ECE750T-28:
Computer-aided Reasoning for Software Engineering

Lecture 13: Decision Procedure for the Theory of Rationals

Vijay Ganesh
(Original notes from Isil Dillig)
Earlier, we looked at signature and axioms of $T_{\mathbb{Q}}$
Theory of Rationals $T_{\mathbb{Q}}$

- Earlier, we looked at signature and axioms of $T_{\mathbb{Q}}$

- Signature

$$\Sigma_{\mathbb{Q}} : \{0, 1, +, -, =, \geq\}$$
Theory of Rationals $T_Q$

- Earlier, we looked at signature and axioms of $T_Q$

- Signature

  $$\Sigma_Q : \{0, 1, +, -, =, \geq\}$$

- Axioms interpret (i.e., give meaning) to all object, function, and relation constants
Theory of Rationals $T_{\mathbb{Q}}$

- Earlier, we looked at signature and axioms of $T_{\mathbb{Q}}$

- Signature

  $\Sigma_{\mathbb{Q}} : \{0, 1, +, -, =, \geq\}$

- Axioms interpret (i.e., give meaning) to all object, function, and relation constants

- **Today:** Talk about how to decide satisfiability of the quantifier-free fragment of $T_{\mathbb{Q}}$
Distinction between Theory of Rationals and Presburger Arithmetic

- $T_\mathbb{Q}$ has too many axioms, so we won’t discuss them
Distinction between Theory of Rationals and Presburger Arithmetic

- $T_Q$ has too many axioms, so we won’t discuss them

- **Distinction between $T_Z$ and $T_Q$:** Rational numbers do not satisfy the more restrictive $T_Z$ axioms
Distinction between Theory of Rationals and Presburger Arithmetic

- \( T_\mathbb{Q} \) has too many axioms, so we won’t discuss them

- Distinction between \( T_\mathbb{Z} \) and \( T_\mathbb{Q} \): Rational numbers do not satisfy the more restrictive \( T_\mathbb{Z} \) axioms

- Example: \( \exists x. (1 + 1)x = 1 + 1 + 1 \) Is this formula valid in \( T_\mathbb{Q} \)?
Distinction between Theory of Rationals and Presburger Arithmetic

- $T_Q$ has too many axioms, so we won’t discuss them

- **Distinction between $T_Z$ and $T_Q$:** Rational numbers do not satisfy the more restrictive $T_Z$ axioms

- **Example:** $\exists x. (1 + 1)x = 1 + 1 + 1$ Is this formula valid in $T_Q$? Yes
Distinction between Theory of Rationals and Presburger Arithmetic

- $T_Q$ has too many axioms, so we won’t discuss them.

- **Distinction between $T_Z$ and $T_Q$:** Rational numbers do not satisfy the more restrictive $T_Z$ axioms.

  - **Example:** $\exists x. (1 + 1)x = 1 + 1 + 1$ Is this formula valid in $T_Q$? Yes

  - Is it valid in $T_Z$?
Distinction between Theory of Rationals and Presburger Arithmetic

- $T_Q$ has too many axioms, so we won’t discuss them

- Distinction between $T_Z$ and $T_Q$: Rational numbers do not satisfy the more restrictive $T_Z$ axioms

- Example: $\exists x. (1 + 1)x = 1 + 1 + 1$ Is this formula valid in $T_Q$? Yes

- Is it valid in $T_Z$? No
Distinction between Theory of Rationals and Presburger Arithmetic

- $T_Q$ has too many axioms, so we won’t discuss them

- Distinction between $T_Z$ and $T_Q$: Rational numbers do not satisfy the more restrictive $T_Z$ axioms

- Example: $\exists x. (1 + 1)x = 1 + 1 + 1$ Is this formula valid in $T_Q$? Yes

- Is it valid in $T_Z$? No

- In general, every formula valid in $T_Z$ is valid in $T_Q$, but not vice versa
Decidability and Complexity Results for $T_Q$

- Full theory of rationals is decidable
Decidability and Complexity Results for $T_Q$

- Full theory of rationals is **decidable**
- High-time complexity: $O(2^{kn})$ ($k$: some positive integer)
Decidability and Complexity Results for $T_Q$

- Full theory of rationals is **decidable**

- High-time complexity: $O(2^{2^kn})$ ($k$: some positive integer)

- Conjunctive quantifier-free fragment efficiently decidable (polynomial time)
Overview

- We’ll only consider quantifier free conjunctive $T_Q$ formulas (i.e., no disjunctions)
We’ll only consider quantifier free conjunctive $T_Q$ formulas (i.e., no disjunctions).

Not a big restriction because if we have a decision procedure for conjunctive fragment, we can decide satisfiability of any quantifier-free formula (e.g., by converting to DNF).
Overview

- We’ll only consider quantifier free conjunctive $T_Q$ formulas (i.e., no disjunctions)

- Not a big restriction because if we have a decision procedure for conjunctive fragment, we can decide satisfiability of any quantifier-free formula (e.g., by converting to DNF)

- Most common technique for deciding satisfiability in $T_Q$ is Simplex algorithm
Overview

- We’ll only consider quantifier free conjunctive $T_Q$ formulas (i.e., no disjunctions)

- Not a big restriction because if we have a decision procedure for conjunctive fragment, we can decide satisfiability of any quantifier-free formula (e.g., by converting to DNF)

- Most common technique for deciding satisfiability in $T_Q$ is Simplex algorithm

- Simplex algorithm developed by Dantzig in 1949 for solving linear programming problems
We’ll only consider quantifier free conjunctive $T_Q$ formulas (i.e., no disjunctions).

Not a big restriction because if we have a decision procedure for conjunctive fragment, we can decide satisfiability of any quantifier-free formula (e.g., by converting to DNF).

Most common technique for deciding satisfiability in $T_Q$ is Simplex algorithm.

Simplex algorithm developed by Dantzig in 1949 for solving linear programming problems.

Since deciding satisfiability of qff conjunctive formulas is a special case of linear programming, we can use Simplex.
The Plan

- Overview of linear programming
The Plan

- Overview of linear programming
- Satisfiability as linear programming
The Plan

- Overview of linear programming
- Satisfiability as linear programming
- Simplex algorithm
In a linear programming (LP) problem, we have an $m \times n$ matrix $A$, an $m$-dimensional vector $\vec{b}$, and $n$-dimensional vector $\vec{c}$. Very important problem; applications in airline scheduling, transportation, telecommunications, finance, production management, marketing, networking, compilers...
Linear Programming

- In a linear programming (LP) problem, we have an $m \times n$ matrix $A$, an $m$-dimensional vector $\vec{b}$, and an $n$-dimensional vector $\vec{c}$.

- Want to find a solution for $\vec{x}$ maximizing objective function $\vec{c}^T \vec{x}$ subject to linear inequality constraint $A\vec{x} \leq \vec{b}$.
Linear Programming

- In a linear programming (LP) problem, we have an $m \times n$ matrix $A$, an $m$-dimensional vector $\vec{b}$, and $n$-dimensional vector $\vec{c}$

- Want to find a solution for $\vec{x}$ maximizing objective function

$$\vec{c}^T \vec{x}$$

subject to linear inequality constraint

$$A\vec{x} \leq \vec{b}$$

- Very important problem; applications in airline scheduling, transportation, telecommunications, finance, production management, marketing, networking, compilers . . .
For \( m \times n \) matrix \( A \), the system \( A\vec{x} \leq \vec{b} \) forms a **convex polytope** in \( n \)-dimensional space.
Geometric Formulation

- For \( m \times n \) matrix \( A \), the system \( A \vec{x} \leq \vec{b} \) forms a convex polytope in \( n \)-dimensional space.

- Polytope is generalization of polyhedron from 3-dim space to higher dimensional space.
For $m \times n$ matrix $A$, the system $A\vec{x} \leq \vec{b}$ forms a convex polytope in $n$-dimensional space.

Polytope is generalization of polyhedron from 3-dim space to higher dimensional space.

Convexity: For all pairs of points $\vec{v}_1, \vec{v}_2$ and for any $\lambda \in [0, 1]$, the point $\lambda \vec{v}_1 + (1 - \lambda) \vec{v}_2$ also lies in polytope.
Geometric Formulation

- For a $m \times n$ matrix $A$, the system $A \vec{x} \leq \vec{b}$ forms a convex polytope in $n$-dimensional space.

- Polytope is a generalization of polyhedron from 3-dim space to higher dimensional space.

- Convexity: For all pairs of points $\vec{v}_1, \vec{v}_2$ and for any $\lambda \in [0, 1]$, the point $\lambda \vec{v}_1 + (1 - \lambda) \vec{v}_2$ also lies in polytope.
Geometric Formulation

- For $m \times n$ matrix $A$, the system $A\vec{x} \leq \vec{b}$ forms a convex polytope in $n$-dimensional space.

- Polytope is a generalization of polyhedron from 3-D space to higher dimensional space.

- **Convexity**: For all pairs of points $\vec{v}_1, \vec{v}_2$ and for any $\lambda \in [0, 1]$, the point $\lambda \vec{v}_1 + (1 - \lambda) \vec{v}_2$ also lies in the polytope.

- **Goal of linear programming**: Find a point that (i) lies inside the polytope, and (ii) maximizes the value of $\vec{c}^T \vec{x}$. 
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called feasible solution; otherwise, called infeasible solution
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called feasible solution; otherwise, called infeasible solution

- **Example:** Maximize $2y - x$ subject to:

  
  \[
  \begin{align*}
  x + y & \leq 3 \\
  2x - y & \leq -5
  \end{align*}
  \]
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called **feasible solution**; otherwise, called **infeasible solution**

- **Example:** Maximize $2y - x$ subject to:

$$
\begin{align*}
x + y & \leq 3 \\
2x - y & \leq -5
\end{align*}
$$

- Is $(0, 0)$ a feasible solution?
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called feasible solution; otherwise, called infeasible solution

- Example: Maximize $2y - x$ subject to:

$$\begin{align*}
x + y & \leq 3 \\
2x - y & \leq -5
\end{align*}$$

- Is $(0, 0)$ a feasible solution? No
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called feasible solution; otherwise, called infeasible solution.

- Example: Maximize $2y - x$ subject to:
  \[
  \begin{align*}
  x + y & \leq 3 \\
  2x - y & \leq -5
  \end{align*}
  \]

- Is $(0, 0)$ a feasible solution? No

- What about $(-2, 1)$?
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called **feasible solution**; otherwise, called **infeasible solution**

- **Example**: Maximize $2y - x$ subject to:

  $$
  \begin{align*}
  x + y & \leq 3 \\
  2x - y & \leq -5
  \end{align*}
  $$

- Is $(0, 0)$ a feasible solution? **No**

- What about $(-2, 1)$? **Yes**
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called \textbf{feasible solution}; otherwise, called \textbf{infeasible solution}

- Example: Maximize $2y - x$ subject to:

\[
\begin{align*}
x + y & \leq 3 \\
2x - y & \leq -5
\end{align*}
\]

- Is $(0, 0)$ a feasible solution? No

- What about $(-2, 1)$? Yes

- For a given solution for $\vec{x}$, the corresponding value of objective function $\vec{c}^T \vec{x}$ called \textbf{objective value}
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called feasible solution; otherwise, called infeasible solution.

- Example: Maximize $2y - x$ subject to:
  
  \[
  \begin{align*}
  x + y & \leq 3 \\
  2x - y & \leq -5 
  \end{align*}
  \]

- Is $(0, 0)$ a feasible solution? No

- What about $(-2, 1)$? Yes

- For a given solution for $\vec{x}$, the corresponding value of objective function $\vec{c}^T \vec{x}$ called objective value

- What is objective value for $(-2, 1)$?
Linear Programming Lingo

- In LP, a value of $\vec{x}$ that satisfies constraints $A\vec{x} \leq \vec{b}$ called feasible solution; otherwise, called infeasible solution.

