Decision Procedures
An Algorithmic Point of View

Gaussian Elimination and Simplex

Daniel Kroening and Ofer Strichman
Gaussian’s elimination

- Given a linear system \( Ax = b \)

\[
\begin{bmatrix}
  a_{11} & a_{12} & \ldots & a_{1k} \\
  a_{21} & a_{22} & \ldots & a_{2k} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{k1} & a_{k2} & \ldots & a_{kk}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_k
\end{bmatrix} =
\begin{bmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_k
\end{bmatrix}
\]

- Manipulate \( A|b \) to an upper-triangular form
Gaussian’s elimination

Then, solve backwards from the $k$’s row according to:

$$x_i = \frac{1}{a'_{ii}} (b'_i - \sum_{j=i+1}^{k} a'_{ij} x_j)$$
Gaussian elimination - example

\[
\begin{pmatrix}
1 & 2 & 1 \\
-2 & 3 & 4 \\
4 & -1 & -8
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
= 
\begin{pmatrix}
6 \\
3 \\
9
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 2 & 1 | 6 \\
-2 & 3 & 4 | 3 \\
0 & -9 & -12 | -15
\end{pmatrix}
= 
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
\]

\[
R3+ = -4R1
\]

\[
\begin{pmatrix}
1 & 2 & 1 | 6 \\
0 & 7 & 6 | 15 \\
0 & -9 & -12 | -15
\end{pmatrix}
= 
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
\]

\[
R2+ = 2R1
\]

\[
\begin{pmatrix}
1 & 2 & 1 | 6 \\
0 & 7 & 6 | 15 \\
0 & 0 & -\frac{30}{7} | \frac{30}{7}
\end{pmatrix}
= 
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
\]

\[
R3+ = \frac{9}{7}R2
\]

And now... \( x_3 = -1, x_2 = 3, x_1 = 1 \)  problem solved.
Feasibility with Simplex

- Simplex was originally designed for solving the optimization problem:

\[
\begin{align*}
\max & \quad \vec{c} \vec{x} \\
\text{s.t.} & \quad A\vec{x} \leq \vec{b}, \quad \vec{x} \geq \vec{0}
\end{align*}
\]

- We are only interested in the feasibility problem.

Is this system feasible? Is this system optimal?
General simplex

- We will learn a variant called general simplex.
- Very suitable for solving the feasibility problem fast.
- The input: \( A\vec{x} \leq \vec{b} \)

  - \( A \) is a \( m \times n \) coefficient matrix
  - The problem variables: \( \vec{x} = x_1, \ldots, x_n \)

- First step: convert the input to general form
General form

- General form: \( A\vec{x} = 0 \) and \( \bigwedge_{i=1}^{m} l_i \leq s_i \leq u_i \)

- A combination of:
  - Linear equalities of the form \( \sum_i a_i x_i = 0 \)
  - Lower and upper bounds on variables.
Converting to General Form

- **A:** Replace $\sum_i a_i x_i \bowtie b_j$ \ (where $\bowtie \in \{=, \leq, \geq\}$) with $\sum_i a_i x_i - s_j = 0$

  and $s_j \bowtie b_j$

- $s_1, \ldots, s_m$ are called the additional variables.
Example 1

Convert \( x + y \geq 2 \)

to: \( x + y - s_1 = 0 \)

\( s_1 \geq 2 \)

It is common to keep the conjunctions implicit.
Example 2

- Convert

\[
\begin{align*}
x &+ y &\geq 2 \\
2x &- y &\geq 0 \\
-x &+ 2y &\geq 1
\end{align*}
\]

to:

\[
\begin{align*}
x &+ y &- s_1 &= 0 \\
2x &- y &- s_2 &= 0 \\
-x &+ 2y &- s_3 &= 0 \\
s_1 &\geq 2 \\
s_2 &\geq 0 \\
s_3 &\geq 1
\end{align*}
\]
Simplex basics…

