Linear Programming: Simplex Method

Daniel Kane

Department of Computer Science and Engineering
University of California, San Diego

Advanced Algorithms and Complexity
Data Structures and Algorithms

Learning Objectives

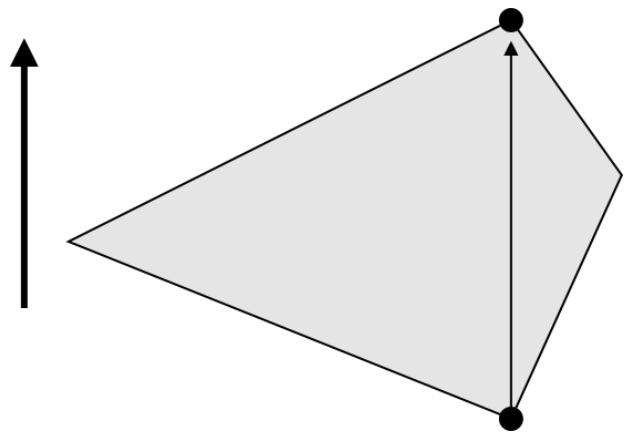
- Understand the idea of moving between vertices of a polytope.
- Implement the simplex method.

Simplex Method

- Oldest algorithm for solving linear programs.
- Still one of the most efficient.
- Not quite the runtime we would like.

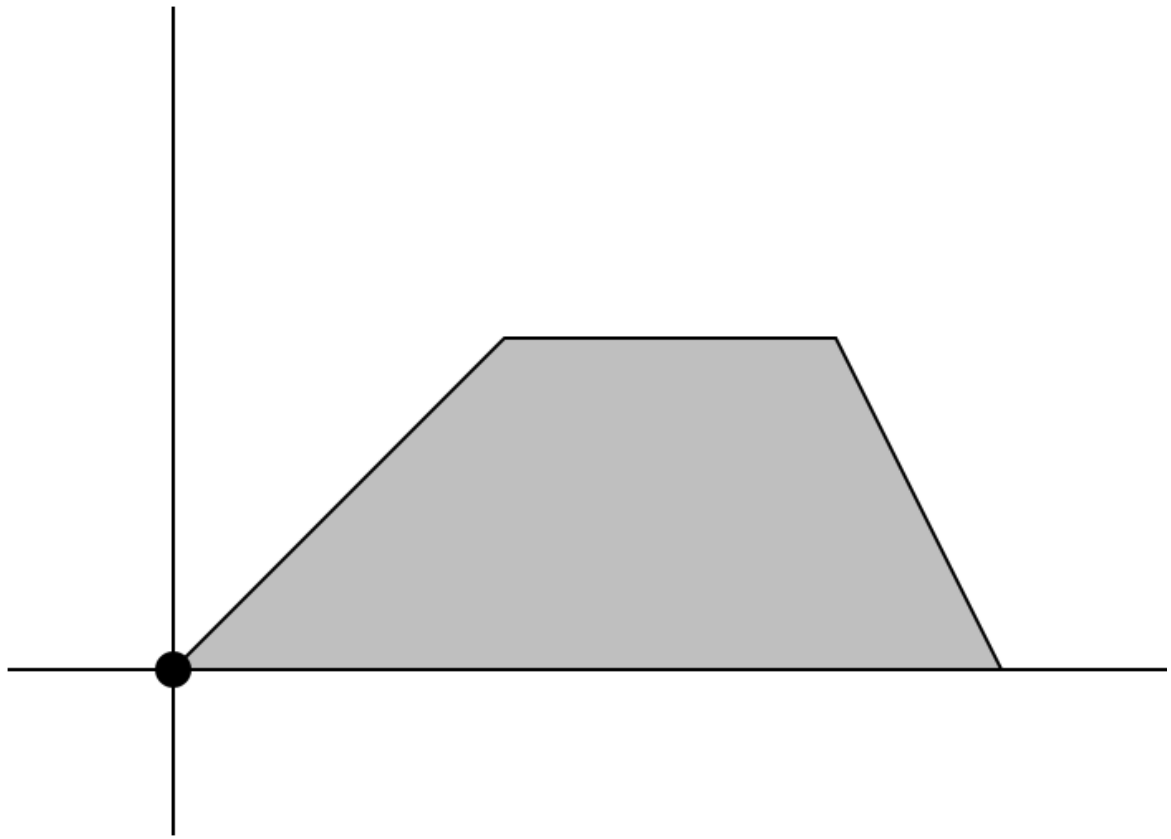
Formulation

Solves the optimization from starting point formulation.



