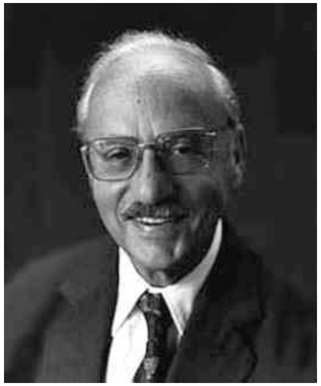
C&O 355
Mathematical Programming
Fall 2010
Lecture 20

N. Harvey

The “Simplex Method”

- “The obvious idea of moving along edges from one vertex of a convex polygon to the next” [Dantzig, 1963]



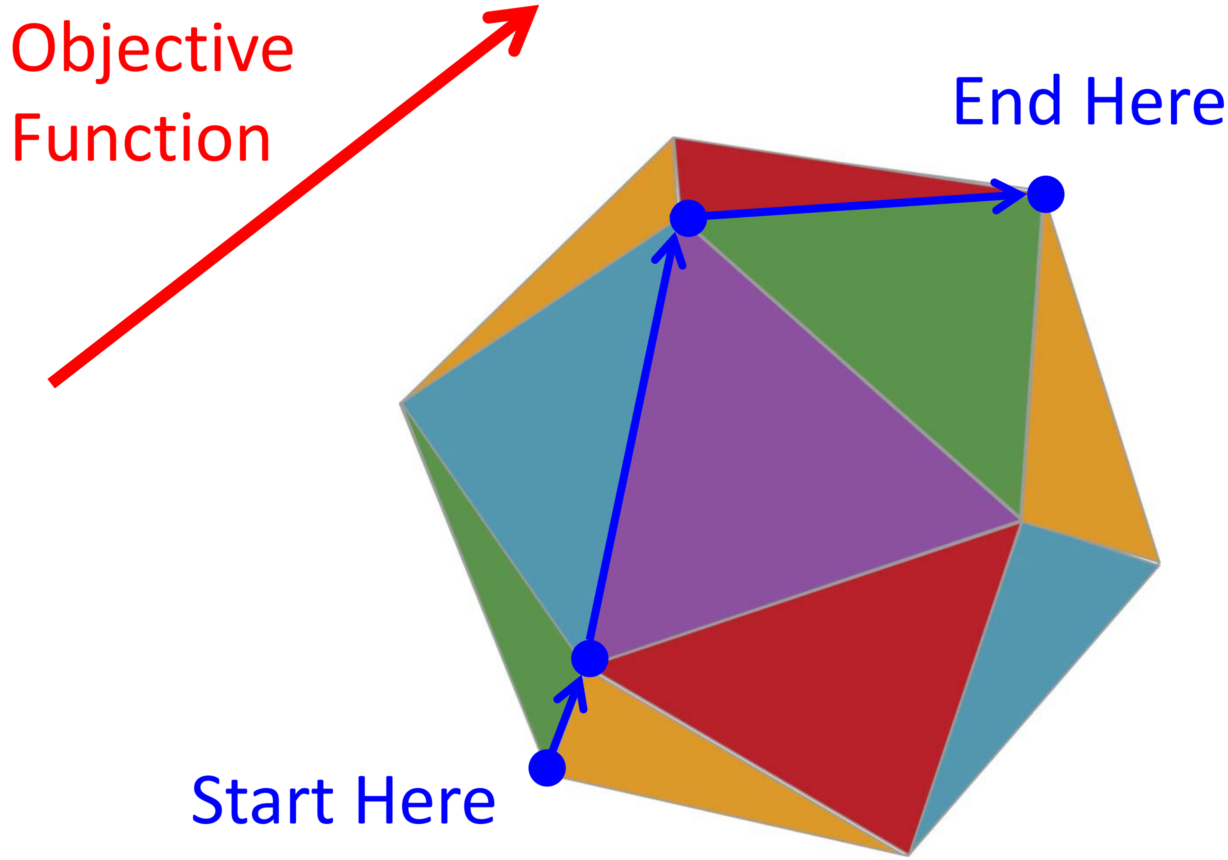
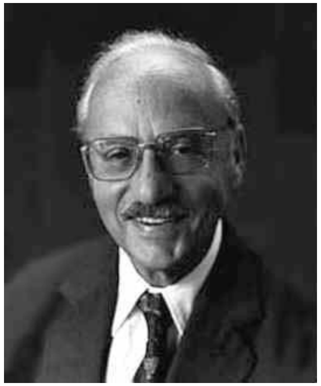