- Linear inequality constraints, geometrically, define a convex polyhedron.
Our example from before, geometrically

\[
\begin{align*}
x + y & \geq 2 \\
2x - y & \geq 0 \\
-x + 2y & \geq 1
\end{align*}
\]

General Simplex begins in the origin...
Matrix form

- Recall the general form: \( A\vec{x} = 0 \) and \( \bigwedge_{i=1}^{m} l_i \leq s_i \leq u_i \)
- Due to the additional variables:
  - now \( A \) is an \( m \times (n + m) \) matrix.

\[
\begin{align*}
  x + y - s_1 &= 0 \\
  2x - y - s_2 &= 0 \\
  -x + 2y - s_3 &= 0 \\
  s_1 &\geq 2 \\
  s_2 &\geq 0 \\
  s_3 &\geq 1
\end{align*}
\]

\[
\begin{pmatrix}
  x & y & s_1 & s_2 & s_3 \\
  1 & 1 & -1 & 0 & 0 \\
  2 & -1 & 0 & -1 & 0 \\
 -1 & 2 & 0 & 0 & -1
\end{pmatrix}
\]
The tableau

- The diagonal part is inherent to the general form

\[
\begin{pmatrix}
  x & y & s_1 & s_2 & s_3 \\
  1 & 1 & -1 & 0 & 0 \\
  2 & -1 & 0 & -1 & 0 \\
  -1 & 2 & 0 & 0 & -1 \\
\end{pmatrix}
\]

- We can instead write:

\[
\begin{pmatrix}
  x & y \\
  s_1 & \begin{pmatrix} 1 & 1 \\ 2 & -1 \\ -1 & 2 \end{pmatrix} \\
\end{pmatrix}
\]

This is called the tableau
The tableau

- The tableau changes throughout the algorithm, but maintains its $m \times n$ structure

\[
\begin{array}{ccc}
\text{Basic variables} & x & y \\
 s_1 & \begin{pmatrix} 1 & 1 \\ 2 & -1 \\ -1 & 2 \end{pmatrix} & \\
 s_2 & \\
 s_3 & \\
\end{array}
\]

- Distinguish between basic and nonbasic variables
- Initially, basic variables = the additional variables.
The tableau

- Denote by
  - $\mathcal{B}$ – Basic variables
  - $\mathcal{N}$ – Nonbasic variables

- The tableau is simply a rewrite of the system:

$$\bigwedge_{x_i \in \mathcal{B}} \left( x_i = \sum_{x_j \in \mathcal{N}} a_{ij} x_j \right)$$

- The basic variables are also called the dependent variables.
The general simplex algorithm

- Simplex maintains:
  - The tableau,
  - an assignment $\alpha$ to all variables
  - The bounds

- Initially,
  - $\mathcal{B} = \text{additional variables}$
  - $\mathcal{N} = \text{problem variables}$
  - $\alpha(x_i) = 0$ for $i \in \{1,\ldots,n+m\}$
Invariants

- Two invariants are maintained throughout:
  1. $A\bar{x} = 0$
  2. All nonbasic variables satisfy their bounds

- Can you see why these invariants are maintained initially?
- We should check that they are indeed maintained
The general simplex algorithm

- The initial assignment satisfies $A\vec{x} = 0$
- If the bounds of all basic variables are satisfied by $\alpha$, return `Satisfiable'.
- Otherwise... pivot.
Pivoting

- Find a basic variable $x_i$ that violates its bounds.
  - Suppose that $\alpha(x_i) < l_i$

- Find a nonbasic variable $x_j$ such that
  - $a_{ij} > 0$ and $\alpha(x_j) < u_j$, or
  - $a_{ij} < 0$ and $\alpha(x_j) > l_j$

