# Comparing Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Worst Case</th>
<th>‘Avg’ Case</th>
<th>Best Suited For ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplex</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ellipsoid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barrier Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparing Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Worst Case</th>
<th>‘Avg’ Case</th>
<th>Best Suited For …</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplex</td>
<td>Move along outside between adjacent vertices</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ellipsoid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barrier Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparing Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Worst Case</th>
<th>‘Avg’ Case</th>
<th>Best Suited For ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplex</td>
<td>Move along outside between adjacent vertices</td>
<td>Exponential for all known pivoting rules</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ellipsoid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barrier Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparing Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Worst Case</th>
<th>‘Avg’ Case</th>
<th>Best Suited For ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplex</td>
<td>Move along outside between adjacent vertices</td>
<td>Exponential for all known pivoting rules</td>
<td>O(nm)</td>
<td></td>
</tr>
<tr>
<td>Ellipsoid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barrier Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Comparing Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Worst Case</th>
<th>‘Avg’ Case</th>
<th>Best Suited For ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplex</td>
<td>Move along outside between adjacent vertices</td>
<td>Exponential for all known pivoting rules</td>
<td>O(nm)</td>
<td>• Re-optimization</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Customizable (Column Gen)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• IP, MIP</td>
</tr>
<tr>
<td>Ellipsoid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barrier Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Some History

- Yudin and Nemirovski (1976)
- Shor (1977)
- Khachiyan (1980)
- Grotschel, Lovasz, Schrijiver (1988)
Let’s play a game....

- Keys to strategy
  - Identify a bounding interval
  - Identify smallest possible interval
  - Find a good “center” $x^{(i)}$
  - Given a guess $x^{(i)}$, determine if bigger or smaller.

How many iterations worst case?
How many iterations “on average”?
Is the $\frac{1}{2}$ critical?

How does this work in larger dimensions?
A Typical Iteration
Key Lemma

- $E = E(z, D)$ be an ellipsoid in $\mathbb{R}^n$; $\alpha$ nonzero $n$-vector.
- $H = \{ x \in \mathbb{R}^n | \alpha' x \geq \alpha' z \}$

\[
\bar{z} = z + \frac{1}{n+1} \frac{Da}{\sqrt{\alpha'D\alpha}}, \\
\bar{D} = \frac{n^2}{n^2 - 1} \left( D - \frac{2}{n+1} \frac{D\alpha\alpha'D}{\alpha'D\alpha} \right).
\]

- The matrix $\bar{D}$ is symmetric and positive definite and thus $E' = E(\bar{z}, \bar{D})$ is an ellipsoid
- $E \cap H \subset E'$
- $\text{Vol}(E') < e^{-1/(2(n+1))} \text{Vol}(E)$
Algorithm 1 Ellipsoid Algorithm

Input:

- \( \{P = \{x \in \mathbb{R}^n | Ax \leq b\}\} \)
- An ellipsoid \( E_0 \) such that \( P \subseteq E_0 \).

Output: A point \( x \in P \) if it \( P \) non-empty, else a statement \( P \) is empty.

1: for \( i = 0, \ldots, t \) do
2: \( x^i \leftarrow \) center of \( E_0 \).
3: if \( x^i \in P \) then
4: \( \text{return } x^i \)
5: else
6: \( \) Find a cut \( a \) such that \( a'x \leq a'x^i \) \( \forall x \in P \).
7: \( E_{i+1} \leftarrow E(x^{i+1}, D_{i+1}) \) defined as in previous Lemma.
8: end if
9: end for
10: \( \text{return } P \) is empty.
Details?

• How should we choose $E_0$? (See Text)
  $$Vol(E_0) \leq V = (2n(nU)^n)^n = (2n)^n(nU)^{n^2}$$

• How big should $t$ be?
  • We can find a $v$ such that $Vol(P) > v$ (See Text)
  $$v = n^{-n}(nU)^{-n^2(n+1)}$$
  $$t = \left\lfloor (2n + 1) \log \left( \frac{Vol(E_0)}{v} \right) \right\rfloor$$
  $$= \mathcal{O}(n \log \left( \frac{Vol(E_0)}{v} \right))$$
  $$= \mathcal{O}(n^4 \log(nU))$$
Theoretical implications

- This leads to a polynomial number of iterations in the worst case!
- How much work is there per iteration?
- I can also optimize!!