- Example: Maximize $2y - x$ subject to:
  
  $\begin{align*}
  x + y & \leq 3 \\
  2x - y & \leq -5
  \end{align*}$

- Is $(0, 0)$ a feasible solution? No
- What about $(-2, 1)$? Yes

- For a given solution for $\vec{x}$, the corresponding value of objective function $\vec{c}^T \vec{x}$ called objective value

- What is objective value for $(-2, 1)$? 4
A feasible solution whose objective value is maximum over all feasible solutions called **optimal solution**.
Linear Programming Lingo, cont

- A feasible solution whose objective value is maximum over all feasible solutions called **optimal solution**

- If a linear program has no feasible solutions, the linear program is **infeasible**
Linear Programming Lingo, cont

- A feasible solution whose objective value is maximum over all feasible solutions called **optimal solution**

- If a linear program has no feasible solutions, the linear program is **infeasible**

- If optimal solution is $\infty$, then problem is called **unbounded**
Geometric Interpretation

- Feasible solution is a point within the polytope
Geometric Interpretation

- Feasible solution is a point within the polytope
- The linear programming problem is infeasible if the polytope defined by $A\vec{x} \leq \vec{b}$ is empty
Geometric Interpretation

- Feasible solution is a point within the polytope
- The linear programming problem is infeasible if the polytope defined by $A\vec{x} \leq \vec{b}$ is empty
- An LP problem is unbounded if the polytope is open in the direction of the objective function
Geometric Interpretation

- Feasible solution is a point within the polytope
- The linear programming problem is infeasible if the polytope defined by $A\vec{x} \leq \vec{b}$ is empty
- An LP problem is unbounded if the polytope is open in the direction of the objective function

**Question:** If polytope is not closed, does this mean optimal solution is $\infty$?
Geometric Interpretation

- Feasible solution is a point within the polytope
- The linear programming problem is infeasible if the polytope defined by $A\vec{x} \leq \vec{b}$ is empty
- An LP problem is unbounded if the polytope is open in the direction of the objective function

**Question:** If polytope is not closed, does this mean optimal solution is $\infty$?
Geometric Interpretation

- Feasible solution is a point within the polytope
- The linear programming problem is infeasible if the polytope defined by $A\vec{x} \leq \vec{b}$ is empty
- An LP problem is unbounded if the polytope is open in the direction of the objective function

**Question:** If polytope is not closed, does this mean optimal solution is $\infty$?

No!
Geometric Interpretation

- Feasible solution is a point within the polytope
- The linear programming problem is infeasible if the polytope defined by $A\vec{x} \leq \vec{b}$ is empty
- An LP problem is unbounded if the polytope is open in the direction of the objective function

**Question:** If polytope is not closed, does this mean optimal solution is $\infty$? No!

- Since the polytope defined by $A\vec{x} \leq \vec{b}$ is convex, the optimal solution for bounded LP problem must lie on exterior boundary of polytope
Deciding $T_\mathbb{Q}$ as Linear Program

- How do we determine $T_\mathbb{Q}$ satisfiability using LP?

First, convert $T_\mathbb{Q}$ formula to NNF.

In this form, every atomic formula is of the form:

$$a_1 x_1 + a_2 x_2 + ... + a_n x_n \triangleleft △c \text{ (△∈\{=, \not=, \geq, <\})}$$

First, rewrite it as equisat formula containing only $\leq$ and $>$. 

- \[\vec{a}^T \vec{x} \geq c \Rightarrow -\vec{a}^T \vec{x} \leq -c\]
- \[\vec{a}^T \vec{x} = c \Rightarrow \vec{a}^T \vec{x} \triangleleft c \triangleleft -\vec{a}^T \vec{x} \leq -c\]
- \[\vec{a}^T \vec{x} + y \leq c \land y > 0\]
- \[\vec{a}^T \vec{x} = c \Rightarrow \vec{a}^T \vec{x} \leq c \land -\vec{a}^T \vec{x} \leq -c\]

---

Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?
- First, convert $T_Q$ formula to NNF.
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?
- First, convert $T_Q$ formula to NNF.
- In this form, every atomic formula is of the form:

  $$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \Join c \quad (\Join \in \{=, \neq, \geq, <\})$$
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?
- First, convert $T_Q$ formula to NNF.
- In this form, every atomic formula is of the form:
  \[ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \cong c \quad (\cong \in \{=, \neq, \geq, <\}) \]
- First, rewrite it as equisat formula containing only $\leq$ and $>$
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?
- First, convert $T_Q$ formula to NNF.
- In this form, every atomic formula is of the form:
  \[ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \natsym{\triangleleft} c \quad (\natsym{\triangleleft} \in \{=, \neq, \geq, <\}) \]
- First, rewrite it as equisat formula containing only $\leq$ and $>$
  \[ \vec{a}^T \vec{x} \geq c \quad \Rightarrow \]
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?
- First, convert $T_Q$ formula to NNF.
- In this form, every atomic formula is of the form:
  $$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \Delta c \quad (\Delta \in \{=, \neq, \geq, <\})$$
- First, rewrite it as equisat formula containing only $\leq$ and $>$
  $$\bar{a}^T \bar{x} \geq c \quad \Rightarrow \quad -\bar{a}^T \bar{x} \leq -c$$
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?

- First, convert $T_Q$ formula to NNF.

- In this form, every atomic formula is of the form:
  \[ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \bowtie c \quad (\bowtie \in \{=, \neq, \geq, <\}) \]

- First, rewrite it as equisat formula containing only $\leq$ and $>$
  \[
  \bar{a}^T \bar{x} \geq c \quad \Rightarrow \quad -\bar{a}^T \bar{x} \leq -c
  \]
  \[
  \bar{a}^T \bar{x} < c \quad \Rightarrow
  \]
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?

- First, convert $T_Q$ formula to NNF.

- In this form, every atomic formula is of the form:

  $$ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \nrightarrow c \quad (\nrightarrow \in \{=, \neq, \geq, <\}) $$

- First, rewrite it as equisat formula containing only $\leq$ and $>$

  $$ \bar{a}^T \bar{x} \geq c \quad \Rightarrow \quad -\bar{a}^T \bar{x} \leq -c $$

  $$ \bar{a}^T \bar{x} < c \quad \Rightarrow \quad \bar{a}^T \bar{x} + y \leq c \wedge y > 0 $$
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?
- First, convert $T_Q$ formula to NNF.
- In this form, every atomic formula is of the form:
  \[ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \bigtriangleup c \quad (\bigtriangleup \in \{=, \neq, \geq, <\}) \]
- First, rewrite it as equisat formula containing only $\leq$ and $>$
  \[
  \begin{align*}
  \bar{a}^T \bar{x} &\geq c \quad \Rightarrow \quad -\bar{a}^T \bar{x} \leq -c \\
  \bar{a}^T \bar{x} &< c \quad \Rightarrow \quad \bar{a}^T \bar{x} + y \leq c \land y > 0 \\
  \bar{a}^T \bar{x} &= c \quad \Rightarrow \quad 
  \end{align*}
\]
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?
- First, convert $T_Q$ formula to NNF.
- In this form, every atomic formula is of the form:
  \[ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \bowtie c \quad (\bowtie \in \{=, \neq, \geq, <\}) \]
- First, rewrite it as equisat formula containing only $\leq$ and $>$
  \[
  \begin{align*}
  \bar{a}^T \bar{x} \geq c & \implies -\bar{a}^T \bar{x} \leq -c \\
  \bar{a}^T \bar{x} < c & \implies \bar{a}^T \bar{x} + y \leq c \land y > 0 \\
  \bar{a}^T \bar{x} = c & \implies \bar{a}^T \bar{x} \leq c \land -\bar{a}^T \bar{x} \leq -c
  \end{align*}
  \]
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?

- First, convert $T_Q$ formula to NNF.

- In this form, every atomic formula is of the form:

  $$a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \triangleright c \quad (\triangleright \in \{=, \neq, \geq, <\})$$

- First, rewrite it as equisat formula containing only $\leq$ and $>$

  $$\begin{align*}
  \vec{a}^T \vec{x} \geq c & \implies -\vec{a}^T \vec{x} \leq -c \\
  \vec{a}^T \vec{x} < c & \implies \vec{a}^T \vec{x} + y \leq c \land y > 0 \\
  \vec{a}^T \vec{x} = c & \implies \vec{a}^T \vec{x} \leq c \land -\vec{a}^T \vec{x} \leq -c \\
  \vec{a}^T \vec{x} \neq c & \implies 
  \end{align*}$$
Deciding $T_Q$ as Linear Program

- How do we determine $T_Q$ satisfiability using LP?
- First, convert $T_Q$ formula to NNF.
- In this form, every atomic formula is of the form:
  
  $$a_1x_1 + a_2x_2 + \ldots + a_nx_n \Box c \quad (\Box \in \{=, \neq, \geq, <\})$$

- First, rewrite it as equisat formula containing only $\leq$ and $>$
  
  $$\bar{a}^T\vec{x} \geq c \quad \Rightarrow \quad -\bar{a}^T\vec{x} \leq -c$$
  
  $$\bar{a}^T\vec{x} < c \quad \Rightarrow \quad \bar{a}^T\vec{x} + y \leq c \land y > 0$$
  
  $$\bar{a}^T\vec{x} = c \quad \Rightarrow \quad \bar{a}^T\vec{x} \leq c \land -\bar{a}^T\vec{x} \leq -c$$
  
  $$\bar{a}^T\vec{x} \neq c \quad \Rightarrow \quad (\bar{a}^T\vec{x} + y \leq c \land y > 0) \lor$$
  
  $$(-\bar{a}^T\vec{x} + y \leq -c \land y > 0)$$
Deciding $T_Q$ as Linear Program, cont

- Current formula in NNF and no negations
Deciding $T_Q$ as Linear Program, cont

- Current formula in NNF and **no negations**
- Each atomic formula is one of three forms:
Deciding $T_\mathbb{Q}$ as Linear Program, cont

- Current formula in NNF and no negations
- Each atomic formula is one of three forms:
  1. $a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$
Deciding $T_{\mathbb{Q}}$ as Linear Program, cont

- Current formula in NNF and no negations

- Each atomic formula is one of three forms:
  
  1. $a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$

  2. $\alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i$
Deciding $T_\mathbb{Q}$ as Linear Program, cont

- Current formula in NNF and no negations

- Each atomic formula is one of three forms:
  1. $a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$
  2. $\alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i$
  3. $y > 0$
Deciding $T_\mathbb{Q}$ as Linear Program, cont

- Current formula in NNF and no negations

- Each atomic formula is one of three forms:
  1. $a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$
  2. $\alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i$
  3. $y > 0$

- Next, convert to DNF: Formula is satisfiable iff any of the clauses satisfiable
Deciding $T_\mathbb{Q}$ as Linear Program, cont

- Current formula in NNF and no negations

- Each atomic formula is one of three forms:
  
  1. $a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$
  
  2. $\alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i$
  
  3. $y > 0$

- Next, convert to DNF: Formula is satisfiable iff any of the clauses satisfiable

- Thus, want to formulate each clause as a linear program
Deciding $T_{\mathbb{Q}}$ as Linear Program, cont

- Each clause is of the following form:
  \[
  \land a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \\
  \land \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i \\
  \land y > 0
  \]

- How can we decide whether this constraint is satisfiable by formulating it as an LP problem?
Deciding $T_Q$ as Linear Program, cont

- Each clause is of the following form:

$$\land a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$$
$$\land \land \land \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i$$
$$\land y > 0$$

- How can we decide whether this constraint is satisfiable by formulating it as an LP problem?

- This constraint is satisfiable iff the optimal solution of the following LP problem is strictly positive:

Maximize $y$

Subject to: $\land a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \land \land \land \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i$
Deciding $T_\mathbb{Q}$ as Linear Program, cont

- Each clause is of the following form:

$$\land a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$$
$$\land \land \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i$$
$$\land y > 0$$

- How can we decide whether this constraint is satisfiable by formulating it as an LP problem?

- This constraint is satisfiable iff the optimal solution of the following LP problem is strictly positive:

  Maximize $y$
  Subject to: $\land a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \land \land \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i$

- Why?
Deciding $T_Q$ as Linear Program, cont

- Each clause is of the following form:

\[
\begin{align*}
\bigwedge a_{i1}x_1 + \ldots + a_{in}x_n & \leq b_i \\
\land \bigwedge \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y & \leq \beta_i \\
\land y & > 0
\end{align*}
\]

- How can we decide whether this constraint is satisfiable by formulating it as an LP problem?