- Why?
Pivoting

- Find a basic variable $x_i$ that violates its bounds.
  - Suppose that $\alpha(x_i) < l_i$
- Find a nonbasic variable $x_j$ such that
  - $a_{ij} > 0$ and $\alpha(x_j) < u_j$, or
  - $a_{ij} < 0$ and $\alpha(x_j) > l_j$
- Such a variable $x_j$ is called suitable.
- If there is no suitable variable – return ‘Unsatisfiable’
  - Why?
Pivoting $x_i$ with $x_j$

- Solve equation $i$ for $x_j$:
  
  From: $x_i = a_{ij}x_j + \sum_{k \neq j} a_{ik}x_k$
  
  To: $x_j = \frac{x_i}{a_{ij}} - \sum_{k \neq j} \frac{a_{ik}}{a_{ij}}x_k$

- Swap $x_i$ and $x_j$, and update the $i$-th row accordingly.

From

| $a_{i1}$ | ... | $a_{ij}$ | ... | $a_{in}$ |

To:

| $-a_{i1}$ | ... | $1$ | ... | $-a_{in}$ |

\[
\begin{array}{c|c|c|c}
\hline
-a_{i1} & \cdots & 1 & -a_{in} \\
\hline
a_{ij} & \cdots & a_{ij} & a_{ij} \\
\hline
\end{array}
\]
Pivoting $x_i$ with $x_j$

- Update all other rows:
  - Replace $x_j$ with its equivalent obtained from row $i$:

$$x_j = \frac{x_i}{a_{ij}} - \sum_{k \neq j} \frac{a_{ik}}{a_{ij}} x_k$$
Pivoting

- Update $\alpha$ as follows:
- Increase $\alpha(x_j)$ by $\theta = \frac{l_i - \alpha(x_i)}{a_{ij}}$
  - Now $x_j$ is a basic variable: it can violate its bounds.

- Update $\alpha(x_i)$ accordingly
  - Q: What is now $\alpha(x_i)$?

- Update $\alpha$ for all other basic (dependent) variables.
Example

- Recall the tableau and constraints in our example:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$s_2$</td>
<td>2</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>$s_3$</td>
<td>-1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

- Initially $\alpha$ assigns 0 to all variables
- Bounds of $s_1$ and $s_3$ are violated
Example

- Recall the tableau and constraints in our example:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$s_2$</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>$s_3$</td>
<td>-1</td>
<td>2</td>
</tr>
</tbody>
</table>

- We will solve $s_1$
- $x$ is a suitable nonbasic variable for pivoting
  - It has no upper bound
- So now we pivot $s_1$ with $x$
Example

- Recall the tableau and constraints in our example:

  \[
  \begin{array}{c|cc}
  & x & y \\
  \hline
  s_1 & 1 & 1 \\
  s_2 & 2 & -1 \\
  s_3 & -1 & 2 \\
  \end{array}
  \begin{array}{c}
  2 \leq s_1 \\
  0 \leq s_2 \\
  1 \leq s_3 \\
  \end{array}
  \]

- Solve 1\textsuperscript{st} row for \(x\): \(s_1 = x + y \iff x = s_1 - y\)

- Replace \(x\) with \(s_1\) in other rows:

  \[s_2 = 2(s_1 - y) - y \iff s_2 = 2s_1 - 3y\]

  \[s_3 = -(s_1 - y) + 2y \iff s_3 = -s_1 + 3y\]
Example

The new state:

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>$s_2$</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>$s_3$</td>
<td>-1</td>
<td>3</td>
</tr>
</tbody>
</table>

Solve 1$^{\text{st}}$ row for $x$: $s_1 = x + y \iff x = s_1 - y$

Replace $x$ with $s_1$ in other rows:

$s_2 = 2(s_1 - y) - y \iff s_2 = 2s_1 - 3y$

$s_3 = -(s_1 - y) + 2y \iff s_3 = -s_1 + 3y$
Example

- The new state:

<table>
<thead>
<tr>
<th>s₁</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>s₂</td>
<td>2</td>
</tr>
<tr>
<td>s₃</td>
<td>-1</td>
</tr>
</tbody>
</table>

  2 ≤ s₁
  0 ≤ s₂
  1 ≤ s₃

- What about the assignment?