Image: http://torantula.blogspot.com/

The “Simplex Method”

- “The obvious idea of moving along edges from one vertex of a convex polygon to the next” [Dantzig, 1963]



Polyhedron:

$$P = \{\, x \,:\, Ax \leq b \,\}$$

LP:

$$\begin{aligned} &\max && c^{\mathsf{T}} x \\ &\text{s.t.} && x \in P \end{aligned}$$

Algorithm

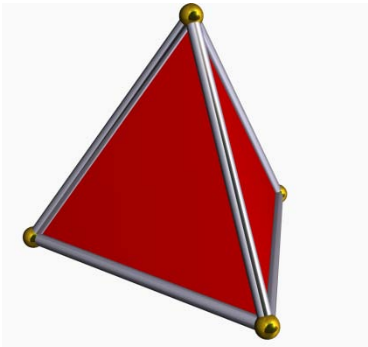
Let x be any vertex of P

For each neighbor y of x

  If $c^T y > c^T x$ then

    Set x=y and go to start

Halt



This is a simplex

- **Remarks**
  - The name sounds fancy, but is meaningless.
  - In practice, very fast. Used in all LP software.
  - In theory, we don’t know whether it’s fast or not.
    (Because we don’t understand the diameter of polyhedra, i.e., Hirsch Conjecture)

Pitfalls

- The simplex method is very simple…

Polyhedron:

$$P = \{\, x \,:\, Ax \leq b \,\}$$

LP:

$$\begin{aligned} \max \quad & c^{\mathsf{T}} x \\ \text{s.t.} \quad & x \in P \end{aligned}$$

Algorithm

Let x be any vertex of P
For each neighbor y of x
    If $c^T y > c^T x$ then
        Set x=y and go to start
Halt

…if we can handle a few issues

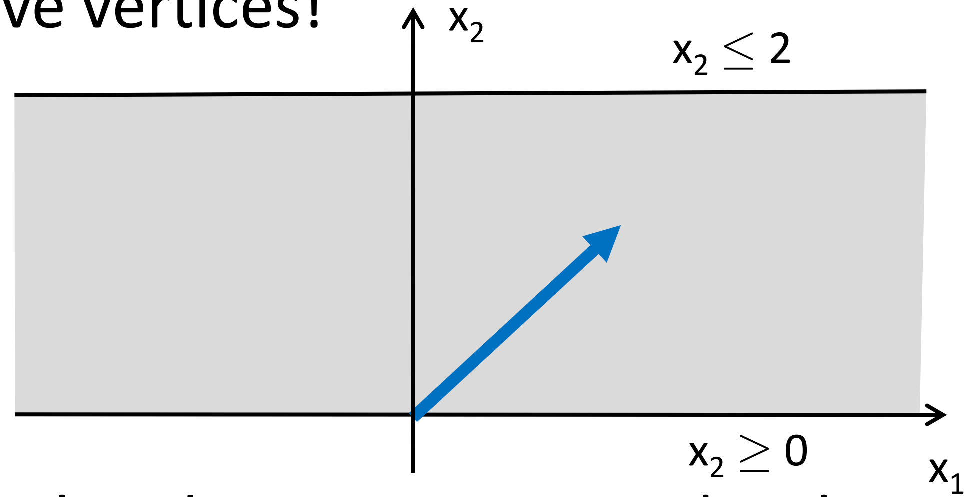
1. What if there are no vertices?
2. How can I find a starting vertex?
3. What are the “neighboring” vertices?
4. Does the algorithm terminate?
5. Does it produce the right answer?