- This constraint is satisfiable iff the optimal solution of the following LP problem is strictly positive:

Maximize $y$

Subject to: 
\[
\bigwedge a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \land \bigwedge \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + y \leq \beta_i
\]

- Why? If maximum value of $y$ positive, we know $y > 0$ can be satisfied. If maximum value is $\leq 0$, $y > 0$ cannot be satisfied.
Satisfiability as Linear Programming

Thus, we can formulate satisfiability of every qff conjunctive $T_Q$ formula as a linear programming problem.
Satisfiability as Linear Programming

- Thus, we can formulate satisfiability of every qff conjunctive $T_Q$ formula as a linear programming problem.

- Hence, we’ll focus on how to solve LP problems
Satisfiability as Linear Programming

- Thus, we can formulate satisfiability of every qff conjunctive $T_Q$ formula as a linear programming problem.

- Hence, we’ll focus on how to solve LP problems

- Three popular methods for solving LP problems:
Satisfiability as Linear Programming

- Thus, we can formulate satisfiability of every qff conjunctive $T_Q$ formula as a linear programming problem.

- Hence, we’ll focus on how to solve LP problems

- Three popular methods for solving LP problems:
  1. Ellipsoid method (Khachian, 1979)
Satisfiability as Linear Programming

Thus, we can formulate satisfiability of every qff conjunctive $T_Q$ formula as a linear programming problem.

Hence, we’ll focus on how to solve LP problems

Three popular methods for solving LP problems:

1. Ellipsoid method (Khachian, 1979)

2. Interior-point algorithm (Karmarkar, 1984)
Thus, we can formulate satisfiability of every qff conjunctive $T_Q$ formula as a linear programming problem.

Hence, we’ll focus on how to solve LP problems.

Three popular methods for solving LP problems:

1. Ellipsoid method (Khachian, 1979)
2. Interior-point algorithm (Karmarkar, 1984)
3. Simplex algorithm (Dantzig, 1949)
Satisfiability as Linear Programming

- Thus, we can formulate satisfiability of every qff conjunctive $T_Q$ formula as a linear programming problem.

- Hence, we’ll focus on how to solve LP problems

- Three popular methods for solving LP problems:
  1. Ellipsoid method (Khachian, 1979)
  2. Interior-point algorithm (Karmarkar, 1984)
  3. Simplex algorithm (Dantzig, 1949)

- Among these, ellipsoid and interior-point method are polynomial-time, but Simplex is worst-case exponential
Satisfiability as Linear Programming

- Thus, we can formulate satisfiability of every qff conjunctive $T_Q$ formula as a linear programming problem.

- Hence, we’ll focus on how to solve LP problems.

- Three popular methods for solving LP problems:
  1. Ellipsoid method (Khachian, 1979)
  2. Interior-point algorithm (Karmarkar, 1984)
  3. Simplex algorithm (Dantzig, 1949)

- Among these, ellipsoid and interior-point method are polynomial-time, but Simplex is worst-case exponential.

- Despite this, Simplex remains most popular and performs better for most problems of interest.
Prerequisites for Simplex

- To apply Simplex, we have to transform linear inequality system into **standard form** and then into **slack form**
Prerequisites for Simplex

▶ To apply Simplex, we have to transform linear inequality system into standard form and then into slack form

▶ Standard form:

Maximize $\vec{c}^T \vec{x}$

Subject to:

\[ A\vec{x} \leq \vec{b} \]
\[ \vec{x} \geq 0 \]
Prerequisites for Simplex

- To apply Simplex, we have to transform linear inequality system into **standard form** and then into **slack form**

- **Standard form:**

  Maximize $\vec{c}^T \vec{x}$
  
  Subject to:

  $A\vec{x} \leq \vec{b}$

  $\vec{x} \geq 0$
Prerequisites for Simplex

- To apply Simplex, we have to transform linear inequality system into standard form and then into slack form

- Standard form:

  \[
  \text{Maximize } \tilde{c}^T \tilde{x} \\
  \text{Subject to: } A\tilde{x} \leq \tilde{b} \\
  \tilde{x} \geq 0
  \]

- Good news: We can convert every LP problem into an equisatisfiable standard form representation
Prerequisites for Simplex

- To apply Simplex, we have to transform linear inequality system into standard form and then into slack form

- **Standard form:**
  
  Maximize $\vec{c}^T \vec{x}$
  
  Subject to:
  
  $A\vec{x} \leq \vec{b}$
  
  $\vec{x} \geq 0$

- **Good news:** We can convert every LP problem into an *equisatisfiable* standard form representation

- **Equisat.** means original problem has optimal objective value $c$ iff problem in standard form has optimal objective value $c$
Main idea: Any negative variable can be written as difference of two non-negative integers.
Conversion to Standard Form

- **Main idea**: Any negative variable can be written as difference of two non-negative integers

- Suppose variable $x_i$ does not have non-negativity constraint
Conversion to Standard Form

- **Main idea:** Any negative variable can be written as difference of two non-negative integers

- Suppose variable $x_i$ does not have non-negativity constraint

- For each such variable, introduce two new variables $x'_i$ and $x''_i$
Conversion to Standard Form

- **Main idea**: Any negative variable can be written as difference of two non-negative integers

- Suppose variable $x_i$ does not have non-negativity constraint

- For each such variable, introduce two new variables $x'_i$ and $x''_i$

- Add non-negativity constraints: $x'_i \geq 0$ and $x''_i \geq 0$
Conversion to Standard Form

- **Main idea:** Any negative variable can be written as difference of two non-negative integers

- Suppose variable $x_i$ does not have non-negativity constraint

- For each such variable, introduce two new variables $x'_i$ and $x''_i$

- Add non-negativity constraints: $x'_i \geq 0$ and $x''_i \geq 0$

- Express $x_i$ as $x'_i - x''_i$ by substituting $x'_i - x''_i$ for each occurrence of $x_i$
Conversion to Standard Form

- **Main idea:** Any negative variable can be written as difference of two non-negative integers

- Suppose variable $x_i$ does not have non-negativity constraint

- For each such variable, introduce two new variables $x'_i$ and $x''_i$

- Add non-negativity constraints: $x'_i \geq 0$ and $x''_i \geq 0$

- Express $x_i$ as $x'_i - x''_i$ by substituting $x'_i - x''_i$ for each occurrence of $x_i$

- **Observe:** Although $x'_i$ and $x''_i$ are non-negative, $x'_i - x''_i$ can be negative
Conversion to Standard Form

- **Main idea:** Any negative variable can be written as difference of two non-negative integers

- Suppose variable $x_i$ does not have non-negativity constraint

- For each such variable, introduce two new variables $x'_i$ and $x''_i$

- Add non-negativity constraints: $x'_i \geq 0$ and $x''_i \geq 0$

- Express $x_i$ as $x'_i - x''_i$ by substituting $x'_i - x''_i$ for each occurrence of $x_i$

- **Observe:** Although $x'_i$ and $x''_i$ are non-negative, $x'_i - x''_i$ can be negative

- Thus, transformation yields equisatisfiable linear program and is in standard form
Consider the following linear program:

Maximize \( 2x_1 - 3x_2 \)

Subject to:
\[
\begin{align*}
  x_1 + x_2 & \leq 7 \\
  -x_1 - x_2 & \leq -7 \\
  x_1 - 2x_2 & \leq 4 \\
  x_1 & \geq 0
\end{align*}
\]

Variable \( x_2 \) does not have non-negativity constraint; thus rewrite it as \( x_2' - x_2'' \).

Equisatisfiable system in standard form:

Maximize \( 2x_1 - 3x_2' + 3x_2'' \)

Subject to:
\[
\begin{align*}
  x_1 + x_2' - x_2'' & \leq 7 \\
  -x_1 - x_2' + x_2'' & \leq -7 \\
  x_1 - 2x_2' + 2x_2'' & \leq 4 \\
  x_1, x_2', x_2'' & \geq 0
\end{align*}
\]
Consider the following linear program:

Maximize \( 2x_1 - 3x_2 \)
Subject to:
\[
\begin{align*}
x_1 + x_2 & \leq 7 \\
-x_1 - x_2 & \leq -7 \\
x_1 - 2x_2 & \leq 4 \\
x_1 & \geq 0
\end{align*}
\]

Variable \( x_2 \) does not have non-negativity constraint; thus rewrite it as \( x'_2 - x''_2 \)
Consider the following linear program:

Maximize \[ 2x_1 - 3x_2 \]
Subject to:
\[
\begin{align*}
& x_1 + x_2 \leq 7 \\
& -x_1 - x_2 \leq -7 \\
& x_1 - 2x_2 \leq 4 \\
& x_1 \geq 0
\end{align*}
\]

Variable \( x_2 \) does not have non-negativity constraint; thus rewrite it as \( x'_2 - x''_2 \).

Equisatisfiable system in standard form:

Maximize \[ 2x_1 - 3x'_2 + 3x''_2 \]
Subject to:
\[
\begin{align*}
& x_1 + x'_2 - x''_2 \leq 7 \\
& -x_1 - x'_2 + x''_2 \leq -7 \\
& x_1 - 2x'_2 + 2x''_2 \leq 4 \\
& x_1, x'_2, x''_2 \geq 0
\end{align*}
\]
Conversion to Slack Form

- To apply Simplex, we need inequalities to be in slack form.
Conversion to Slack Form

- To apply Simplex, we need inequalities to be in slack form.

- In slack form, we only have equalities; the only inequality allowed is non-negativity constraints.
Conversion to Slack Form

- To apply Simplex, we need inequalities to be in **slack form**

- In slack form, we only have equalities; the only inequality allowed is non-negativity constraints

- For each inequality $A_i \vec{x} \leq b_i$, introduce a new **slack variable** $s_i$
Conversion to Slack Form

- To apply Simplex, we need inequalities to be in **slack form**

- In slack form, we only have equalities; the only inequality allowed is non-negativity constraints

- For each inequality $A_i \vec{x} \leq b_i$, introduce a new **slack variable** $s_i$

- Slack variables measure the difference (i.e., "slack") between left-hand and right-hand side
Conversion to Slack Form

- To apply Simplex, we need inequalities to be in **slack form**

- In slack form, we only have equalities; the only inequality allowed is non-negativity constraints

- For each inequality $A_i \vec{x} \leq b_i$, introduce a new slack variable $s_i$

- Slack variables measure the difference (i.e., "slack") between left-hand and right-hand side

- Rewrite inequality as equality $s_i = b_i - A_i \vec{x}$ and introduce non-negativity constraint $s_i \geq 0$
Conversion to Slack Form

- To apply Simplex, we need inequalities to be in *slack form*

- In slack form, we only have equalities; the only inequality allowed is non-negativity constraints

- For each inequality $A_i \vec{x} \leq b_i$, introduce a new *slack variable* $s_i$

- Slack variables measure the difference (i.e., "slack") between left-hand and right-hand side

- Rewrite inequality as equality $s_i = b_i - A_i \vec{x}$ and introduce non-negativity constraint $s_i \geq 0$

- New LP problem is equisatisfiable to the original one and in slack form
Consider LP problem from previous example:

Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
Subject to:
\[ x_1 + x_2 - x_3 \leq 7 \]
\[ -x_1 - x_2 + x_3 \leq -7 \]
\[ x_1 - 2x_2 + 2x_3 \leq 4 \]
\[ x_1, x_2, x_3 \geq 0 \]
Slack Form Conversion Example

- Consider LP problem from previous example:

  Maximize \( 2x_1 - 3x_2 + 3x_3 \)

  Subject to:

  \[
  x_1 + x_2 - x_3 \leq 7 \\
  -x_1 - x_2 + x_3 \leq -7 \\
  x_1 - 2x_2 + 2x_3 \leq 4 \\
  x_1, x_2, x_3 \geq 0
  \]

- In slack form:

  Maximize \( 2x_1 - 3x_2 + 3x_3 \)

  Subject to:
Consider LP problem from previous example:

Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
Subject to:
\[ x_1 + x_2 - x_3 \leq 7 \]
\[ -x_1 - x_2 + x_3 \leq -7 \]
\[ x_1 - 2x_2 + 2x_3 \leq 4 \]
\[ x_1, x_2, x_3 \geq 0 \]

In slack form:

Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
Subject to:
\[ x_4 = 7 - x_1 - x_2 + x_3 \]
Slack Form Conversion Example

- Consider LP problem from previous example:

  Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
  Subject to:
  \[ x_1 + x_2 - x_3 \leq 7 \]
  \[ -x_1 - x_2 + x_3 \leq -7 \]
  \[ x_1 - 2x_2 + 2x_3 \leq 4 \]
  \[ x_1, x_2, x_3 \geq 0 \]