- We should increase x by \( \theta = \frac{2 - 0}{1} = 2 \)
  - Hence, \( \alpha(x) = 0 + 2 = 2 \)
  - Now \( s₁ \) is equal to its lower bound: \( \alpha(s₁) = 2 \)
  - Update all the others
Example

- **The new state:**

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>$s_2$</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>$s_3$</td>
<td>-1</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$\alpha(x)$</th>
<th>$\alpha(y)$</th>
<th>$2 \leq s_1$</th>
<th>$\alpha(s_1)$</th>
<th>$\alpha(s_2)$</th>
<th>$\alpha(s_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>-2</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>-2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>-2</td>
</tr>
</tbody>
</table>

- **Now $s_3$ violates its lower bound**

- **Which nonbasic variable is suitable for pivoting?**

  - That’s right… $y$
Example

The new state:

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$y$</th>
<th>$\alpha(x)$</th>
<th>$\alpha(y)$</th>
<th>$\alpha(s_1)$</th>
<th>$\alpha(s_2)$</th>
<th>$\alpha(s_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>-2</td>
</tr>
<tr>
<td>$s_2$</td>
<td>2</td>
<td>-3</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$s_3$</td>
<td>-1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We should increase $y$ by $\theta = \frac{1 - (-2)}{3} = 1$
Example

- The final state:

<table>
<thead>
<tr>
<th>x</th>
<th>2/3</th>
<th>-1/3</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1</td>
<td>s_3</td>
<td></td>
</tr>
<tr>
<td>s_2</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>y</td>
<td>1/3</td>
<td>1/3</td>
</tr>
</tbody>
</table>

\[ \alpha(x) = 1 \]
\[ \alpha(y) = 1 \]
\[ \alpha(s_1) = 2 \]
\[ \alpha(s_2) = 1 \]
\[ \alpha(s_3) = 1 \]

- All constraints are now satisfied
Observations

- The additional variables:
  - Only additional variables have bounds.
  - These bounds are permanent.
  - Additional variables exit the base only on extreme points (their lower or upper bounds).
  - When entering the base, they shift towards the other bound and possibly cross it (violate it).
Observations

Can it be that we $\text{pivot}(x_i, x_j)$ and then $\text{pivot}(x_j, x_i)$ and enter a (local) cycle?

□ No.
□ For example, suppose that $a_{ij} > 0$.
□ We increased $\alpha(x_j)$ so now $\alpha(x_i) = l_i$.
□ After pivoting, possibly $\alpha(x_j) > u_j$
□ But $a_{ij} = 1 / a_{ij} > 0$, hence $x_i$ is not suitable.
Observations

- Is termination guaranteed?
  - Not obvious.
    - Perhaps there are bigger cycles.
- In order to avoid circles, we use Bland’s rule:
  - Determine a total order on the variables.
  - Choose the first basic variable that violates its bounds, and first nonbasic suitable variable for pivoting.
  - It can be proven that this guarantees that no base is repeated, which implies termination.
1. Transform the system into the general form

\[ A\vec{x} = 0 \quad \text{and} \quad \bigwedge_{i=1}^{m} l_i \leq s_i \leq u_i. \]

2. Set \( B \) to be the set of additional variables \( s_1, \ldots, s_m \).

3. Construct the tableau for \( A \).

4. Determine a fixed order on the variables.

5. If there is no basic variable that violates its bounds, return “Satisfiable”. Otherwise, let \( x_i \) be the first basic variable in the order that violates its bounds.

6. Search for the first suitable nonbasic variable \( x_j \) in the order for pivoting with \( x_i \). If there is no such variable, return “Unsatisfiable”.

7. Perform the pivot operation on \( x_i \) and \( x_j \).

8. Go to step 5.