Chapter 1
Linear Programming

Paragraph 6
LPs in Polynomial Time

What we did so far

- We developed a standard form in which all linear programs can be formulated.
- We developed a group of algorithms that solves LPs in that standard form.
- While we could guarantee termination, and the “average” runtime is quite good, the worst-case runtime of Simplex and its variants may be exponential.
- We shall now look into other algorithms for solving LPs in polynomial time – guaranteed!

The Ellipsoid Algorithm

- Whether or not LP was in P was a long outstanding question.
- Only in 1979, Soviet mathematician Khachian proved that an algorithm for non-linear convex minimization named Ellipsoid Method could actually solve LPs in polynomial time.
- The method has important theoretical implications. However, the performance is so bad that its practical importance is immaterial.

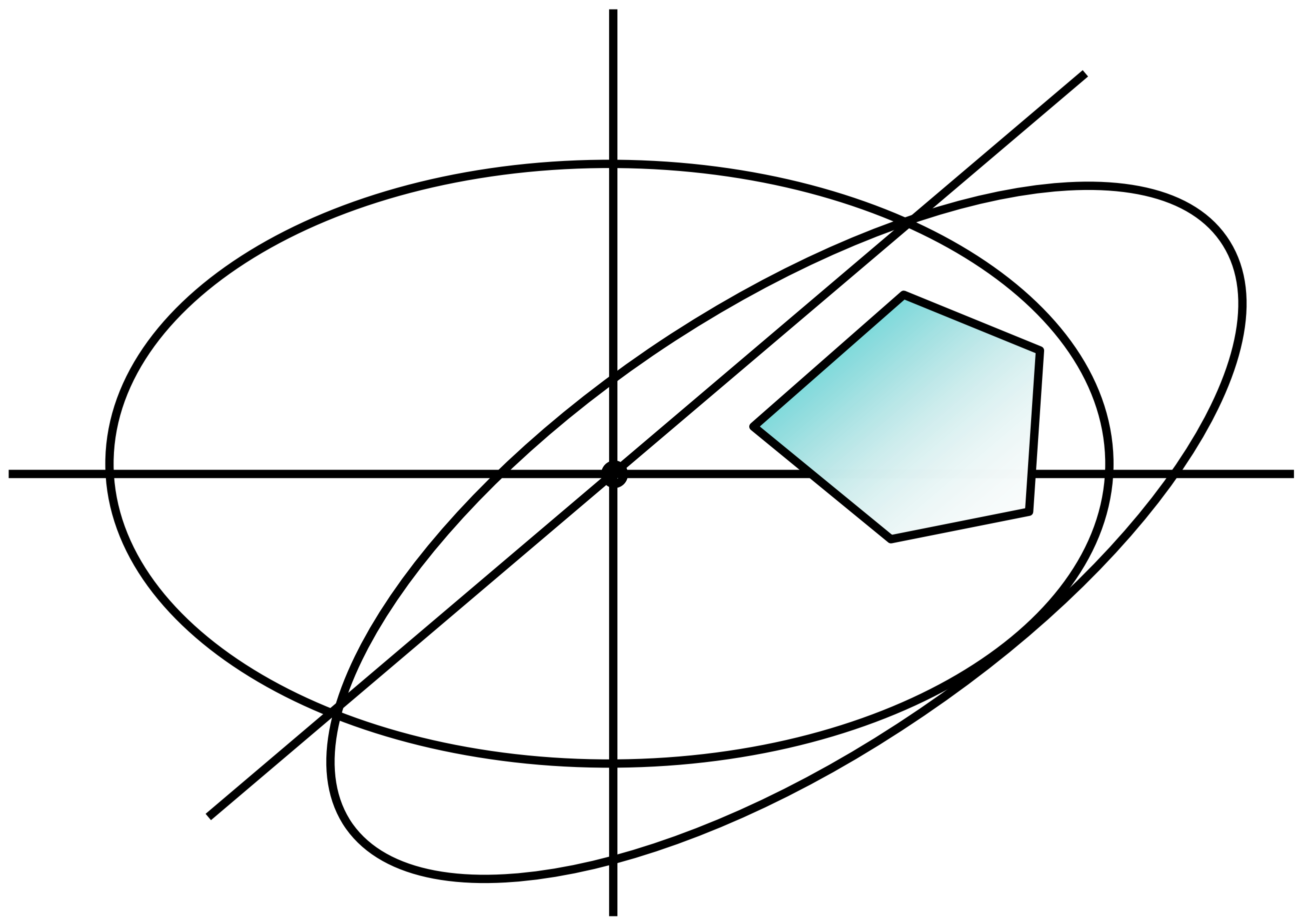
The Ellipsoid Algorithm