- In slack form:

  Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
  Subject to:
  \[ x_4 = 7 - x_1 - x_2 + x_3 \]
  \[ x_5 = -7 + x_1 + x_2 - x_3 \]
Consider LP problem from previous example:

Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
Subject to:
\[ x_1 + x_2 - x_3 \leq 7 \]
\[ -x_1 - x_2 + x_3 \leq -7 \]
\[ x_1 - 2x_2 + 2x_3 \leq 4 \]
\[ x_1, x_2, x_3 \geq 0 \]

In slack form:

Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
Subject to:
\[ x_4 = 7 - x_1 - x_2 + x_3 \]
\[ x_5 = -7 + x_1 + x_2 - x_3 \]
\[ x_6 = 4 - x_1 + 2x_2 - 2x_3 \]
Consider LP problem from previous example:

Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
Subject to:
\[ x_1 + x_2 - x_3 \leq 7 \]
\[ -x_1 - x_2 + x_3 \leq -7 \]
\[ x_1 - 2x_2 + 2x_3 \leq 4 \]
\[ x_1, x_2, x_3 \geq 0 \]

In slack form:

Maximize \[ 2x_1 - 3x_2 + 3x_3 \]
Subject to:
\[ x_4 = 7 - x_1 - x_2 + x_3 \]
\[ x_5 = -7 + x_1 + x_2 - x_3 \]
\[ x_6 = 4 - x_1 + 2x_2 - 2x_3 \]
\[ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \]
Basic and Non-Basic Variables

- In slack form, there is exactly one variable on the left hand side of equalities
Basic and Non-Basic Variables

- In slack form, there is exactly one variable on the left hand side of equalities
- Variables appearing on the left-hand side called basic variables
Basic and Non-Basic Variables

- In slack form, there is exactly one variable on the left hand side of equalities

- Variables appearing on the left-hand side called basic variables

- Variables appearing on RHS called non-basic variables
Basic and Non-Basic Variables

- In slack form, there is exactly one variable on the left hand side of equalities

- Variables appearing on the left-hand side called basic variables

- Variables appearing on RHS called non-basic variables

- Invariant: Only non-basic variables can appear in the objective function
Basic and Non-Basic Variables

- In slack form, there is exactly one variable on the left hand side of equalities

- Variables appearing on the left-hand side called **basic variables**

- Variables appearing on RHS called **non-basic variables**

- **Invariant:** Only non-basic variables can appear in the objective function

- Initially, all basic variables are slack variables, but this will change as algorithm proceeds
We’ll denote the set of basic variables by $B$ and non-basic variables by $N$. 
Slack Form: Summary

- We’ll denote the set of basic variables by $B$ and non-basic variables by $N$.

- Then we’ll write the slack form as a set of equations of the following form:

$$z = v + \sum_{x_j \in N} c_j x_j \quad \text{(objective function)}$$
We’ll denote the set of basic variables by $B$ and non-basic variables by $N$.

Then we’ll write the slack form as a set of equations of the following form:

$$z = v + \sum_{x_j \in N} c_j x_j \quad (\text{objective function})$$

$$x_i = b_i - \sum_{x_j \in N} a_{ij} x_j \quad (\text{for every } x_i \in B)$$
We’ll denote the set of basic variables by $B$ and non-basic variables by $N$.

Then we’ll write the slack form as a set of equations of the following form:

$$
\begin{align*}
z &= v + \sum_{x_j \in N} c_j x_j \quad \text{(objective function)} \\
x_i &= b_i - \sum_{x_j \in N} a_{ij} x_j \quad \text{(for every } x_i \in B) 
\end{align*}
$$

There are implicit non-negativity constraints on all variables, but we omit them.
We’ll denote the set of basic variables by $B$ and non-basic variables by $N$.

Then we’ll write the slack form as a set of equations of the following form:

$$z = v + \sum_{x_j \in N} c_j x_j \quad \text{(objective function)}$$

$$x_i = b_i - \sum_{x_j \in N} a_{ij} x_j \quad \text{(for every } x_i \in B)$$

There are implicit non-negativity constraints on all variables, but we omit them.

**Question:** Given original matrix $A$ is $m \times n$, what is $|B|$?
Slack Form: Summary

- We’ll denote the set of basic variables by $B$ and non-basic variables by $N$.

- Then we’ll write the slack form as a set of equations of the following form:

  $$ z = v + \sum_{x_j \in N} c_j x_j \quad \text{(objective function)} $$

  $$ x_i = b_i - \sum_{x_j \in N} a_{ij} x_j \quad \text{(for every } x_i \in B) $$

- There are implicit non-negativity constraints on all variables, but we omit them.

- Question: Given original matrix $A$ is $m \times n$, what is $|B|$? $m$
Basic Solution

- For each LP problem in slack form, there is a basic solution

\[ \begin{align*}
  z &= 3x_1 + x_2 + 2x_3 \\
  x_4 &= 30 - x_1 - x_2 - 3x_3 \\
  x_5 &= 24 - 2x_1 - 2x_2 - 5x_3 \\
  x_6 &= 36 - 4x_1 - x_2 - 2x_3
\end{align*} \]

Basic solution called feasible basic solution if it doesn't violate non-negativity constraints.
Basic Solution

- For each LP problem in slack form, there is a **basic solution**
- To obtain basic solution, set all non-basic variables to zero

\[ z = 3x_1 + x_2 + 2x_3 \]
\[ x_4 = 30 - x_1 - x_2 - 3x_3 \]
\[ x_5 = 24 - 2x_1 - 2x_2 - 5x_3 \]
\[ x_6 = 36 - 4x_1 - x_2 - 2x_3 \]
Basic Solution

- For each LP problem in slack form, there is a basic solution.
- To obtain basic solution, set all non-basic variables to zero.
- Compute values of basic variables on the left-hand side.
Basic Solution

- For each LP problem in slack form, there is a basic solution.
- To obtain basic solution, set all non-basic variables to zero.
- Compute values of basic variables on the left-hand side.
- What is basic solution for this slack form?

\[
\begin{align*}
z & = 3x_1 + x_2 + 2x_3 \\
x_4 & = 30 - x_1 - x_2 - 3x_3 \\
x_5 & = 24 - 2x_1 - 2x_2 - 5x_3 \\
x_6 & = 36 - 4x_1 - x_2 - 2x_3
\end{align*}
\]
Basic Solution

- For each LP problem in slack form, there is a basic solution
- To obtain basic solution, set all non-basic variables to zero
- Compute values of basic variables on the left-hand side

What is basic solution for this slack form? \( (0, 0, 0, 30, 24, 36) \)

\[
\begin{align*}
z &= 3x_1 + x_2 + 2x_3 \\
x_4 &= 30 - x_1 - x_2 - 3x_3 \\
x_5 &= 24 - 2x_1 - 2x_2 - 5x_3 \\
x_6 &= 36 - 4x_1 - x_2 - 2x_3
\end{align*}
\]
Basic Solution

- For each LP problem in slack form, there is a basic solution.
- To obtain basic solution, set all non-basic variables to zero.
- Compute values of basic variables on the left-hand side.
- What is basic solution for this slack form? \((0, 0, 0, 30, 24, 36)\)

\[
\begin{align*}
z & = 3x_1 + x_2 + 2x_3 \\
x_4 & = 30 - x_1 - x_2 - 3x_3 \\
x_5 & = 24 - 2x_1 - 2x_2 - 5x_3 \\
x_6 & = 36 - 4x_1 - x_2 - 2x_3
\end{align*}
\]

- Basic solution called feasible basic solution if it doesn’t violate non-negativity constraints.
Simplex Algorithm Phases

- Simplex algorithm has two phases:
  1. Phase I: Compute a feasible basic solution, if one exists
  2. Phase II: Optimize value of objective function

Understanding Phase I relies on understanding phase II.

Thus, we’ll talk about Phase II first.
Simplex Algorithm Phases

- Simplex algorithm has two phases:

  1. **Phase I**: Compute a feasible basic solution, if one exists
Simplex Algorithm Phases

- Simplex algorithm has two phases:

  1. Phase I: Compute a feasible basic solution, if one exists

  2. Phase II: Optimize value of objective function
Simplex Algorithm Phases

- Simplex algorithm has two phases:
  1. Phase I: Compute a feasible basic solution, if one exists
  2. Phase II: Optimize value of objective function

- Understanding Phase I relies on understanding phase II
Simplex Algorithm Phases

- Simplex algorithm has two phases:
  
  1. **Phase I**: Compute a feasible basic solution, if one exists
  2. **Phase II**: Optimize value of objective function

- Understanding Phase I relies on understanding phase II

- Thus, we’ll talk about Phase II first
Simplex Algorithm Optimization Phase Overview

- Starting with a feasible basic solution, each iteration rewrites one slack form into an equivalent slack form
Simplex Algorithm Optimization Phase Overview

- Starting with a feasible basic solution, each iteration rewrites one slack form into an equivalent slack form.

- This rewriting is similar to Gaussian elimination: involves pivot operations on matrix.
Simplex Algorithm Optimization Phase Overview

- Starting with a feasible basic solution, each iteration rewrites one slack form into an equivalent slack form.

- This rewriting is similar to Gaussian elimination: involves pivot operations on matrix.

- Geometrically, each iteration of Simplex "walks" from one vertex to an adjacent vertex until it reaches a local maximum.
Simplex Algorithm Optimization Phase Overview

- Starting with a feasible basic solution, each iteration rewrites one slack form into an equivalent slack form.

- This rewriting is similar to Gaussian elimination: involves pivot operations on matrix.

- Geometrically, each iteration of Simplex "walks" from one vertex to an adjacent vertex until it reaches a local maximum.

- By convexity, local optimum is global optimum; thus algorithm can safely stop when local maximum is reached.
Simplex Algorithm Optimization Phase

- When rewriting one slack form to another, goal is to increase value of objective function associated with basic solution
Simplex Algorithm Optimization Phase

- When rewriting one slack form to another, goal is to increase value of objective function associated with basic solution

- Recall: Objective function is $z = v + \sum_{x_j \in N} c_j x_j$
Simplex Algorithm Optimization Phase

- When rewriting one slack form to another, goal is to increase value of objective function associated with basic solution

- **Recall:** Objective function is $z = v + \sum_{x_j \in N} c_j x_j$

- How can we increase value of $z$?
Simplex Algorithm Optimization Phase

- When rewriting one slack form to another, goal is to increase value of objective function associated with basic solution.

- Recall: Objective function is \( z = v + \sum_{x_j \in N} c_j x_j \)

- How can we increase value of \( z \)?

- If there is a term \( c_j x_j \) with positive \( c_j \), we can increase value of \( z \) by increasing \( x_j \)'s value, i.e., by making \( x_j \) a basic variable.
Simplex Algorithm Optimization Phase

- When rewriting one slack form to another, goal is to **increase** value of objective function associated with basic solution

- **Recall**: Objective function is 
  \[ z = v + \sum_{x_j \in N} c_j x_j \]

- How can we increase value of \( z \)?

- If there is a term \( c_j x_j \) with positive \( c_j \), we can increase value of \( z \) by increasing \( x_j \)'s value, i.e., by making \( x_j \) a basic variable

- What if there are no positive \( c_j \)'s?
Simplex Algorithm Optimization Phase

- When rewriting one slack form to another, goal is to increase value of objective function associated with basic solution

- **Recall**: Objective function is $z = v + \sum_{x_j \in N} c_j x_j$

- How can we increase value of $z$?

- If there is a term $c_j x_j$ with positive $c_j$, we can increase value of $z$ by increasing $x_j$’s value, i.e., by making $x_j$ a basic variable

- What if there are no positive $c_j$’s?

- Then, we know we can’t increase value of $z$, thus we are done!
Suppose we can increase objective value, i.e., there exists a term $c_j x_j$ with positive $c_j$.
Suppose we can increase objective value, i.e., there exists a term $c_j x_j$ with positive $c_j$.

We want to increase $x_j$'s value, but is there a limit on how much we can increase $x_j$?
Suppose we can increase objective value, i.e., there exists a term $c_j x_j$ with positive $c_j$.