Issue #1

What if there are no vertices?

- Not all polyhedrons have vertices!

$$\begin{aligned} \max \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_2 \leq 2 \\ & x_2 \geq 0 \end{aligned}$$



- **Recall:** Any polyhedron that does not contain a line has at least one vertex.
- **A fix:** Instead of $\max \{ c^T x : Ax \leq b \}$
  we could solve $\max \{ c^T(u-v) : A(u-v)+w=b,\ u,v,w \geq 0 \}$.
  These LPs are equivalent.
  The feasible region of the new LP contains no line.
- **Summary:** Can assume we're solving an LP with a vertex.

Pitfalls

- The simplex method is very simple…

  …if we can handle a few issues

1. What if there are no vertices?

   Can modify polyhedron so that it has a vertex.

2. How can I find a starting vertex?
3. What are the “neighboring” vertices?
4. Does the algorithm terminate?
5. Does it produce the right answer?

Issue #2

How can I find a starting vertex?

- This is non-trivial! As shown in Lecture 3, maximizing the LP is equivalent to finding a feasible point for:

$$Ax \leq b \qquad A^{\mathsf{T}}y = c \qquad y \geq 0 \qquad c^{\mathsf{T}}x \geq b^{\mathsf{T}}y$$

  So, in general, finding a feasible point is not easy.

- **A fix:**

The problem "find a feasible point for my LP" can be solved by a new LP. How does this help?!?

The new LP has an obvious feasible point!
So we solve the new LP, get feasible point for old LP.

- Once you have a feasible point, it's easy to find a vertex:
  - **Lecture 10:** Any LP whose feasible region contains no line has an optimal solution at a vertex.
  - That proof actually gives an **algorithm** to find a vertex.

Finding a starting point

- Consider LP max { $c^T x$: $x \in P$ } where P={ x : Ax=b, $x \geq 0$ }
- We'll find a feasible point by solving a new LP!
  - Note: c is irrelevant. We can introduce a new objective function
  - WLOG, $b \geq 0$ (Can multiply constraints by -1)
  - Allow "Ax=b" constraint to be violated via "artificial variables":

    $$Q = \{ (x,y) : Ax+y=b,\ x \geq 0,\ y \geq 0 \}$$

  - Note: $(x,0) \in Q \Leftrightarrow x \in P$. Can we find such a point?
  - Solve the new LP min { $\Sigma_i y_i$: $(x,y) \in Q$ }
  - If the optimal value is 0, then $x \in P$. If not, P is empty!
  - How do we find feasible point for the new LP?
    - $(x,y)=(0,b)$ is a trivial solution!

Pitfalls

- The simplex method is very simple…
  …if we can handle a few issues

1. What if there are no vertices?
   Can modify polyhedron so that it has a vertex.
2. How can I find a starting vertex?
   Can find a feasible point by solving a different LP.
   Can move from that feasible point towards a vertex.
3. What are the "neighboring" vertices?
4. Does the algorithm terminate?
5. Does it produce the right answer?

Issue #4

Does the algorithm terminate?

- This is easy!
  - In every iteration, the algorithm sets x to a new vertex.
  - Note that the objective function **strictly improves** by moving to the new vertex.
  - So the algorithm cannot have visited that vertex before.
  - **Recall from Lecture 10:**
    Every polyhedron has only finitely many vertices.
  - So the algorithm must terminate after finitely many steps.

Pitfalls

- The simplex method is very simple…

  …if we can handle a few issues

1. What if there are no vertices?

   Can modify polyhedron so that it has a vertex.

2. How can I find a starting vertex?

   Can find a feasible point by solving a different LP.

   Can move from that feasible point towards a vertex.

3. What are the “neighboring” vertices?

4. Does the algorithm terminate?

   Yes: objective function increases only finitely many times.