- Important Theoretical Tool: If you can separate over the feasible set, you can optimize over it.

Separation = Optimization
What should I remember?

- Remember the link to bisection search
- Remember the picture, and number of iterations
- Remember the factor for reduction

\[
\frac{Vol(E_{i+1})}{Vol(E_i)} \leq e^{\frac{1}{2n+2}}
\]

Finally, SEPARATION!
## Comparing Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Worst Case</th>
<th>‘Avg’ Case</th>
<th>Best Suited For ...</th>
</tr>
</thead>
</table>
| Simplex    | Move along outside between adjacent vertices                                | Super-polynomial for all known pivoting rules   | O(nm)                               | • Re-optimization  
• IP, MIP  
• Customizable (Column) |
| Ellipsoid  | Bound feasible region by ellipsoid and sequentially add cuts to find feasible point, or decide none exists. | $O(n^6 \log(nU))$ | Close to worst case                  | • Separation Oracle  
• Exponential number of constraints  
• PROVING complexity results |
| Barrier Methods |                                                                 |                                                  |                                     |                                                          |
Motivating Barrier Methods

- Large class of methods
- ‘State of the Art’
- Applicable to many GENERIC CONVEX programs
- Focus on Primal Barrier
- Historical Aside....
Central Path

Central path is a concept in optimization theory.
Solution to the Barrier

What does this remind you of?

\[
\begin{align*}
Ax(\mu) &= b \\
\quad x(\mu) &\geq 0 \\
A'p(\mu) + s(\mu) &= c \\
\quad s(\mu) &\geq 0 \\
X(\mu)S(\mu)e &= e\mu
\end{align*}
\]
Umm....

- The important thing is that we can devise an algorithm (using Newton’s method) that approximately solves the above system.
- We call this algorithm many times to solve the above system for smaller and smaller values of \( u \).
- Complexity?

Let

\[ \epsilon_0 = (s^0)'x^0 \]

initial duality gap.

To reduce duality gap to \( \epsilon \) requires

\[ O(\sqrt{n} \log \frac{\epsilon_0}{\epsilon}) \]
More importantly…

- What might we deduce about the behavior of the algorithm?
# Comparing Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Worst Case</th>
<th>‘Avg’ Case</th>
<th>Best Suited For ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplex</td>
<td>Move along outside between adjacent vertices</td>
<td>Super-polynomial for all known pivoting rules</td>
<td>O(nm)</td>
<td>• Re-optimization</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• IP, MIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Customizable (Column Gen)</td>
</tr>
<tr>
<td>Ellipsoid</td>
<td>Bound feasible region by ellipsoid and sequentially add cuts to find feasible point, or decide none exists.</td>
<td>O(n^6 log(nU))</td>
<td>Close to worst case</td>
<td>• Separation Oracle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Exponential number of constraints</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• PROVING complexity results</td>
</tr>
<tr>
<td>Barrier Methods</td>
<td>Penalize objective for approaching boundary. Follow central path.</td>
<td>O(n^{3.5} log(1/ε))</td>
<td>‘log(n) iterations’</td>
<td>• Large, degenerate problems</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Large unstructured problems</td>
</tr>
</tbody>
</table>
A closing quote

“What can one say in general about the best way to solve large models? Which algorithm is best? If this question had been asked in 1998, our response would have been that barrier was clearly best for large models. If that question were asked now, our response would be that there is no clear, best algorithm. Each of primal, dual, and barrier is superior in a significant number of important instances.”

Bixby (Taken from Todd’s “The Many Facets of Linear Programming)
Some Further Resources

- Ellipsoid Algorithm
  - From your book: 8.2 (just results), 8.3-8.5
  - The definitive: Grotschel, Lovasz, Schrijver

- Barrier Methods
  - From your book: 9.4 (intuition only)
  - R. Freund’s Notes available online
The Next Step?

- Network Optimization
- Integer Programming
- Nonlinear Programming, Convex Optimization, Convex Analysis
- Systems Optimization
- Algebraic Methods in Semidefinite Optimization