We want to increase $x_j$’s value, but is there a limit on how much we can increase $x_j$? In general, yes.
Simplex Algorithm Optimization Phase, cont

- Suppose we can increase objective value, i.e., there exists a term $c_j x_j$ with positive $c_j$

- We want to increase $x_j$’s value, but is there a limit on how much we can increase $x_j$? In general, yes

- Consider equality $x_i = b_i - a_{ij} x_j - \ldots$
Simplex Algorithm Optimization Phase, cont

- Suppose we can increase objective value, i.e., there exists a term \( c_j x_j \) with positive \( c_j \)

- We want to increase \( x_j \)’s value, but is there a limit on how much we can increase \( x_j \)? In general, yes

- Consider equality \( x_i = b_i - a_{ij} x_j - \ldots \)

- Observe: If \( a_{ij} \) is positive and we increase \( x_j \) beyond \( \frac{b_i}{a_{ij}} \), \( x_i \) becomes negative and we violate constraints
Suppose we can increase objective value, i.e., there exists a term $c_j x_j$ with positive $c_j$

We want to increase $x_j$’s value, but is there a limit on how much we can increase $x_j$? In general, yes

Consider equality $x_i = b_i - a_{ij} x_j - \ldots$

Observe: If $a_{ij}$ is positive and we increase $x_j$ beyond $\frac{b_i}{a_{ij}}$, $x_i$ becomes negative and we violate constraints

Thus, the amount by which we can increase $x_j$ is limited by the smallest $\frac{b_i}{a_{ij}}$ among all $i$’s
Simplex Algorithm Optimization Phase, cont

- Suppose we can increase objective value, i.e., there exists a term $c_j x_j$ with positive $c_j$

- We want to increase $x_j$'s value, but is there a limit on how much we can increase $x_j$? In general, yes

- Consider equality $x_i = b_i - a_{ij} x_j - \ldots$

- Observe: If $a_{ij}$ is positive and we increase $x_j$ beyond $\frac{b_i}{a_{ij}}$, $x_i$ becomes negative and we violate constraints

- Thus, the amount by which we can increase $x_j$ is limited by the smallest $\frac{b_i}{a_{ij}}$ among all $i$'s

- If there is no positive coefficient $a_{ij}$, we can increase $x_j$ (and thus $z$) without limit $\Rightarrow$ optimal solution $= \infty$
Thus, given term $c_j x_j$ with positive $c_j$ in objective function, we want to increase $x_j$ as much as possible.
Simplex Algorithm Optimization Phase, cont

- Thus, given term $c_j x_j$ with positive $c_j$ in objective function, we want to increase $x_j$ as much as possible.

- To increase $x_j$ as much as possible, we find equality that most severely restricts how much we can increase $x_j$. 

Equality that most severely restricts $x_j$ has following characteristics:

1. $x_j$'s coefficient $a_{ij}$ is positive (otherwise doesn't limit $x_j$).
2. has smallest value of $b_i a_{ij}$ (most severely restricting).
Thus, given term $c_j x_j$ with positive $c_j$ in objective function, we want to increase $x_j$ as much as possible.

To increase $x_j$ as much as possible, we find equality that most severely restricts how much we can increase $x_j$.

Equality that most severely restricts $x_j$ has following characteristics:
Thus, given term $c_j x_j$ with positive $c_j$ in objective function, we want to increase $x_j$ as much as possible.

To increase $x_j$ as much as possible, we find equality that most severely restricts how much we can increase $x_j$.

Equality that most severely restricts $x_j$ has following characteristics:

1. $x_j$'s coefficient $a_{ij}$ is positive (otherwise doesn't limit $x_j$).
Thus, given term $c_j x_j$ with positive $c_j$ in objective function, we want to increase $x_j$ as much as possible.

To increase $x_j$ as much as possible, we find equality that most severely restricts how much we can increase $x_j$.

Equality that most severely restricts $x_j$ has following characteristics:

1. $x_j$’s coefficient $a_{ij}$ is positive (otherwise doesn’t limit $x_j$)

2. has smallest value of $\frac{b_i}{a_{ij}}$ (most severely restricting)
Simplex Algorithm Optimization Phase, cont

- Suppose equality with basic var. $x_i$ is most restrictive for $x_j$

- Swap roles of $x_i$ and $x_j$ by making $x_j$ basic and $x_i$ non-basic
- To do this, rewrite $x_j$ in terms of $x_i$ and plug this in to all other equations; this operation is called a pivot
- After performing this pivot operation, what is new value of $x_j$?
- Assuming $b_i$ is non-zero, we have increased the value of $x_j$ from 0 to $b_i a_{ij}$
- Thus, after performing pivot we still have feasible solution but objective value is now greater
Suppose equality with basic var. $x_i$ is most restrictive for $x_j$

Swap roles of $x_i$ and $x_j$ by making $x_j$ basic and $x_i$ non-basic
Suppose equality with basic var. $x_i$ is most restrictive for $x_j$

Swap roles of $x_i$ and $x_j$ by making $x_j$ basic and $x_i$ non-basic

To do this, rewrite $x_j$ in terms of $x_i$ and plug this in to all other equations; this operation is called a pivot
Suppose equality with basic var. \( x_i \) is most restrictive for \( x_j \).

Swap roles of \( x_i \) and \( x_j \) by making \( x_j \) basic and \( x_i \) non-basic.

To do this, rewrite \( x_j \) in terms of \( x_i \) and plug this in to all other equations; this operation is called a pivot.

After performing this pivot operation, what is new value of \( x_j \)?
Simplex Algorithm Optimization Phase, cont

- Suppose equality with basic var. $x_i$ is most restrictive for $x_j$

- Swap roles of $x_i$ and $x_j$ by making $x_j$ basic and $x_i$ non-basic

- To do this, rewrite $x_j$ in terms of $x_i$ and plug this in to all other equations; this operation is called a pivot

- After performing this pivot operation, what is new value of $x_j$? $\frac{b_i}{a_{ij}}$
Simplex Algorithm Optimization Phase, cont

- Suppose equality with basic var. $x_i$ is most restrictive for $x_j$

- Swap roles of $x_i$ and $x_j$ by making $x_j$ basic and $x_i$ non-basic

- To do this, rewrite $x_j$ in terms of $x_i$ and plug this in to all other equations; this operation is called a pivot

- After performing this pivot operation, what is new value of $x_j$? $\frac{b_i}{a_{ij}}$

- Assuming $b_i$ is non-zero, we have increased the value of $x_j$ from 0 to $\frac{b_i}{a_{ij}}$
Simplex Algorithm Optimization Phase, cont

- Suppose equality with basic var. \( x_i \) is most restrictive for \( x_j \)

- Swap roles of \( x_i \) and \( x_j \) by making \( x_j \) basic and \( x_i \) non-basic

- To do this, rewrite \( x_j \) in terms of \( x_i \) and plug this in to all other equations; this operation is called a pivot

- After performing this pivot operation, what is new value of \( x_j \)? \( \frac{b_i}{a_{ij}} \)

- Assuming \( b_i \) is non-zero, we have increased the value of \( x_j \) from 0 to \( \frac{b_i}{a_{ij}} \)

- Thus, after performing pivot we still have feasible solution but objective value is now greater
Simplex Optimization Phase Summary

- Pivot operation exchanges a basic variable with a non-basic variable to increase objective value of basic solution.
Simplex Optimization Phase Summary

- Pivot operation exchanges a basic variable with a non-basic variable to increase objective value of basic solution

- Simplex repeats this pivot operation until one of two conditions hold:
Simplex Optimization Phase Summary

- Pivot operation exchanges a basic variable with a non-basic variable to increase objective value of basic solution

- Simplex repeats this pivot operation until one of two conditions hold:
  1. All coefficients in objective function are negative $\Rightarrow$ optimal solution found
Simplex Optimization Phase Summary

- Pivot operation exchanges a basic variable with a non-basic variable to **increase** objective value of basic solution

- Simplex repeats this pivot operation until one of two conditions hold:

1. All coefficients in objective function are **negative** ⇒ optimal solution found

2. There exists a non-basic variable $x_j$ with positive coefficient $c_j$ in objective function, but all coefficients $a_{ij}$ are negative ⇒ optimal solution $= \infty$
Example

\begin{align*}
z & = 3x_1 + x_2 + 2x_3 \\
x_4 & = 30 - x_1 - x_2 - 3x_3 \\
x_5 & = 24 - 2x_1 - 2x_2 - 5x_3 \\
x_6 & = 36 - 4x_1 - x_2 - 2x_3 \\
\end{align*}

▶ How can we increase value of objective function?
Example

\[ z = 3x_1 + x_2 + 2x_3 \]
\[ x_4 = 30 - x_1 - x_2 - 3x_3 \]
\[ x_5 = 24 - 2x_1 - 2x_2 - 5x_3 \]
\[ x_6 = 36 - 4x_1 - x_2 - 2x_3 \]

- How can we increase value of objective function?

- By increasing any of \( x_1, x_2, x_3 \); let’s pick \( x_1 \)
Example

\[ z = 3x_1 + x_2 + 2x_3 \]
\[ x_4 = 30 - x_1 - x_2 - 3x_3 \]
\[ x_5 = 24 - 2x_1 - 2x_2 - 5x_3 \]
\[ x_6 = 36 - 4x_1 - x_2 - 2x_3 \]

- How can we increase value of objective function?
- By increasing any of \( x_1, x_2, x_3 \); let’s pick \( x_1 \)
- Which equality restricts \( x_1 \) the most?
Example

\[ z = 3x_1 + x_2 + 2x_3 \]
\[ x_4 = 30 - x_1 - x_2 - 3x_3 \]
\[ x_5 = 24 - 2x_1 - 2x_2 - 5x_3 \]
\[ x_6 = 36 - 4x_1 - x_2 - 2x_3 \]

- How can we increase value of objective function?
- By increasing any of \( x_1, x_2, x_3 \); let's pick \( x_1 \)
- Which equality restricts \( x_1 \) the most? \( x_6 \)
Example

\[
\begin{align*}
z &= 3x_1 + x_2 + 2x_3 \\
x_4 &= 30 - x_1 - x_2 - 3x_3 \\
x_5 &= 24 - 2x_1 - 2x_2 - 5x_3 \\
x_6 &= 36 - 4x_1 - x_2 - 2x_3
\end{align*}
\]

- How can we increase value of objective function?

- By increasing any of \( x_1, x_2, x_3 \); let’s pick \( x_1 \)

- Which equality restricts \( x_1 \) the most? \( x_6 \)

- Rewrite \( x_1 \) in terms of \( x_6 \):
Example

\[
\begin{align*}
  z &= 3x_1 + x_2 + 2x_3 \\
  x_4 &= 30 - x_1 - x_2 - 3x_3 \\
  x_5 &= 24 - 2x_1 - 2x_2 - 5x_3 \\
  x_6 &= 36 - 4x_1 - x_2 - 2x_3
\end{align*}
\]

- How can we increase value of objective function?
- By increasing any of \(x_1, x_2, x_3\); let’s pick \(x_1\)
- Which equality restricts \(x_1\) the most? \(x_6\)
- Rewrite \(x_1\) in terms of \(x_6\):

\[
x_1 = 9 - \frac{1}{4}x_2 - \frac{1}{2}x_3 - \frac{1}{4}x_6
\]
Example, cont

- Plug this in for $x_1$ in all other equations (i.e., pivot):

\[
\begin{align*}
  z & = 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4} \\
  x_1 & = 9 - \frac{x_2}{4} - \frac{x_3}{4} - \frac{x_6}{4} \\
  x_4 & = 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4} \\
  x_5 & = 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}
\end{align*}
\]
Example, cont

- Plug this in for $x_1$ in all other equations (i.e., pivot):

  
  \[
  \begin{align*}
  z &= 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4} \\
  x_1 &= 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4} \\
  x_4 &= 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4} \\
  x_5 &= 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}
  \end{align*}
  \]

- How can we increase value of $z$?
Plug this in for $x_1$ in all other equations (i.e., pivot):

\[
\begin{align*}
    z &= 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4} \\
    x_1 &= 9 - \frac{x_2}{4} - \frac{x_3}{4} - \frac{x_6}{4} \\
    x_4 &= 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4} \\
    x_5 &= 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}
\end{align*}
\]

How can we increase value of $z$?

Either by increasing $x_2$ or $x_3$, but not $x_6$; let’s pick $x_3$
Example, cont

- Plug this in for $x_1$ in all other equations (i.e., pivot):

  \[
  z = 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4} \\
  x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{4} - \frac{x_6}{4} \\
  x_4 = 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4} \\
  x_5 = 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}
  \]

- How can we increase value of $z$?