- It can be shown that Linear Programming is polynomially equivalent to finding a solution to a system of strict linear inequalities (LSI): $Ax < b$.
- It can further be shown:
  - If an LSI is solvable, then so is the bounded system
    - $Ax < b$
    - $-2^D < x_i < 2^D$ where D is the binary size of the LSI.
  - If an LSI has a solution, then $\{x \mid Ax \leq b\}$ must have a minimal volume of $2^{-(n+1)D}$.

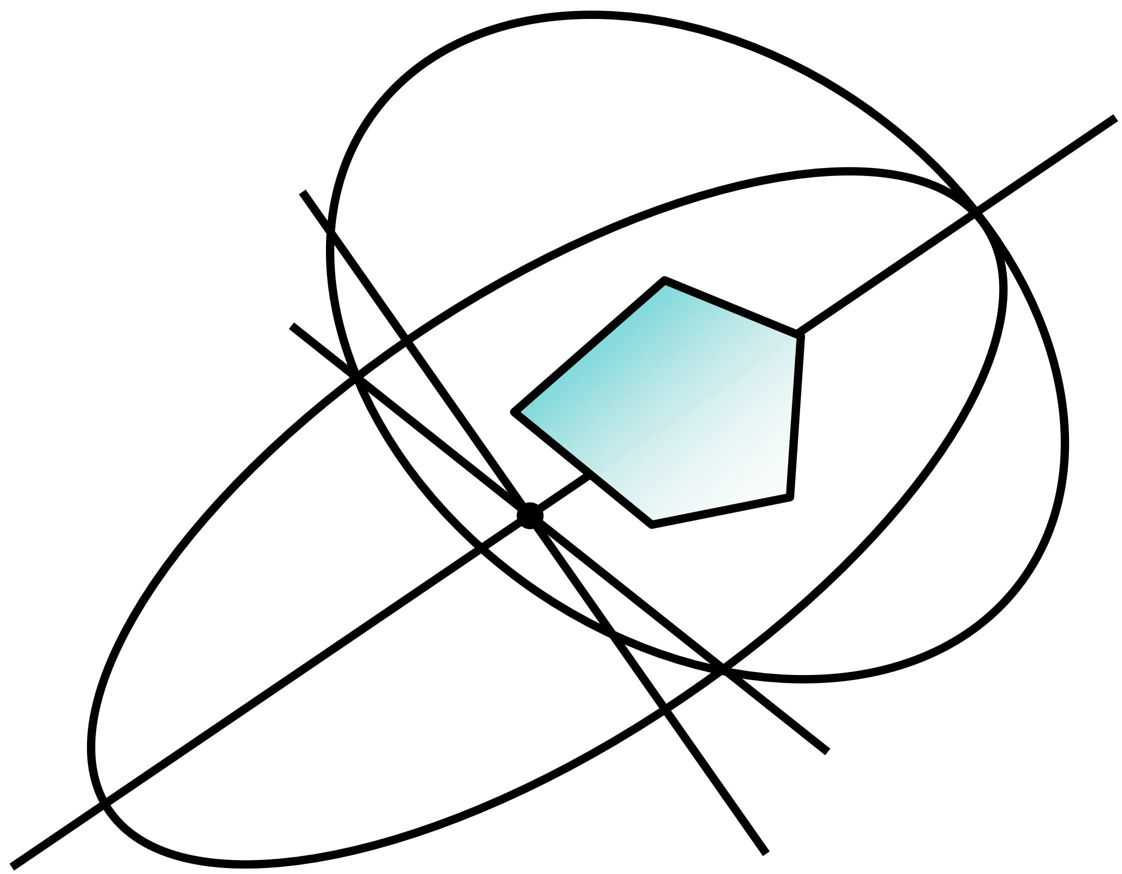
The Ellipsoid Algorithm

- The Algorithm works as follows:
  1. Find an ellipsoid that is guaranteed to contain all solutions to the system.
  2. If the center of the ellipsoid is feasible:
     return success!
  3. If the volume of the ellipsoid is too small:
     return not solvable!
  4. Using a violated constraint, slice the ellipsoid in half so that one side must contain all solutions.
  5. Construct a new ellipsoid that covers the solution containing half-ellipsoid and go back to step 2.