5. Does it produce the right answer?

Edges

From Lecture 11:

Fact 1.3. Let $P = \{ \mathbf{x} : \mathbf{a}_i^\mathsf{T} \mathbf{x} \leq b_i \ \forall i \}$ be a polyhedron in $\mathbb{R}^n$. Let $\mathbf{x}$ and $\mathbf{y}$ be two distinct vertices. Recall our notation $\mathcal{I}_\mathbf{x} = \{ i : \mathbf{a}_i^\mathsf{T} \mathbf{x} = b_i \}$. Suppose rank $\{ \mathbf{a}_i : i \in \mathcal{I}_\mathbf{x} \cap \mathcal{I}_\mathbf{y} \} = n - 1$. Then the line segment

$$L_{\mathbf{x},\mathbf{y}} = \{ \lambda \mathbf{x} + (1 - \lambda) \mathbf{y} : \lambda \in [0, 1] \} \qquad (1.2)$$

is an edge of P. Moreover, every bounded edge arises in this way.

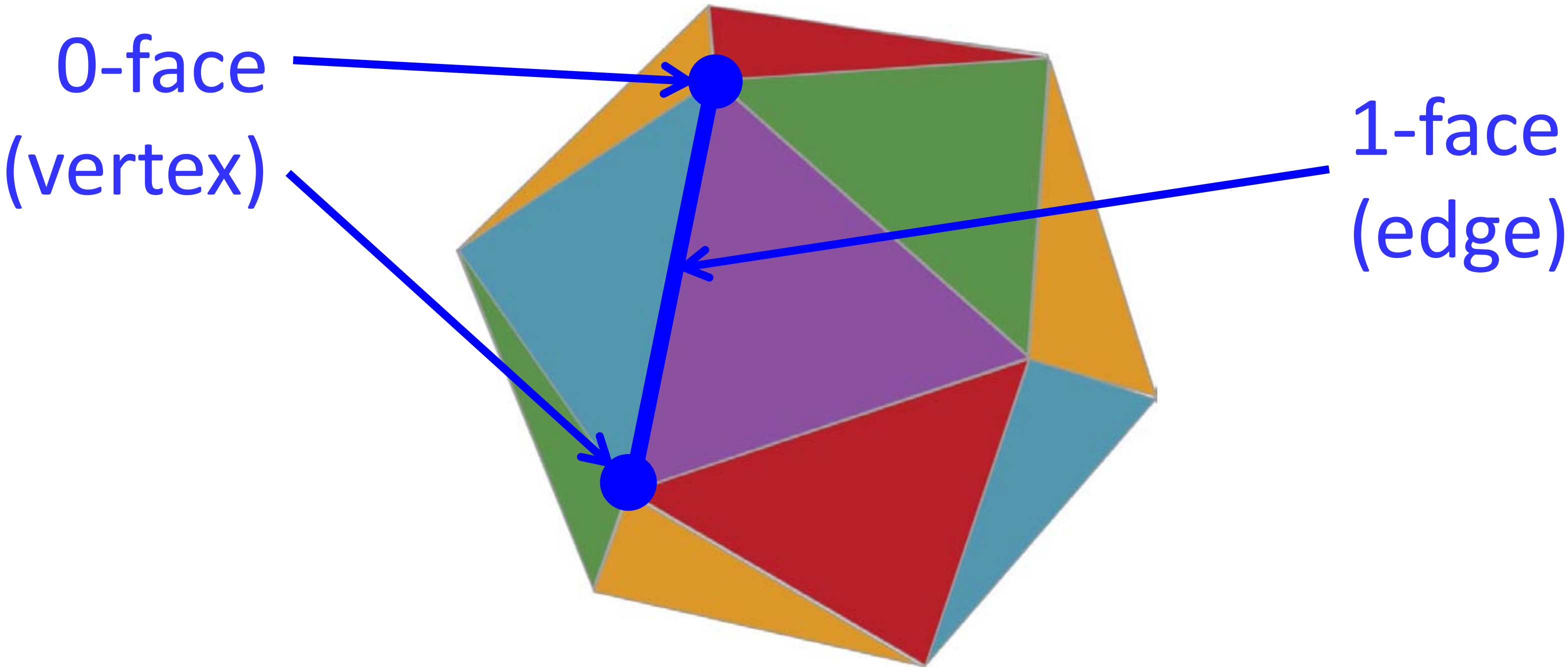


Image: http://torantula.blogspot.com/

- **Summary:** Two vertices are neigboring if the constraints that are tight at **both** vertices have rank n-1.