- Either by increasing $x_2$ or $x_3$, but not $x_6$; let’s pick $x_3$

- Which equality restricts $x_3$ the most?
Example, cont

- Plug this in for $x_1$ in all other equations (i.e., pivot):

\[
\begin{align*}
  z &= 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4} \\
  x_1 &= 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4} \\
  x_4 &= 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4} \\
  x_5 &= 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}
\end{align*}
\]

- How can we increase value of $z$?

- Either by increasing $x_2$ or $x_3$, but not $x_6$; let’s pick $x_3$

- Which equality restricts $x_3$ the most? $x_5$
Plug this in for $x_1$ in all other equations (i.e., pivot):

\[
\begin{align*}
  z &= 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4} \\
  x_1 &= 9 - \frac{x_2}{4} - \frac{x_3}{4} - \frac{x_6}{4} \\
  x_4 &= 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4} \\
  x_5 &= 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}
\end{align*}
\]

How can we increase value of $z$?

Either by increasing $x_2$ or $x_3$, but not $x_6$; let’s pick $x_3$.

Which equality restricts $x_3$ the most? $x_5$.

What is $x_3$ in terms of $x_5$, $x_2$, $x_6$?
Example, cont

- Plug this in for $x_1$ in all other equations (i.e., pivot):

  \[
  z = 27 + \frac{x_2}{4} + \frac{x_3}{2} - \frac{3x_6}{4} \\
  x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{4} - \frac{x_6}{4} \\
  x_4 = 21 - \frac{3x_2}{4} - \frac{5x_3}{2} + \frac{x_6}{4} \\
  x_5 = 6 - \frac{3x_2}{2} - 4x_3 + \frac{x_6}{2}
  \]

- How can we increase value of $z$?

- Either by increasing $x_2$ or $x_3$, but not $x_6$; let’s pick $x_3$

- Which equality restricts $x_3$ the most? $x_5$

- What is $x_3$ in terms of $x_5$, $x_2$, $x_6$?

  \[
  x_3 = \frac{3}{2} - \frac{3}{8}x_2 - \frac{1}{4}x_5 + \frac{1}{8}x_6
  \]
Example, cont

- New slack form after making $x_3$ basic, $x_5$ non-basic:

\[
\begin{align*}
    z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\
    x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\
    x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{4x_5}{8} + \frac{x_6}{8} \\
    x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16}
\end{align*}
\]
Example, cont

- New slack form after making $x_3$ basic, $x_5$ non-basic:

\[
\begin{align*}
z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\
x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{16x_6}{16} \\
x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} + \frac{x_6}{8} \\
x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16}
\end{align*}
\]

- Can we increase $z$?
Example, cont

- New slack form after making $x_3$ basic, $x_5$ non-basic:

\[
\begin{align*}
z  &= \frac{11}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\
 x_1 &= \frac{3}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\
 x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\
 x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16}
\end{align*}
\]

- Can we increase $z$? Yes, increase $x_2$.
Example, cont

- New slack form after making $x_3$ basic, $x_5$ non-basic:

$$
\begin{align*}
  z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\
  x_1 &= \frac{3}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\
  x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\
  x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16}
\end{align*}
$$

- Can we increase $z$? Yes, increase $x_2$

- Which equality restricts $x_2$ the most?
Example, cont

- New slack form after making $x_3$ basic, $x_5$ non-basic:

\[
\begin{align*}
  z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\
  x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\
  x_3 &= \frac{3}{2} - \frac{x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\
  x_4 &= \frac{69}{4} + \frac{x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16}
\end{align*}
\]

- Can we increase $z$? Yes, increase $x_2$

- Which equality restricts $x_2$ the most?

- $x_4$ does not restrict; $x_2$ restricts by 132, $x_3$ restricts by 4 \(\Rightarrow x_3\)
Example, cont

- New slack form after making $x_3$ basic, $x_5$ non-basic:

\[
\begin{align*}
z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\
x_1 &= \frac{4}{33} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\
x_3 &= \frac{3}{4} - \frac{x_2}{8} - \frac{4}{3x_2} + \frac{x_6}{8} \\
x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16}
\end{align*}
\]

- Can we increase $z$? Yes, increase $x_2$

- Which equality restricts $x_2$ the most?

- $x_4$ does not restrict; $x_2$ restricts by 132, $x_3$ restricts by 4 $\implies x_3$

- Solve $x_2$ in terms of $x_3$: 
Example, cont

- New slack form after making $x_3$ basic, $x_5$ non-basic:

\[
\begin{align*}
    z &= 111 + 3x_2 - 8x_5 - 11x_6 \\
    x_1 &= 33 - 16x_2 + 8x_5 - 16x_6 \\
    x_3 &= 4 - 16x_2 - 8x_5 + 16x_6 \\
    x_4 &= 69 + 16x_2 - 8x_5 - 16x_6 \\
\end{align*}
\]

- Can we increase $z$? Yes, increase $x_2$

- Which equality restricts $x_2$ the most?

- $x_4$ does not restrict; $x_2$ restricts by 132, $x_3$ restricts by 4 $\Rightarrow x_3$

- Solve $x_2$ in terms of $x_3$:

\[
x_2 = 4 - \frac{8}{3}x_3 - \frac{2}{3}x_5 + \frac{1}{3}x_6
\]
Example, cont.

- New slack form after making \( x_2 \) basic, \( x_3 \) non-basic:

\[
\begin{align*}
  z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
  x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{3} - \frac{x_6}{3} \\
  x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
  x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}
\end{align*}
\]
New slack form after making $x_2$ basic, $x_3$ non-basic:

\[
\begin{align*}
z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\
x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}
\end{align*}
\]

Can we increase objective value?
Example, cont.

- New slack form after making $x_2$ basic, $x_3$ non-basic:

\[
\begin{align*}
  z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
  x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{3} - \frac{x_6}{3} \\
  x_2 &= 4 - \frac{6x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
  x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}
\end{align*}
\]

- Can we increase objective value? No, Simplex terminates
Example, cont.

- New slack form after making $x_2$ basic, $x_3$ non-basic:

  \[
  z = 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
  x_1 = 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\
  x_2 = 4 - \frac{6x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
  x_4 = 18 - \frac{x_3}{2} + \frac{x_5}{2}
  \]

- Can we increase objective value? No, Simplex terminates

- What is optimal objective value?
New slack form after making $x_2$ basic, $x_3$ non-basic:

$$z = 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3}$$
$$x_1 = 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3}$$
$$x_2 = 4 - \frac{8x_3}{3} - \frac{6x_5}{3} + \frac{x_6}{3}$$
$$x_4 = 18 - \frac{x_3}{2} + \frac{x_5}{2}$$

Can we increase objective value? No, Simplex terminates

What is optimal objective value? 28
Example, cont.

► New slack form after making $x_2$ basic, $x_3$ non-basic:

\[
\begin{align*}
z & = 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
x_1 & = 8 + \frac{x_3}{3} + \frac{x_5}{3} - \frac{x_6}{3} \\
x_2 & = 4 - \frac{6}{8x_3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
x_4 & = 18 - \frac{x_3}{2} + \frac{x_5}{2}
\end{align*}
\]

► Can we increase objective value? No, Simplex terminates

► What is optimal objective value? 28

► What is optimal solution?
Example, cont.

- New slack form after making $x_2$ basic, $x_3$ non-basic:
  \[
  z = 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3}
  
  x_1 = 8 + \frac{x_3}{6} + \frac{x_5}{3} - \frac{x_6}{3}
  
  x_2 = 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3}
  
  x_4 = 18 - \frac{x_3}{2} + \frac{x_5}{2}
  \]

- Can we increase objective value? No, Simplex terminates

- What is optimal objective value? 28

- What is optimal solution? $(8, 4, 0, 18, 0, 0)$
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$’th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$’th iteration.
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$’th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$’th iteration.

- Is it possible that $c_{n+1} < c_n$?
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$’th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$’th iteration.

- Is it possible that $c_{n+1} < c_n$? No
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$’th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$’th iteration.

- Is it possible that $c_{n+1} < c_n$? No

- Consider objective function at $n$’th iteration: $z = v + \sum c_j x_j$
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$’th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$’th iteration.

- Is it possible that $c_{n+1} < c_n$? No

- Consider objective function at $n$’th iteration: $z = v + \sum c_j x_j$

- What is objective value at $n$’th iteration?
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$'th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$'th iteration.

- Is it possible that $c_{n+1} < c_n$? No

- Consider objective function at $n$'th iteration: $z = v + \sum c_j x_j$

- What is objective value at $n$'th iteration? $v$
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$’th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$’th iteration.

- Is it possible that $c_{n+1} < c_n$? No

- Consider objective function at $n$’th iteration: $z = v + \sum c_j x_j$

- What is objective value at $n$’th iteration? $v$

- Suppose Simplex makes $x_j$ basic variable in next iteration.
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$’th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$’th iteration.

- Is it possible that $c_{n+1} < c_n$? No

- Consider objective function at $n$’th iteration: $z = v + \sum c_j x_j$

- What is objective value at $n$’th iteration? $v$

- Suppose Simplex makes $x_j$ basic variable in next iteration.

- At $n$’th iteration, value of $x_j$ was 0 (since $x_j$ non-basic)

- At $n + 1$’th iteration, $x_j \geq 0$ because we don’t violate non-negativity constraints
Can the Objective Value Decrease?

- Let $c_n$ be the objective value at $n$’th iteration of Simplex, and let $c_{n+1}$ be the objective value at $n + 1$’th iteration.

- Is it possible that $c_{n+1} < c_n$? No

- Consider objective function at $n$’th iteration: $z = v + \sum c_j x_j$

- What is objective value at $n$’th iteration? $v$

- Suppose Simplex makes $x_j$ basic variable in next iteration.

- At $n$’th iteration, value of $x_j$ was 0 (since $x_j$ non-basic)

- At $n + 1$’th iteration, $x_j \geq 0$ because we don’t violate non-negativity constraints

- Thus, Simplex never decreases value of the objective function!
Degenerate Problems

- Objective value can’t decrease; but can it stay the same?

Example: Suppose we make $x_2$ the new basic variable, and the most constraining equality is:

$$x_1 = x_2 + 2x_3 + x_4$$

$x_2$'s old value was 0; what is its new value?

Thus, the objective value does not decrease, but does not increase either!

These kinds of problems where objective value can stay the same after pivoting are called degenerate problems.
Degenerate Problems

- Objective value can’t decrease; but can it stay the same? Yes
 Degenerate Problems

- Objective value can’t decrease; but can it stay the same? Yes

- Example: Suppose we make $x_2$ the new basic variable, and most constraining equality is:

$$x_1 = x_2 + 2x_3 + x_4$$

$x_2$'s old value was 0; what is its new value?

Thus, the objective value does not decrease, but does not increase either!

These kinds of problems where objective value can stay the same after pivoting are called degenerate problems.
Degenerate Problems

- Objective value can’t decrease; but can it stay the same? Yes

- Example: Suppose we make $x_2$ the new basic variable, and most constraining equality is:

  $$x_1 = x_2 + 2x_3 + x_4$$

- $x_2$’s old value was 0; what is its new value?
Degenerate Problems

- Objective value can’t decrease; but can it stay the same? **Yes**

- **Example:** Suppose we make \( x_2 \) the new basic variable, and most constraining equality is:

\[
x_1 = x_2 + 2x_3 + x_4
\]

- \( x_2 \)'s old value was 0; what is its new value? **Also 0**
Objective value can’t decrease; but can it stay the same? Yes

Example: Suppose we make $x_2$ the new basic variable, and most constraining equality is:

$$x_1 = x_2 + 2x_3 + x_4$$

$x_2$’s old value was 0; what is its new value? Also 0

Thus, the objective value does not decrease, but does not increase either!
Degenerate Problems

- Objective value can’t decrease; but can it stay the same? Yes

- **Example:** Suppose we make $x_2$ the new basic variable, and most constraining equality is:

  \[ x_1 = x_2 + 2x_3 + x_4 \]

- $x_2$’s old value was 0; what is its new value? Also 0

- Thus, the objective value does not decrease, but does not increase either!

