CS 473: Algorithms

Ruta Mehta

University of Illinois, Urbana-Champaign

Spring 2021
LP & Strong Duality

Lecture 19
April 15, 2021

Some of the slides are courtesy Prof. Chekuri
Part I

Recall
Properties and the Simplex Algorithm

- Solution at a vertex of the polyhedron $\mathcal{P}$.
- If vertex $v$ is not optimal then it has a neighbor where the objective value improves.
- If the $\mathcal{P}$ in $d$ dimension, then every vertex has exactly $d$ neighboring vertices (almost always).
Properties and the Simplex Algorithm

- Solution at a vertex of the polyhedron $\mathcal{P}$.
- If vertex $v$ is not optimal then it has a neighbor where the objective value improves.
- If the $\mathcal{P}$ in $d$ dimension, then every vertex has exactly $d$ neighboring vertices (almost always).

**Simplex:** Moves from a vertex to its neighboring vertex

**Questions + Answers**

- Which neighbor to move to? **One where objective value increases.**
Properties and the Simplex Algorithm

- Solution at a vertex of the polyhedron $\mathcal{P}$.
- If vertex $v$ is not optimal then it has a neighbor where the objective value improves.
- If the $\mathcal{P}$ in $d$ dimension, then every vertex has exactly $d$ neighboring vertices (almost always).

**Simplex:** Moves from a vertex to its neighboring vertex

Questions + Answers

- Which neighbor to move to? **One where objective value increases.**
- When to stop? **When no neighbor with better objective value.**
Properties and the Simplex Algorithm

- Solution at a vertex of the polyhedron $\mathcal{P}$.
- If vertex $v$ is not optimal then it has a neighbor where the objective value improves.
- If the $\mathcal{P}$ in $d$ dimension, then every vertex has exactly $d$ neighboring vertices (almost always).

Simplex: Moves from a vertex to its neighboring vertex

Questions + Answers

- Which neighbor to move to? One where objective value increases.
- When to stop? When no neighbor with better objective value.
- How much time does it take? At most $d$ neighbors to consider in each step.
Issues

1. **Starting vertex**

2. The linear program could be **infeasible**: No point satisfy the constraints.

3. The linear program could be **unbounded**: Polygon unbounded in the direction of the objective function.
Computing the Starting Vertex

Equivalent to solving another LP!

Find an \( x \) such that \( Ax \leq b \).
If \( b \geq 0 \) then trivial!
Computing the Starting Vertex

Equivalent to solving another LP!

Find an $x$ such that $Ax \leq b$.
If $b \geq 0$ then trivial! $x = 0$. Otherwise.
Computing the Starting Vertex

Equivalent to solving another LP!

Find an $x$ such that $Ax \leq b$.
If $b \geq 0$ then trivial! $x = 0$. Otherwise.

\[
\begin{align*}
\text{min : } & \quad s \\
\text{s.t.} : & \quad \sum_j a_{ij} x_j - s \leq b_i, \quad \forall i \\
& \quad s \geq 0
\end{align*}
\]

Trivial feasible solution:
Computing the Starting Vertex

Equivalent to solving another LP!

Find an \( x \) such that \( Ax \leq b \).

If \( b \geq 0 \) then trivial! \( x = 0 \). Otherwise.

\[
\begin{align*}
\min : & \quad s \\
\text{s.t.} : & \quad \sum_j a_{ij} x_j - s \leq b_i, \quad \forall i \\
& \quad s \geq 0
\end{align*}
\]

Trivial feasible solution: \( x = 0, \ s = |\min_i b_i| \).
Computing the Starting Vertex
Equivalent to solving another LP!

Find an $x$ such that $Ax \leq b$.
If $b \geq 0$ then trivial! $x = 0$. Otherwise.

$$\begin{align*}
\min : & \quad s \\
\text{s.t.} : & \quad \sum_j a_{ij}x_j - s \leq b_i, \quad \forall i \\
& \quad s \geq 0
\end{align*}$$

Trivial feasible solution: $x = 0, s = |\min_i b_i|$.