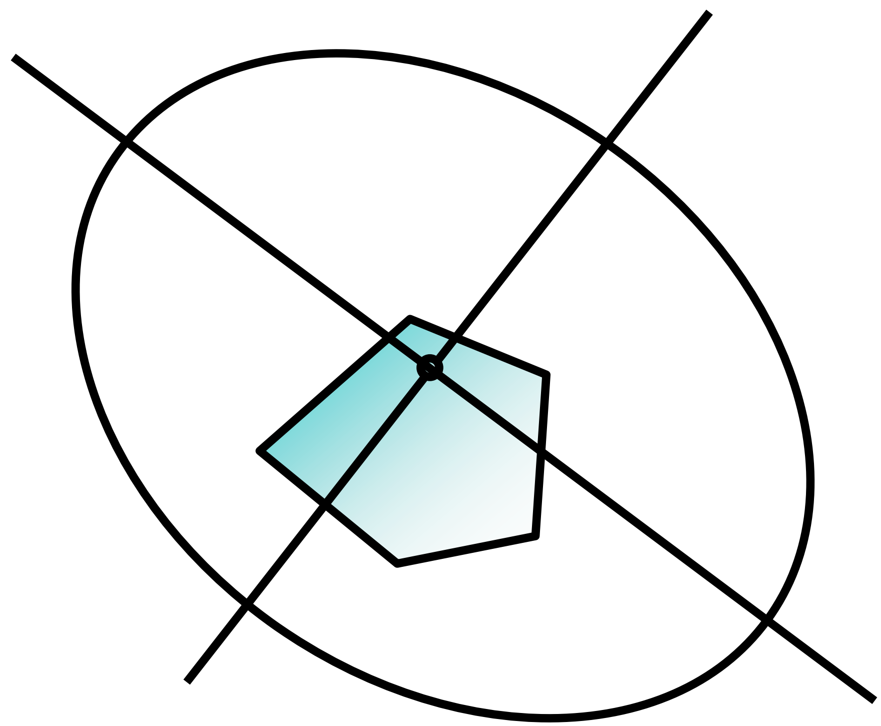
The Ellipsoid Algorithm



The Ellipsoid Algorithm



The Ellipsoid Algorithm



The Ellipsoid Algorithm

- Crucial to the polynomial runtime guarantee is the following key lemma:
  - Every half-ellipsoid is contained in an ellipsoid whose volume is less than $e^{-1/2(n+1)}$ times the volume of the original ellipsoid.
- Corollary
  - The smallest ellipsoid containing a polyhedron P has its center in P.
  - The inner loop of the ellipsoid algorithm is carried out at most a polynomial number of times.

Implications

- The two most important implications of the ellipsoid algorithm are:
  - LPs are solvable in polynomial time.
  - A linear program is polynomial time solvable even if all we can do efficiently is to provide a violated hyperplane when a suggested solution is violated.
- An algorithm that does the latter is called a separation oracle. If we can provide a violated linear constraint in polynomial time, we can even solve LPs with an exponential number of constraints!

Constraint Generation for a Lower Bound of TSP

- The Traveling Salesman Problem
  - Given a weighted graph (V,E,c), find a roundtrip that visits each node once such that the total distance is minimal.
  - We formulate this an integer program (IP):
    1. Min $\Sigma_{(i,j) \in E}\, c_{ij}\, x_{ij}$ such that
    2. $\Sigma_{j:(i,j) \in E}\, x_{ij} = 1$ for all $i \in V$
    3. $\Sigma_{i:(i,j) \in E}\, x_{ij} = 1$ for all $j \in V$
    4. $\Sigma_{i \in S,\, j \in V \setminus S}\, x_{ij} \geq 1$ for all $\emptyset \subset S \subset V$
    5. $x_{ij} \in \{0,1\}$
- To get a lower bound on the objective, we can relax (5) to $x_{ij} \geq 0$. But: The number of constraints is exponential!
- Can we find a separation oracle?