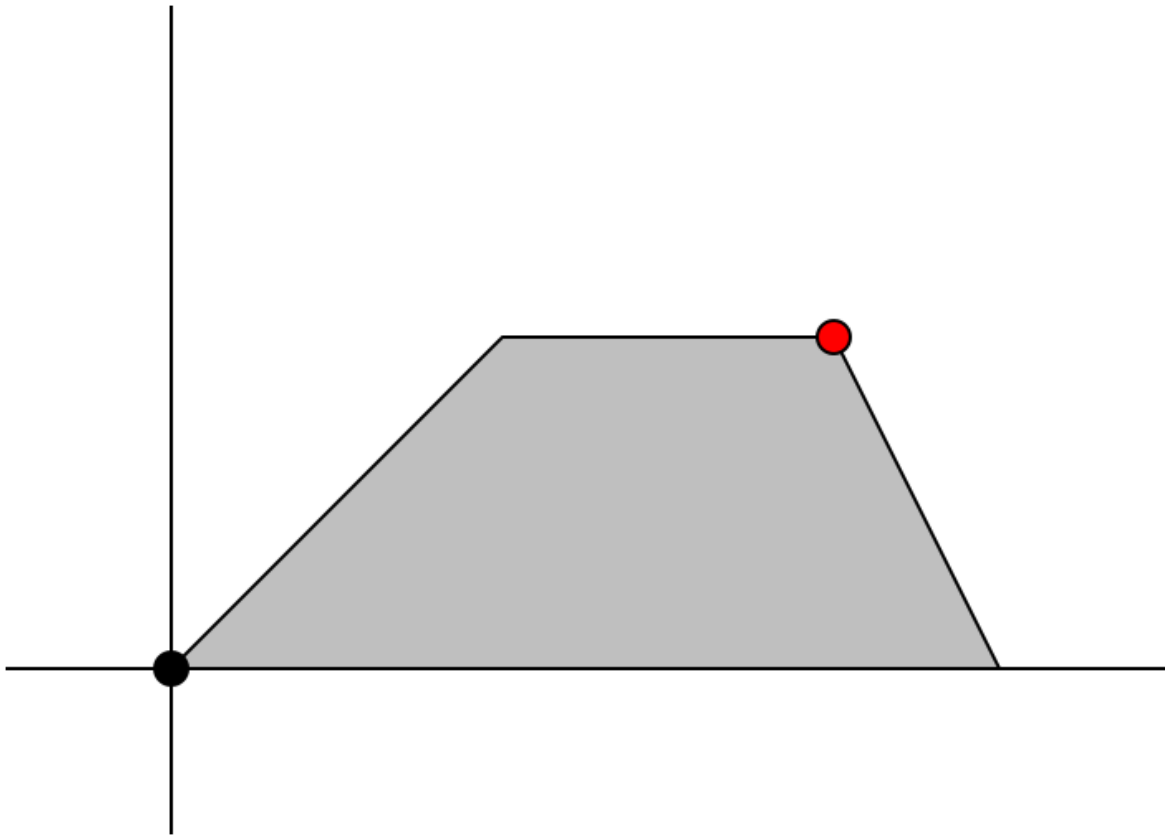
Idea

Start at vertex.