Issue #3

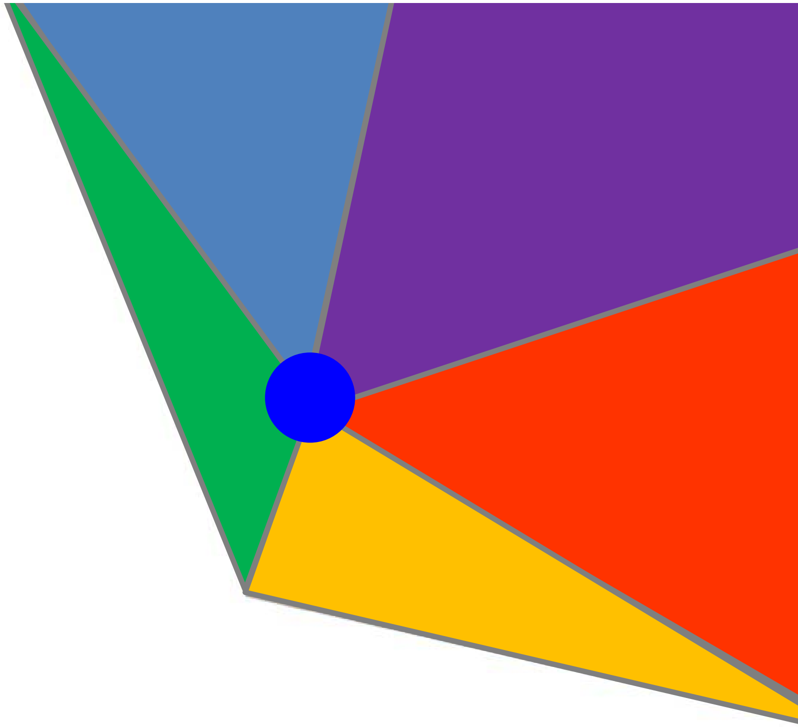
What are the "neighboring" vertices?

- Consider a vertex x.
  It is also a BFS, so the tight constraints at x have rank n.
- Choose a subset of these constraints of rank n-1.
  Consider the set of points for which this subset of constraints are all tight. This is an edge. (By Asst 3, Question 3)
- Uh oh! If there are t tight constraints at x, then the number of such subsets could be $\binom{t}{n-1}$. Enumerating all of these subsets could be very slow.
- **A fix:**
  - Add very small "noise" to every entry of the matrix A defining the constraints.
  - Then every vertex has **exactly** n tight constraints, and at most n edges leaving it.

Issue #3

What are the “neighboring” vertices?