Interior Point Algorithms

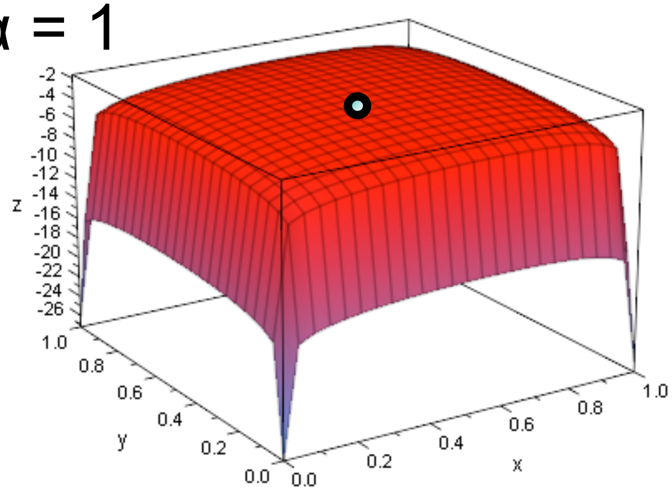
- Linear Programming is also polynomially equivalent to finding the maximum objective value of max $p^T x$, $Ax \leq b$ whereby for $\{x \mid Ax \leq b\}$ it is easy to find an interior solution.
- What prevents us actually from using methods from calculus to solve our problem?
- The non-differentiable shape of the polytope (corners!) causes problems.
- Can we smoothen the shape of the feasible region?

Interior Point Algorithms

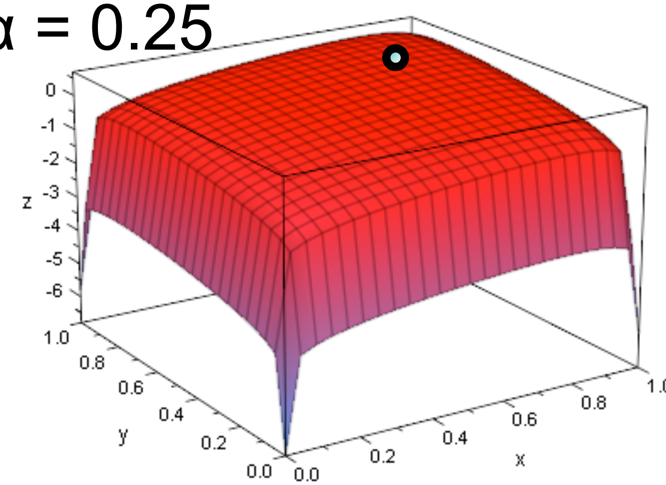
- Instead of enforcing that solutions are within the feasible region via inequalities, instead we can make solutions more and more unattractive the closer we get to the border.
- This idea yields to the notion of barrier functions:
  - A barrier function goes to $-\infty$ as $Ax \rightarrow b$ and should be differentiable.
  - $\max p^T x + \alpha \left(\Sigma_i \log (x_i) + \Sigma_i \log (\Sigma_j a_{ij} x_i - b_i)\right)$
- Using standard methods from calculus, we can maximize such functions $\Rightarrow$ Newton method
- By decreasing the barrier parameter α, we get closer and closer to the true maximal value.