If $Ax \leq b$ feasible then optimal value of the above LP is $s = 0$. 
Computing the Starting Vertex

Equivalent to solving another LP!

Find an $x$ such that $Ax \leq b$.
If $b \geq 0$ then trivial! $x = 0$. Otherwise.

$$
\begin{align*}
\min & : & s \\
\text{s.t.} & : & \sum_j a_{ij} x_j - s \leq b_i, & \forall i \\
& : & s \geq 0
\end{align*}
$$

Trivial feasible solution: $x = 0$, $s = |\min_i b_i|$.

If $Ax \leq b$ feasible then optimal value of the above LP is $s = 0$.

Checks Feasibility!
Part II

Duality
Consider the program

\[
\begin{align*}
\text{maximize} & \quad 4x_1 + 2x_2 \\
\text{subject to} & \quad x_1 + 3x_2 \leq 5 \\
& \quad 2x_1 - 4x_2 \leq 10 \\
& \quad x_1 + x_2 \leq 7 \\
& \quad x_1 \leq 5
\end{align*}
\]
Consider the program

\[
\begin{align*}
\text{maximize} & \quad 4x_1 + 2x_2 \\
\text{subject to} & \quad x_1 + 3x_2 \leq 5 \\
& \quad 2x_1 - 4x_2 \leq 10 \\
& \quad x_1 + x_2 \leq 7 \\
& \quad x_1 \leq 5
\end{align*}
\]

\((0, 1)\) satisfies all the constraints and gives value 2 for the objective function.
Consider the program

\[
\begin{align*}
\text{maximize} & \quad 4x_1 + 2x_2 \\
\text{subject to} & \quad x_1 + 3x_2 \leq 5 \\
& \quad 2x_1 - 4x_2 \leq 10 \\
& \quad x_1 + x_2 \leq 7 \\
& \quad x_1 \leq 5
\end{align*}
\]

1. \((0, 1)\) satisfies all the constraints and gives value 2 for the objective function.

2. Thus, optimal value \(\sigma^*\) is at least 2.
Consider the program

\[
\begin{align*}
\text{maximize} & \quad 4x_1 + 2x_2 \\
\text{subject to} & \quad x_1 + 3x_2 \leq 5 \\
& \quad 2x_1 - 4x_2 \leq 10 \\
& \quad x_1 + x_2 \leq 7 \\
& \quad x_1 \leq 5
\end{align*}
\]

1. \((0, 1)\) satisfies all the constraints and gives value 2 for the objective function.

2. Thus, optimal value \(\sigma^*\) is at least 2.

3. \((2, 0)\) also feasible, and gives a better bound of 8.
Consider the program

\[
\begin{align*}
\text{maximize} & \quad 4x_1 + 2x_2 \\
\text{subject to} & \quad x_1 + 3x_2 \leq 5 \\
& \quad 2x_1 - 4x_2 \leq 10 \\
& \quad x_1 + x_2 \leq 7 \\
& \quad x_1 \leq 5
\end{align*}
\]

1. \((0, 1)\) satisfies all the constraints and gives value 2 for the objective function.

2. Thus, optimal value \(\sigma^*\) is at least 2.

3. \((2, 0)\) also feasible, and gives a better bound of 8.

4. How good is 8 when compared with \(\sigma^*\)?
Obtaining Upper Bounds

maximize \[ 4x_1 + 2x_2 \]
subject to \[ \begin{align*}
  x_1 + 3x_2 & \leq 5 \\
  2x_1 - 4x_2 & \leq 10 \\
  x_1 + x_2 & \leq 7 \\
  x_1 & \leq 5
\end{align*} \]

Let us multiply the first constraint by 2 and then add it to the second constraint.
Obtaining Upper Bounds

maximize \quad 4x_1 + 2x_2 \\
subject to \quad x_1 + 3x_2 \leq 5 \\
\quad 2x_1 - 4x_2 \leq 10 \\
\quad x_1 + x_2 \leq 7 \\
\quad x_1 \leq 5 \\