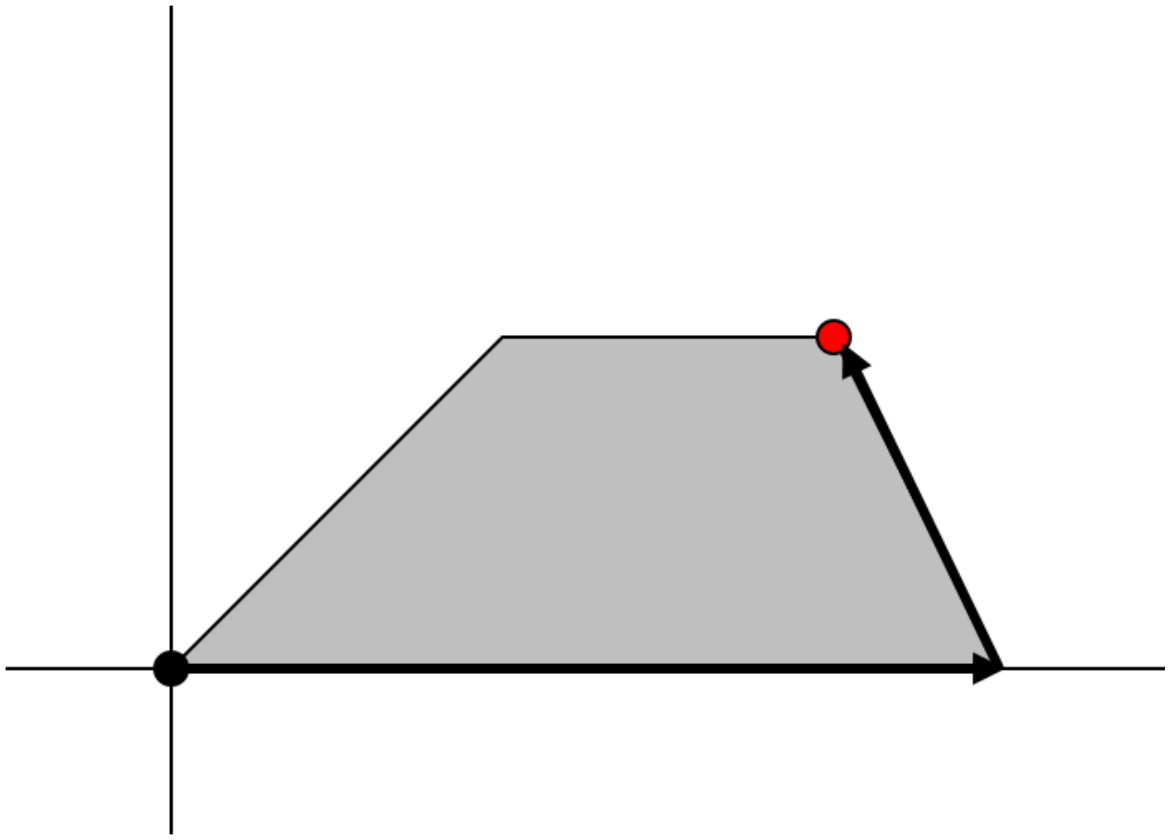
Idea

Optimum at another vertex.



Idea

Path of vertices to find optimum.



Vertices and Edges

- Vertex p when n defining equations are tight (solve with Gaussian elimination).

Vertices and Edges

- Vertex p when n defining equations are tight (solve with Gaussian elimination).
- Relax one equation to get an edge. Points of form $p + tw$, $t \geq 0$.
- Edge continues until it violates some other constraint.

Vertices and Edges

- Vertex p when n defining equations are tight (solve with Gaussian elimination).
- Relax one equation to get an edge. Points of form $p + tw$, $t \geq 0$.
- Edge continues until it violates some other constraint.
- If $v \cdot w > 0$, can follow edge to find larger value of objective.

Pseudocode

Simplex

```
Start at vertex p
repeat
  for each equation through p
    relax equation to get edge
    if edge improves objective:
      replace p by other end
      break
  if no improvement:  return p
```

OtherEndOfEdge

Vertex p defined by n equations
Relax one, write general solution as $p + tw$ (Gaussian elimination)
Relaxed inequality requires $t \geq 0$
For each other inequality in system:
  Largest t so $p + tw$ satisfies
Let t_0 be smallest such t
return $p + t_0 w$.

Correctness

Theorem

If p is a vertex that is not optimal, there is some adjacent vertex that does better.

Proof (sketch)

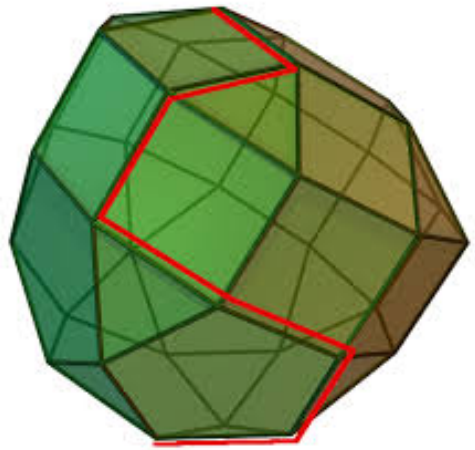
- p at intersection of equations $E_1, \ldots, E_n$.
- Can always write $x \cdot v \leq \ldots$ as linear combination.
- If positive coefficients, p is an optimum.
- Otherwise, relax equation with negative coefficient.