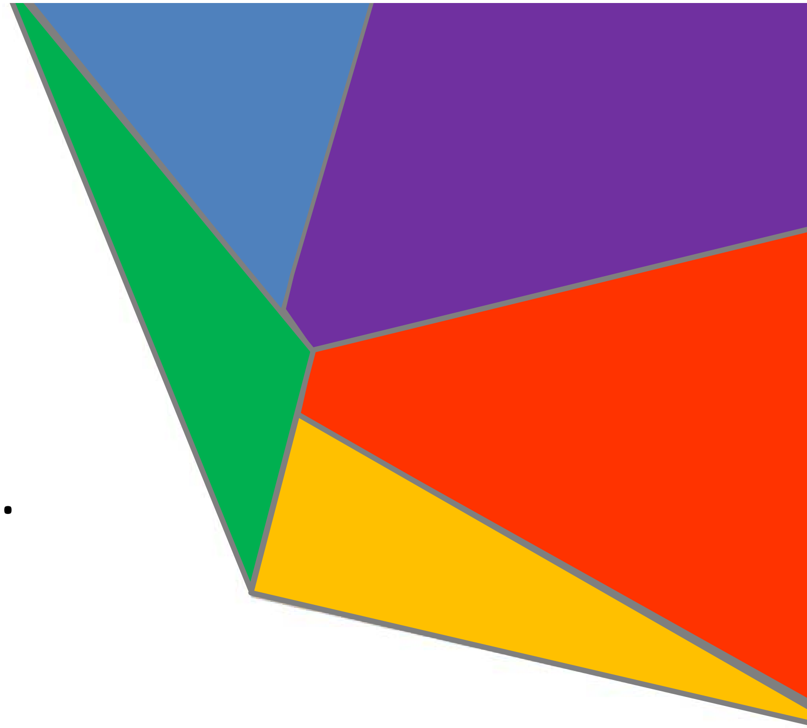
- **A fix:**
  - Add very small “noise” to every entry of the matrix A defining the constraints.
  - Then every vertex has **exactly** n tight constraints, and at most n edges leaving it.
- **Example:**
  - Want only 3 edges leaving x, but there are 5.



Issue #3

What are the “neighboring” vertices?

- **A fix:**
  - Add very small “noise” to every entry of the matrix A defining the constraints.
  - Then every vertex has **exactly** n tight constraints, and at most n edges leaving it.
- **Example:**
  - Want only 3 edges leaving x, but there are 5.
  - If we perturb the constraints slightly, every vertex has only 3 tight constraints and 3 edges.



Issue #3

What are the "neighboring" vertices?

- **A Fix:**
  - Add very small "noise" to every entry of the matrix A defining the constraints.
  - Then every vertex has **exactly** n tight constraints, and at most n edges leaving it.
- **Finding the neighbors:**
  For each edge leaving the vertex
  - Move along edge while remaining in feasible region
  - When a new constraint becomes tight, we've arrived at a neighboring vertex
  - If no constraint becomes tight, it's an unbounded edge
    - Check if the objective function increases when moving along the edge. If so, LP is unbounded.

Pitfalls

- The simplex method is very simple…

  …if we can handle a few issues

1. What if there are no vertices?
   Can modify polyhedron so that it has a vertex.
2. How can I find a starting vertex?
   Can find a feasible point by solving a different LP.
   Can move from that feasible point towards a vertex.
3. What are the “neighboring” vertices?
   Add noise to constraints so that only each vertex has few edges. Find edges by choosing n-1 tight constraints.
4. Does the algorithm terminate?
   Yes: objective function increases only finitely many times.
5. Does it produce the right answer?

Issue #5

Does algorithm produce the right answer?

- **Yes!**
  If you cannot increase the objective function by moving along any edge leaving x, then x must be optimal.
- That is very intuitive, but formalizing it takes some work: See Notes for Lecture 20.

Pitfalls

- The simplex method is very simple…

  …if we can handle a few issues

1. What if there are no vertices?
   Can modify polyhedron so that it has a vertex.
2. How can I find a starting vertex?
   Can find a feasible point by solving a different LP.
   Can move from that feasible point towards a vertex.
3. What are the “neighboring” vertices?
   Add noise to constraints so that only each vertex has few edges. Find edges by choosing n-1 tight constraints.
4. Does the algorithm terminate?
   Yes: objective function increases only finitely many times.
5. Does it produce the right answer?
   Yes: if no edge increases objective function, x is optimal.

Summary

- "The obvious idea of moving along edges from one vertex of a convex polygon to the next" [Dantzig, 1963]

Algorithm

Let x be any vertex of P
For each neighbor y of x
    If $c^T y > c^T x$ then
        Set x=y and go to start
Halt

- The idea is very simple
- There are many pitfalls which complicate things.
  - Main idea to handle complications is to modify P so that it becomes "nice" in various ways.
- Finding neighbors is conceptually simple, but to formalize it, the notation gets a bit messy.
- We used Farkas' Lemma to prove optimality.