- These kinds of problems where objective value can stay the same after pivoting are called **degenerate problems**
Degenerate Problems and Termination

- If problem is not degenerate, Simplex guaranteed to terminate for any pivot selection strategy (b/c objective value increases)

- Good news: There are pivot selection strategies for which Simplex is always guaranteed to terminate, even for degenerate problems

- One such strategy is Bland’s rule: If there are multiple variables with positive coefficients in objective function, always choose the variable with smallest index

- Example: If \( z = 2x_1 + 5x_2 - 4x_3 \), Bland’s rule chooses \( x_1 \) as new basic variable since it has smallest index
Degenerate Problems and Termination

- If problem is not degenerate, Simplex guaranteed to terminate for any pivot selection strategy (b/c objective value increases)

- **Bad news:** For degenerate problems, Simplex might not terminate

Example: If 

\[ z = 2x_1 + 5x_2 - 4x_3 \]

Bland’s rule chooses \( x_1 \) as new basic variable since it has smallest index
Degenerate Problems and Termination

- If problem is not degenerate, Simplex guaranteed to terminate for any pivot selection strategy (b/c objective value increases)

- Bad news: For degenerate problems, Simplex might not terminate

- Good news: There are pivot selection strategies for which Simplex is always guaranteed to terminate, even for degenerate problems

Example: If $z = 2x_1 + 5x_2 - 4x_3$, Bland's rule chooses $x_1$ as new basic variable since it has smallest index.
Degenerate Problems and Termination

- If problem is not degenerate, Simplex guaranteed to terminate for any pivot selection strategy (b/c objective value increases)

- **Bad news:** For degenerate problems, Simplex might not terminate

- **Good news:** There are pivot selection strategies for which Simplex is always guaranteed to terminate, even for degenerate problems

- One such strategy is **Bland’s rule:** If there are multiple variables with positive coefficients in objective function, always choose the variable with smallest index
Degenerate Problems and Termination

- If problem is not degenerate, Simplex guaranteed to terminate for any pivot selection strategy (b/c objective value increases)

- **Bad news:** For degenerate problems, Simplex might not terminate

- **Good news:** There are pivot selection strategies for which Simplex is always guaranteed to terminate, even for degenerate problems

- One such strategy is **Bland’s rule:** If there are multiple variables with positive coefficients in objective function, always choose the variable with smallest index

- **Example:** If \( z = 2x_1 + 5x_2 - 4x_3 \), Bland’s rule chooses \( x_1 \) as new basic variable since it has smallest index
Simplex Algorithm Phases

- Simplex algorithm has two phases:

1. **Phase I**: Compute a feasible basic solution, if one exists

2. **Phase II**: Optimize value of objective function
Simplex Algorithm Phases

- Simplex algorithm has two phases:

  1. **Phase I**: Compute a feasible basic solution, if one exists

  2. **Phase II**: Optimize value of objective function

- So far, we talked about the second phase, assuming we already have a feasible basic solution
Simplex Algorithm Phases

- Simplex algorithm has two phases:

1. **Phase I**: Compute a feasible basic solution, if one exists

2. **Phase II**: Optimize value of objective function

- So far, we talked about the second phase, assuming we already have a feasible basic solution

- However, the initial basic solution might not feasible even if the linear program is feasible
Example of Infeasible Initial Basic Solution

Consider the following linear program:

\[
\begin{align*}
  z &= 2x_1 - x_2 \\
  x_3 &= 2 - 2x_1 + x_2 \\
  x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

What is the initial basic solution? (0, 0, 2, -4)

Is this solution feasible? No, violates non-negativity constraints

Goal of Phase I of Simplex is to determine if a feasible basic solution exists, and if so, what it is.
Example of Infeasible Initial Basic Solution

Consider the following linear program:

\[ z = 2x_1 - x_2 \]
\[ x_3 = 2 - 2x_1 + x_2 \]
\[ x_4 = -4 - x_1 + 5x_2 \]

What is the initial basic solution?

No, violates non-negativity constraints

Goal of Phase I of Simplex is to determine if a feasible basic solution exists, and if so, what it is
Example of Infeasible Initial Basic Solution

- Consider the following linear program:
  
  \[
  \begin{align*}
  z &= 2x_1 - x_2 \\
  x_3 &= 2 - 2x_1 + x_2 \\
  x_4 &= -4 - x_1 + 5x_2
  \end{align*}
  \]

- What is the initial basic solution? \((0, 0, 2, -4)\)
Consider the following linear program:

\[
\begin{align*}
    z &= 2x_1 - x_2 \\
    x_3 &= 2 - 2x_1 + x_2 \\
    x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

What is the initial basic solution? (0, 0, 2, -4)

Is this solution feasible?
Example of Infeasible Initial Basic Solution

- Consider the following linear program:

\[
\begin{align*}
    z &= 2x_1 - x_2 \\
    x_3 &= 2 - 2x_1 + x_2 \\
    x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

- What is the initial basic solution? (0, 0, 2, −4)

- Is this solution feasible? No, violates non-negativity constraints
Example of Infeasible Initial Basic Solution

Consider the following linear program:

\[
\begin{align*}
  z &= 2x_1 - x_2 \\
  x_3 &= 2 - 2x_1 + x_2 \\
  x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

What is the initial basic solution? (0, 0, 2, −4)

Is this solution feasible? No, violates non-negativity constraints

Goal of Phase I of Simplex is to determine if a feasible basic solution exists, and if so, what it is
Overview of Phase I

- To find an initial basic solution, we construct an auxiliary linear program $L_{aux}$.
Overview of Phase I

- To find an initial basic solution, we construct an auxiliary linear program $L_{aux}$.

- This auxiliary linear program has the property that we can find a feasible basic solution for it after at most one pivot operation.
Overview of Phase I

- To find an initial basic solution, we construct an auxiliary linear program $L_{aux}$.

- This auxiliary linear program has the property that we can find a feasible basic solution for it after at most one pivot operation.

- Furthermore, original LP problem has a feasible solution if and only if the optimal objective value for $L_{aux}$ is zero.
Overview of Phase I

- To find an initial basic solution, we construct an auxiliary linear program $L_{aux}$.

- This auxiliary linear program has the property that we can find a feasible basic solution for it after at most one pivot operation.

- Furthermore, original LP problem has a feasible solution if and only if the optimal objective value for $L_{aux}$ is zero.

- If optimal value of $L_{aux}$ is 0, we can extract basic feasible solution of original problem from optimal solution to $L_{aux}$. 
Constructing the Auxiliary Linear Program

- Consider the original LP problem:

Maximize \( \sum_{j=1}^{n} c_j x_j \)

Subject to:

\[ \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (i \in [1, m]) \]

\[ x_j \geq 0 \quad (j \in [1, n]) \]
Constructing the Auxiliary Linear Program

Consider the original LP problem:

\[
\begin{align*}
\text{Maximize} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{Subject to:} & \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (i \in [1, m]) \\
& \quad x_j \geq 0 \quad (j \in [0, n])
\end{align*}
\]

This problem is feasible iff the following LP problem \( L_{aux} \) has optimal value 0:

\[
\begin{align*}
\text{Maximize} & \quad -x_0 \\
\text{Subject to:} & \quad \sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i \quad (i \in [1, m]) \\
& \quad x_j \geq 0 \quad (j \in [0, n])
\end{align*}
\]
Justification for Auxiliary LP

Maximize $-x_0$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i \quad (i \in [1, m])$$

$$x_j \geq 0 \quad (j \in [0, n])$$
Justification for Auxiliary LP

Maximize $-x_0$
Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i \quad (i \in [1, m])$$
$$x_j \geq 0 \quad (j \in [0, n])$$

$\Rightarrow$ Suppose $x_0$ has optimal value 0. Then clearly $a_{ij} x_j \leq b_i$ is satisfied for all inequalities
Justification for Auxiliary LP

Maximize $-x_0$
Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i \quad (i \in [1, m])$$
$$x_j \geq 0 \quad (j \in [0, n])$$

$\Rightarrow$ Suppose $x_0$ has optimal value 0. Then clearly $a_{ij} x_j \leq b_i$ is satisfied for all inequalities.

$\Leftarrow (a)$ Suppose original problem has feasible solution $\vec{x}^*$. Then $\vec{x}^*$ combined with $x_0 = 0$ is feasible solution for $L_{aux}$.
Justification for Auxiliary LP

Maximize \(-x_0\)
Subject to:

\[
\sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i \quad (i \in [1, m])
\]

\[
x_j \geq 0 \quad (j \in [0, n])
\]

⇒ Suppose \(x_0\) has optimal value 0. Then clearly \(a_{ij} x_j \leq b_i\) is satisfied for all inequalities

⇐ (a) Suppose original problem has feasible solution \(\vec{x}^*\). Then \(\vec{x}^*\) combined with \(x_0 = 0\) is feasible solution for \(L_{aux}\).

⇐ (b) Due to the non-negativity constraint, \(-x_0\) can be at most 0; thus, this solution is optimal for \(L_{aux}\).
Finding Feasible Basic Solution for $L_{aux}$

- So far, we argued that original problem $L$ has feasible solution iff $L_{aux}$ has optimal value 0.
Finding Feasible Basic Solution for $L_{aux}$

- So far, we argued that original problem $L$ has feasible solution iff $L_{aux}$ has optimal value 0.

- But we still need to figure out how to find feasible basic solution to $L_{aux}$. 
Finding Feasible Basic Solution for $L_{aux}$

- So far, we argued that original problem $L$ has feasible solution iff $L_{aux}$ has optimal value 0.

- But we still need to figure out how to find feasible basic solution to $L_{aux}$.

- Next: We’ll see how we can find feasible basic solution for $L_{aux}$ after one pivot operation.
Auxiliary Problem in Slack Form

\[ z = -x_0 \]
\[ x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j \]
Auxiliary Problem in Slack Form

\[ z = -x_0 \]
\[ x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j \]

- If all \( b_i \)'s are positive, basic solution already feasible
Auxiliary Problem in Slack Form

\[ z = -x_0 \]
\[ x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j \]

- If all \( b_i \)'s are positive, basic solution already feasible
- If there is at least some negative \( b_i \), find equality \( x_i \) with most negative \( b_i \)
Auxiliary Problem in Slack Form

\[ z = -x_0 \]
\[ x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j \]

- If all \( b_i \)'s are positive, basic solution already feasible
- If there is at least some negative \( b_i \), find equality \( x_i \) with most negative \( b_i \)
- Make \( x_0 \) new basic variable, and \( x_i \) non-basic
Auxiliary Problem in Slack Form

\[
\begin{align*}
z & = -x_0 \\
x_i & = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j
\end{align*}
\]

- If all $b_i$’s are positive, basic solution already feasible
- If there is at least some negative $b_i$, find equality $x_i$ with most negative $b_i$
- Make $x_0$ new basic variable, and $x_i$ non-basic
- **Claim:** After this one pivot operation, all $b_i$’s are non-negative; thus basic solution is feasible
Why is This True?

Suppose this equality has most negative $b_i$:

$$x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j$$

$-b_i$ is positive and greater than all other $|b_j|$'s.

Thus, when we plug in equality for $x_0$ into other equations, their new constants will be positive.

Hence, we find a feasible basic solution after at most one pivot step.
Why is This True?

- Suppose this equality has most negative $b_i$:

$$x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j$$

- Rewrite to make $x_0$ basic:

$$x_0 = -b_i + x_i + \sum_{j=1}^{n} a_{ij} x_j$$
Why is This True?

- Suppose this equality has most negative \( b_i \):

\[
x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j
\]

- Rewrite to make \( x_0 \) basic:

\[
x_0 = -b_i + x_i + \sum_{j=1}^{n} a_{ij} x_j
\]

- Now, \(-b_i\) is positive and greater than all other \(|b_j|\)'s
Why is This True?

- Suppose this equality has most negative $b_i$:

$$x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j$$

- Rewrite to make $x_0$ basic:

$$x_0 = -b_i + x_i + \sum_{j=1}^{n} a_{ij} x_j$$

- Now, $-b_i$ is positive and greater than all other $|b_j|$’s

- Thus, when we plug in equality for $x_0$ into other equations, their new constants will be positive
Why is This True?

