac{(x_B)_i}{(B^{-1}N_j)_i}$.
- Thus, we arrive at new iterate $\bar{x} + \theta^* d_j$ with $(x_B - \theta^* B^{-1}N_j)_i = 0$. Note: θ^* can be zero if $\bar{x}$ is degenerate!

BFS $\bar{x} = (1,1,0,0)$, $B = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}$, $N = \begin{pmatrix} 1 & 1 \\ 3 & 4 \end{pmatrix}$, $c_B = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$, $c_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Reduced costs:

$$\tilde{c}_N^T = (c_N^T - c_B^T B^{-1} N) = -\begin{pmatrix} 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 \\ 3 & 4 \end{pmatrix} = (-3, -4)$$

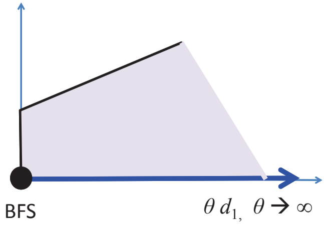
Both dir's $d_1 = (-3/2, 1/2, 1, 0)$ and $d_2 = (-2, 1, 0, 1)$ are profitable.

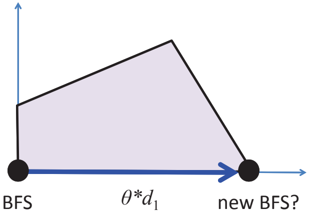
For d_1, $B^{-1}N_1 = \begin{pmatrix} 3/2 \\ -1/2 \end{pmatrix}$, max step-size $\theta^* = \frac{(x_B)_1}{(B^{-1}N_1)_1} = \frac{1}{3/2} = 2/3$.

Updated iterate: $\bar{x} + \theta^* d_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + (2/3) \begin{pmatrix} -3/2 \\ 1/2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 4/3 \\ 2/3 \\ 0 \end{pmatrix}$

Iterate update, in picture

Updating $\bar{x} + \theta d_j$ either tells us objective value is unbounded (left picture), or a possibly new point $\bar{x} + \theta^* d_j$ is reached (right picture).





Question: What is $\bar{x} + \theta^* d_j$? Is it a BFS? How is it related to $\bar{x}$?

Change of basis

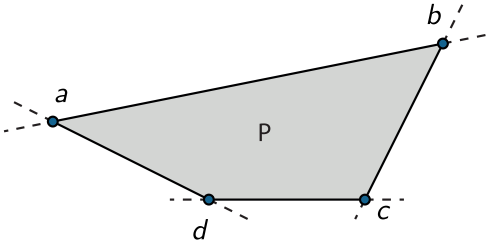
From $\bar{x}$ to $\bar{x} + \theta^* d_j$, the i-th basic variable $(x_B)_i$ becomes zero, whereas j-th non-basic variable $(x_N)_j$ (i.e., the $(m+j)$-th variable) becomes θ^*:

$$\bar{x} = \begin{pmatrix} \vdots \\ (x_B)_i \\ \vdots \\ \mathbf{0} \end{pmatrix} \qquad \bar{x} + \theta^* d_j = \begin{pmatrix} \vdots \\ 0 \\ \vdots \\ \theta^* e_j \end{pmatrix}$$

- Can show the columns $a_1, \ldots, a_{i-1}, a_{m+j}, a_{i+1}, \ldots, a_m$ are linearly independent (forming a new basis), and $\bar{x} + \theta^* d_j$ is indeed a BFS.
- We say $(x_B)_i$ **leaves the basis** to become non-basic variable, whereas $(x_N)_j$ **enters the basis** to become basic variable.
- $\bar{x}$ and $\bar{x} + \theta^* d_j$ are in fact **adjacent** BFS's (see text for formal definition of adjacency).

Adjacent BFS's

In the figure, a is adjacent to b and d, and b is adjacent to a and c



Search direction d_j is along the edge of polyhedron, from a BFS to a BFS.

Adjacency

Theorem
Two BFS's are adjacent if and only if their sets of basic variables differs in exactly **one** place.

Remember, $(x_B)_i$ leaves the basis whereas $(x_N)_j$ enters the basis during the change from $\bar{x}$ to $\bar{x} + \theta^* d_j$.

- In principle, the simplex algorithm starts with a BFS corresponding to a matrix B and then iteratively changes one of the columns in B until optimality occurs.
- Basis matrix B is modified at every iteration, but BFS might stay because of degeneracy (a degenerate BFS can be associated with more than one basis)!

To recap so far

We have seen so far...