Interior Point Algorithms

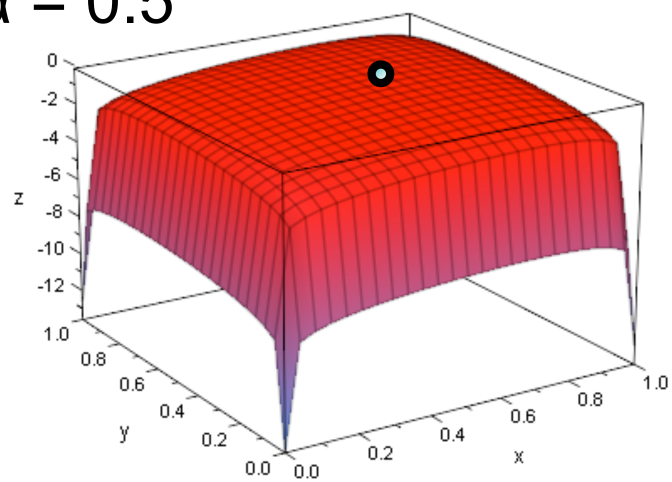
α = 1



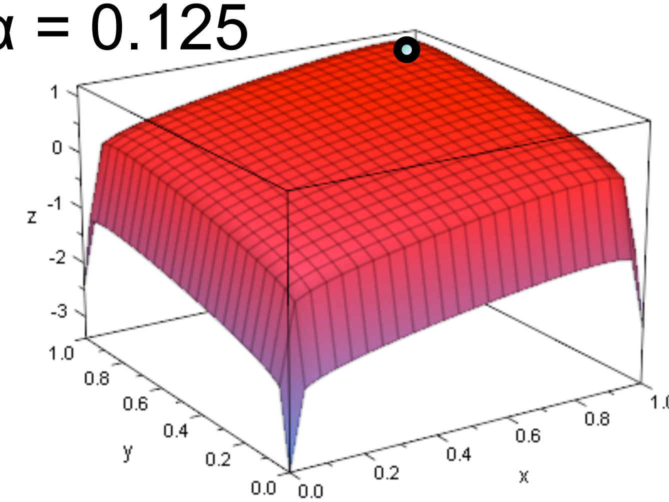
α = 0.25



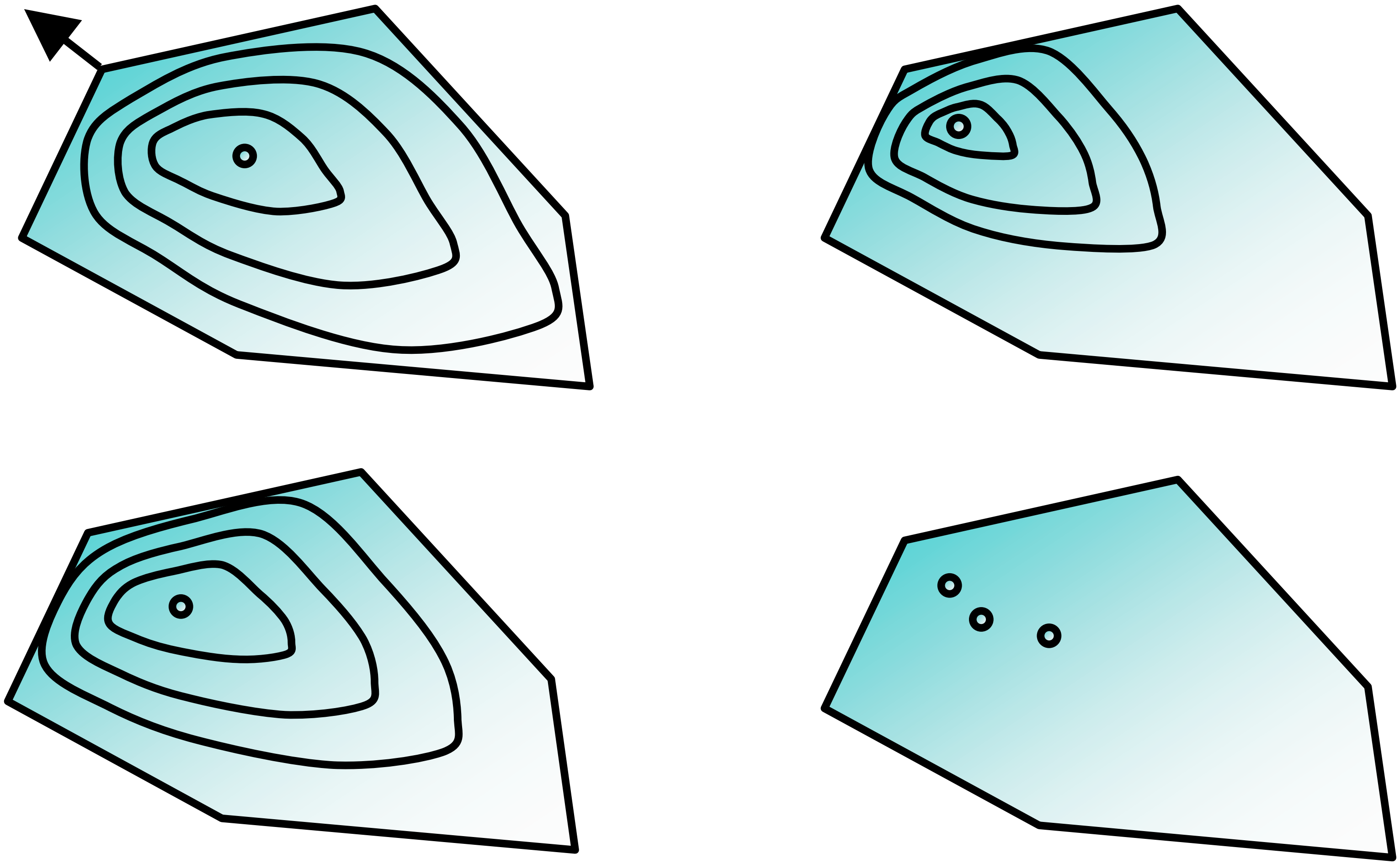
α = 0.5



α = 0.125



Interior Point Algorithms



Thank you!