- Suppose this equality has most negative $b_i$:

\[ x_i = b_i + x_0 - \sum_{j=1}^{n} a_{ij} x_j \]

- Rewrite to make $x_0$ basic:

\[ x_0 = -b_i + x_i + \sum_{j=1}^{n} a_{ij} x_j \]

- Now, $-b_i$ is positive and greater than all other $|b_j|$’s

- Thus, when we plug in equality for $x_0$ into other equations, their new constants will be positive

- Hence, we find a feasible basic solution after at most one pivot step
Example

Consider the following linear program from earlier:

\[
    \begin{align*}
    z &= 2x_1 - x_2 \\
    x_3 &= 2 - 2x_1 + x_2 \\
    x_4 &= -4 - x_1 + 5x_2
    \end{align*}
\]
Example

Consider the following linear program from earlier:

\[
\begin{align*}
z &= 2x_1 - x_2 \\
x_3 &= 2 - 2x_1 + x_2 \\
x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

Construct \( L_{aux} \):

\[
\begin{align*}
z &= 2x_1 - x_2 \\
x_3 &= 2 - 2x_1 + x_2 \\
x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]
Example

- Consider the following linear program from earlier:

  \[
  z = 2x_1 - x_2 \\
  x_3 = 2 - 2x_1 + x_2 \\
  x_4 = -4 - x_1 + 5x_2
  \]

- Construct \( L_{aux} \):

  \[
  z = -x_0
  \]
Example

- Consider the following linear program from earlier:

\[
\begin{align*}
  z &= 2x_1 - x_2 \\
  x_3 &= 2 - 2x_1 + x_2 \\
  x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

- Construct \( L_{aux} \):

\[
\begin{align*}
  z &= -x_0 \\
  x_3 &= 2 + x_0 - 2x_1 + x_2
\end{align*}
\]
Example

- Consider the following linear program from earlier:
  
  \[
  \begin{align*}
  z &= 2x_1 - x_2 \\
  x_3 &= 2 - 2x_1 + x_2 \\
  x_4 &= -4 - x_1 + 5x_2
  \end{align*}
  \]

- Construct \( L_{aux} \):
  
  \[
  \begin{align*}
  z &= -x_0 \\
  x_3 &= 2 + x_0 - 2x_1 + x_2 \\
  x_4 &= -4 + x_0 - x_1 + 5x_2
  \end{align*}
  \]
Consider the following linear program from earlier:

\[
\begin{align*}
  z &= 2x_1 - x_2 \\
  x_3 &= 2 - 2x_1 + x_2 \\
  x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

Construct \( L_{aux} \):

\[
\begin{align*}
  z &= -x_0 \\
  x_3 &= 2 + x_0 - 2x_1 + x_2 \\
  x_4 &= -4 + x_0 - x_1 + 5x_2
\end{align*}
\]

Which equation has most negative constant?
Consider the following linear program from earlier:

\[
\begin{align*}
z &= 2x_1 - x_2 \\
x_3 &= 2 - 2x_1 + x_2 \\
x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

Construct \( L_{aux} \):

\[
\begin{align*}
z &= -x_0 \\
x_3 &= 2 + x_0 - 2x_1 + x_2 \\
x_4 &= -4 + x_0 - x_1 + 5x_2
\end{align*}
\]

Which equation has most negative constant? \( x_4 \)
Example

- Consider the following linear program from earlier:

\[
\begin{align*}
  z &= 2x_1 - x_2 \\
  x_3 &= 2 - 2x_1 + x_2 \\
  x_4 &= -4 - x_1 + 5x_2
\end{align*}
\]

- Construct \( L_{aux} \):

\[
\begin{align*}
  z &= -x_0 \\
  x_3 &= 2 + x_0 - 2x_1 + x_2 \\
  x_4 &= -4 + x_0 - x_1 + 5x_2
\end{align*}
\]

- Which equation has most negative constant? \( x_4 \)

- Swap \( x_4 \) and \( x_0 \):
Example

- Consider the following linear program from earlier:

  \[
  \begin{align*}
  z &= 2x_1 - x_2 \\
  x_3 &= 2 - 2x_1 + x_2 \\
  x_4 &= -4 - x_1 + 5x_2
  \end{align*}
  \]

- Construct \( L_{aux} \):

  \[
  \begin{align*}
  z &= -x_0 \\
  x_3 &= 2 + x_0 - 2x_1 + x_2 \\
  x_4 &= -4 + x_0 - x_1 + 5x_2
  \end{align*}
  \]

- Which equation has most negative constant? \( x_4 \)

- Swap \( x_4 \) and \( x_0 \):

  \[
  x_0 = 4 + x_4 + x_1 - 5x_2
  \]
Example, cont

After pivoting, we obtain the new slack form:

\[
\begin{align*}
z &= -4 - x_4 - x_1 + 5x_2 \\
x_3 &= 6 - x_1 - 4x_2 + x_4 \\
x_0 &= 4 + x_4 + x_1 - 5x_2
\end{align*}
\]
Example, cont

- After pivoting, we obtain the new slack form:

\[
\begin{align*}
 z & = -4 - x_4 - x_1 + 5x_2 \\
 x_3 & = 6 - x_1 - 4x_2 + x_4 \\
 x_0 & = 4 + x_4 + x_1 - 5x_2
\end{align*}
\]

- What is current objective value?
After pivoting, we obtain the new slack form:

\[
\begin{align*}
  z &= -4 - x_4 - x_1 + 5x_2 \\
  x_3 &= 6 - x_1 - 4x_2 + x_4 \\
  x_0 &= 4 + x_4 + x_1 - 5x_2
\end{align*}
\]

What is current objective value? -4
Example, cont

- After pivoting, we obtain the new slack form:

\[
\begin{align*}
  z &= -4 - x_4 - x_1 + 5x_2 \\
  x_3 &= 6 - x_1 - 4x_2 + x_4 \\
  x_0 &= 4 + x_4 + x_1 - 5x_2
\end{align*}
\]

- What is current objective value? -4

- How can we increase it?
After pivoting, we obtain the new slack form:

\[
\begin{align*}
    z &= -4 - x_4 - x_1 + 5x_2 \\
    x_3 &= 6 - x_1 - 4x_2 + x_4 \\
    x_0 &= 4 + x_4 + x_1 - 5x_2
\end{align*}
\]

What is current objective value? -4

How can we increase it? increase \( x_2 \)
Example, cont

- After pivoting, we obtain the new slack form:

\[
\begin{align*}
    z &= -4 - x_4 - x_1 + 5x_2 \\
x_3 &= 6 - x_1 - 4x_2 + x_4 \\
x_0 &= 4 + x_4 + x_1 - 5x_2
\end{align*}
\]

- What is current objective value? -4

- How can we increase it? increase \( x_2 \)

- Which equation constrains \( x_2 \) the most?
After pivoting, we obtain the new slack form:

\[
\begin{align*}
z &= -4 - x_4 - x_1 + 5x_2 \\
x_3 &= 6 - x_1 - 4x_2 + x_4 \\
x_0 &= 4 + x_4 + x_1 - 5x_2
\end{align*}
\]

What is current objective value? -4

How can we increase it? increase \(x_2\)

Which equation constrains \(x_2\) the most? \(x_0\)
Example, cont

- After pivoting, we obtain the new slack form:

\[
\begin{align*}
  z &= -4 - x_4 - x_1 + 5x_2 \\
  x_3 &= 6 - x_1 - 4x_2 + x_4 \\
  x_0 &= 4 + x_4 + x_1 - 5x_2
\end{align*}
\]

- What is current objective value? \(-4\)

- How can we increase it? \(\text{increase } x_2\)

- Which equation constrains \(x_2\) the most? \(x_0\)

- Swap \(x_2\) and \(x_0\):
Example, cont

- After pivoting, we obtain the new slack form:

\[
\begin{align*}
  z &= -4 - x_4 - x_1 + 5x_2 \\
  x_3 &= 6 - x_1 - 4x_2 + x_4 \\
  x_0 &= 4 + x_4 + x_1 - 5x_2
\end{align*}
\]

- What is current objective value? \(-4\)

- How can we increase it? increase \(x_2\)

- Which equation constrains \(x_2\) the most? \(x_0\)

- Swap \(x_2\) and \(x_0\):

\[
x_2 = \frac{4}{5} - \frac{1}{5}x_0 + x_4 + x_1
\]
Example, cont

- After pivoting, new slack form:

\[
\begin{align*}
  z &= -x_0 \\
  x_2 &= \frac{4}{5} - \frac{x_0}{5} - \frac{x_1}{5} + \frac{x_4}{5} \\
  x_3 &= \frac{14}{5} + \frac{4x_0}{5} - \frac{9x_1}{5} + \frac{x_4}{5}
\end{align*}
\]
Example, cont

- After pivoting, new slack form:

\[
\begin{align*}
z & = -x_0 \\
x_2 & = \frac{4}{5} - \frac{x_0}{5} - \frac{x_1}{5} + \frac{x_4}{5} \\
x_3 & = \frac{14}{5} + \frac{4x_0}{5} - \frac{9x_1}{5} + \frac{x_4}{5}
\end{align*}
\]

- Objective function cannot be increased, so we are done!
Example, cont

- After pivoting, new slack form:

\[
\begin{align*}
z &= -x_0 \\
x_2 &= \frac{4}{5} - \frac{x_0}{5} - \frac{x_1}{5} + \frac{x_4}{5} \\
x_3 &= \frac{14}{5} + \frac{4x_0}{5} - \frac{9x_1}{5} + \frac{x_4}{5}
\end{align*}
\]

- Objective function cannot be increased, so we are done!

- In original problem, objective function was \( z = 2x_1 - x_2 \)
Example, cont

- After pivoting, new slack form:
  
  \[
  \begin{align*}
  z &= -x_0 \\
  x_2 &= \frac{4}{5} - \frac{x_0}{5} - \frac{x_1}{5} + \frac{x_4}{5} \\
  x_3 &= \frac{14}{5} + \frac{4x_0}{5} - \frac{9x_1}{5} + \frac{x_4}{5}
  \end{align*}
  \]

- Objective function cannot be increased, so we are done!

- In original problem, objective function was \( z = 2x_1 - x_2 \)

- Since \( x_2 \) is now a basic variable, substitute for \( x_2 \) with RHS:
  
  \[
  z = \frac{-4}{5} + \frac{9x_1}{5} - \frac{x_4}{5}
  \]
Example, cont

- After pivoting, new slack form:

\[
\begin{align*}
z &= -x_0 \\
x_2 &= \frac{4}{5} - \frac{x_0}{5} - \frac{x_1}{5} + \frac{x_4}{5} \\
x_3 &= \frac{14}{5} + \frac{4x_0}{5} - \frac{9x_1}{5} + \frac{x_4}{5}
\end{align*}
\]

- Objective function cannot be increased, so we are done!

- In original problem, objective function was \( z = 2x_1 - x_2 \)

- Since \( x_2 \) is now a basic variable, substitute for \( x_2 \) with RHS:

\[
z = \frac{-4}{5} + \frac{9x_1}{5} - \frac{x_4}{5}
\]

- Thus, Phase I returns the following slack form to Phase II:

\[
\begin{align*}
z &= \frac{-4}{5} + \frac{9x_1}{5} - \frac{x_4}{5} \\
x_2 &= \frac{4}{5} - \frac{x_1}{5} + \frac{x_4}{5} \\
x_3 &= \frac{14}{5} - \frac{9x_1}{5} + \frac{x_4}{5}
\end{align*}
\]
Summary

To solve constraints in $T_Q$ (linear inequalities over rationals), we use Simplex algorithm for LP.
Summary

- To solve constraints in $T_Q$ (linear inequalities over rationals), we use Simplex algorithm for LP

- Simplex has two phases
Summary

- To solve constraints in $T_Q$ (linear inequalities over rationals), we use Simplex algorithm for LP

- Simplex has two phases

- In first phase, we construct slack form such that it has a basic feasible solution
To solve constraints in $T_Q$ (linear inequalities over rationals), we use Simplex algorithm for LP

Simplex has two phases

In first phase, we construct slack form such that it has a basic feasible solution

In second phase, we start with basic feasible solution and rewrite one slack form into equivalent one until objective value can’t increase
To solve constraints in $T_\mathbb{Q}$ (linear inequalities over rationals), we use Simplex algorithm for LP

Simplex has two phases

In first phase, we construct slack form such that it has a basic feasible solution

In second phase, we start with basic feasible solution and rewrite one slack form into equivalent one until objective value can’t increase

Although Simplex is a worst-case exponential, it is more popular than polynomial-time algorithms for LP