Analysis

How long does simplex take?

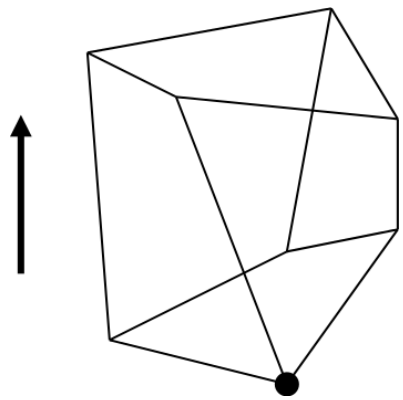
Analysis

How long does simplex take? Must follow path to optimum.



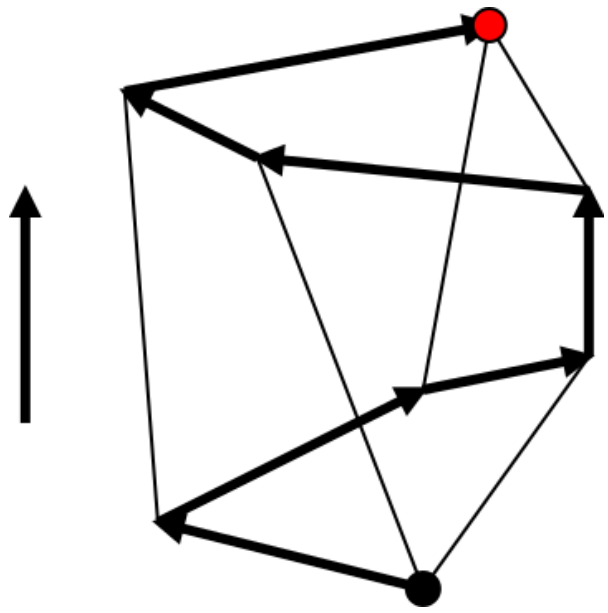
Problem

What is the largest number of steps that the simplex method might require to find the optimum in the following situation? Assume that points drawn higher on the screen have larger values of the objective.



Solution

As many as 7 steps (though potentially as few as 3).



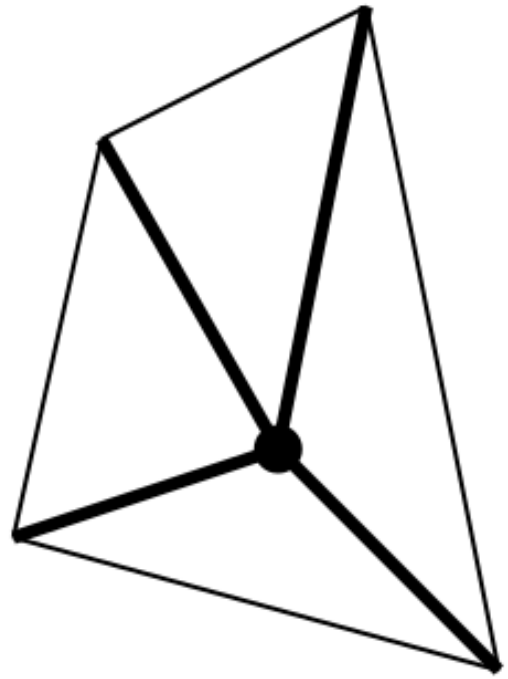
Runtime

- Runtime proportional to path length.
- Path length usually pretty reasonable.
- However, there are examples where path is exponential!

Degeneracy

One technical problem:

If more than n hyperplanes intersect at a vertex, don't know which to relax.



Fix

Tweak equations a tiny bit to avoid these intersections.

Fix

Tweak equations a tiny bit to avoid these intersections.

You can actually make this work nicely with infinitesimal changes.

Fix

Tweak equations a tiny bit to avoid these intersections.

You can actually make this work nicely with infinitesimal changes.

Number constraints. Strengthen first by ε, next by ε^2, etc.

Fix

- Don't need to actually change equations.
- At degenerate point, keep track of which n you are "really" on.
- When travelling along an edge to a degenerate corner, add the lowest-numbered constraint at the new corner.
- Edges from degenerate corner to itself: change "corner" if edge "improves" objective.

Summary

- Solve LP by moving between adjacent vertices towards optimum.
- Works well in practice.
- Potentially exponential time.