- At a BFS with $A = (B, N)$, compute search directions $d_j = \begin{pmatrix} -B^{-1}N_j \\ e_j \end{pmatrix}$, $j = 1, \ldots, n - m$.
- Evaluate the reduced costs $\tilde{c}_N := (c_N^T - c_B^T B^{-1} N)^T$ to see which directions are profitable (which non-basic variable to enter basis).
- What happens when we update $\bar{x} + \theta d_j$ for some d_j with $(\tilde{c}_N)_j < 0$ – either objective value is unbounded or an adjacent BFS is reached.

But...

- What if $\tilde{c}_N \geq \mathbf{0}$, as all our search directions are not profitable?

Optimality criterion

If at a BFS with basis B, all reduced costs are nonnegative ($\tilde{c}_N \geq \mathbf{0}^{n-m}$), then it is optimal and the simplex algorithm terminates:

> **Theorem**
> Let $\bar{x}$ be a BFS associated with basis matrix B, and let $\tilde{c}_N = (c_N^T - c_B^T B^{-1} N)^T$ be the corresponding vector of reduced costs for the non-basic variables. If $\tilde{c}_N \geq 0$, then $\bar{x}$ is optimal.

Proof: All feasible directions d at $\bar{x} = (x_B^T, x_N^T)^T$ are of the form

$$d = \begin{pmatrix} -B^{-1}N \, d_N \\ d_N \end{pmatrix} \implies c^T d = \underbrace{(c_N^T - c_B^T B^{-1} N)}_{\tilde{c}_N^T} d_N$$

$\tilde{c}_N \geq 0$ and $d_N \geq 0$ (i.e., feasible direction) concludes the statement.

The conceptual simplex algorithm

1. Assume we have an initial BFS $\bar{x} = (x_B^T, x_N^T)^T$ with $A = (B, N)$.
2. Compute reduced costs $\tilde{c}_N = (c_N^T - c_B^T B^{-1} N)^T$.
   - If $\tilde{c}_N \geq \mathbf{0}$, then current BFS is optimal, terminate.
   - If $\tilde{c}_N \ngeq \mathbf{0}$, choose some non-basic var index j s.t. $(\tilde{c}_N)_j < 0$.
3. Compute $B^{-1} N_j$.
   - If $B^{-1} N_j \leq 0$, then objective value is $-\infty$, terminate.
   - If $B^{-1} N_j \nleq 0$, compute $\theta^* = \min_{k:(B^{-1}N_j)_k > 0} \frac{(x_B)_k}{(B^{-1}N_j)_k}$.
4. Update $\bar{x} \leftarrow \bar{x} + \theta^* d_j$. Let i be s.t. $\theta^* = \frac{(x_B)_i}{(B^{-1}N_j)_i}$. Update basis

$$B \leftarrow \begin{bmatrix} | & & | & | & | & & | \\ a_1 & \cdots & a_{i-1} & a_{m+j} & a_{i+1} & \cdots & a_m \\ | & & | & | & | & & | \end{bmatrix}.$$

Reorder variables s.t. the first m variables are basic in $\bar{x} + \theta^* d_j$.

Finite termination

Q: Will the simplex algorithm terminate in finite number of steps?
A: Yes, if all BFS's are non-degenerate.

Theorem

If feasible set is nonempty and every BFS is non-degenerate, then the simplex algorithm terminates in finite number of iterations. At termination, two possibilities are allowed:
(a) an optimal basis B found with the associated optimal BFS.
(b) a direction d found s.t. $Ad = 0$, $d \geq 0$ and $c^T d < 0$, thus optimal objective value is $-\infty$.

The simplex algorithm can be modified to finitely terminate even with degenerate BFS (e.g., Bland's rule, see text)!

Computational complexity

- The simplex algorithm works very well in practice.
- The simplex algorithm can, in the worst case, visits all $\binom{n}{m}$ BFS's before termination – worst-case computation effort is exponential.
- Polynomial-time algorithms are available (e.g., ellipsoid algorithm, interior point algorithms). See coming lectures.

We create a so called **Phase-I problem** by introducing **artificial variables** a_i in every row.

$$\begin{aligned} w^* = \text{minimize } \; w &= (\mathbf{1}^m)^T a, \\ \text{subject to} \quad Ax + I^m a &= b, \\ x \qquad &\geq \mathbf{0}^n, \\ a &\geq \mathbf{0}^m. \end{aligned}$$

- Why is this easier? Because $a = b$, $x = \mathbf{0}^n$ is an initial BFS.
- $w^* = 0 \implies$ Optimal solution $a^* = \mathbf{0}^m$
  x^* BFS in the original problem
- $w^* > 0 \implies$ There is no BFS to the original problem
  The original problem is infeasible