Let us multiply the first constraint by 2 and the second constraint and add it to the new constraint:

\[
2( x_1 + 3x_2 ) \leq 2(5) \\
+ 1( 2x_1 - 4x_2 ) \leq 1(10) \\
\]

\[
\frac{4x_1 + 2x_2 \leq 20}{\text{Thus, 20 is an upper bound on the optimum value!}}
\]
Obtaining Upper Bounds

maximize \[ 4x_1 + 2x_2 \]
subject to
\[ x_1 + 3x_2 \leq 5 \]
\[ 2x_1 - 4x_2 \leq 10 \]
\[ x_1 + x_2 \leq 7 \]
\[ x_1 \leq 5 \]

1. Let us multiply the first constraint by 2 and add it to the second constraint:

\[
2(x_1 + 3x_2) \leq 2(5) \\
+1(2x_1 - 4x_2) \leq 1(10) \\
\hline
4x_1 + 2x_2 \leq 20
\]

2. Thus, 20 is an upper bound on the optimum value!
Generalizing ...

Multiply first equation by $y_1$, second by $y_2$, third by $y_3$ and fourth by $y_4$ ($y_1, y_2, y_3, y_4 \geq 0$) and add

$$
egin{align*}
\quad & y_1(x_1 + 3x_2) & \leq & y_1 (5) \\
+ & y_2(2x_1 - 4x_2) & \leq & y_2 (10) \\
+ & y_3(x_1 + x_2) & \leq & y_3 (7) \\
+ & y_4(x_1) & \leq & y_4 (5) \\
\hline
(y_1 + 2y_2 + y_3 + y_4)x_1 + (3y_1 - 4y_2 + y_3)x_2 \leq & \ldots
\end{align*}
$$
1. Multiply first equation by $y_1$, second by $y_2$, third by $y_3$ and fourth by $y_4$ ($y_1, y_2, y_3, y_4 \geq 0$) and add

\[
\begin{align*}
&y_1( x_1 + 3x_2 ) \leq y_1(5) \\
&+ y_2( 2x_1 - 4x_2 ) \leq y_2(10) \\
&+ y_3( x_1 + x_2 ) \leq y_3(7) \\
&+ y_4( x_1 ) \leq y_4(5)
\end{align*}
\]

\[
(y_1 + 2y_2 + y_3 + y_4)x_1 + (3y_1 - 4y_2 + y_3)x_2 \leq \ldots
\]

2. $5y_1 + 10y_2 + 7y_3 + 5y_4$ is an upper bound,
Generalizing...

1. Multiply first equation by $y_1$, second by $y_2$, third by $y_3$ and fourth by $y_4$ ($y_1, y_2, y_3, y_4 \geq 0$) and add

\[
y_1(x_1 + x_2) + y_2(2x_1 - 4x_2) + y_3(x_1 + x_2) + y_4(x_1) \leq y_1(5) + y_2(10) + y_3(7) + y_4(5)
\]

\[
(y_1 + 2y_2 + y_3 + y_4)x_1 + (3y_1 - 4y_2 + y_3)x_2 \leq \ldots
\]

2. $5y_1 + 10y_2 + 7y_3 + 5y_4$ is an upper bound, provided coefficients of $x_i$ are same as in the objective function $(4x_1 + 2x_2)$,

\[
y_1 + 2y_2 + y_3 + y_4 = 4 \quad 3y_1 - 4y_2 + y_3 = 2
\]
1. Multiply first equation by \( y_1 \), second by \( y_2 \), third by \( y_3 \) and fourth by \( y_4 \) (\( y_1, y_2, y_3, y_4 \geq 0 \)) and add

\[
\begin{align*}
y_1 (x_1 + 3x_2) & \leq y_1 (5) \\
y_2 (2x_1 - 4x_2) & \leq y_2 (10) \\
y_3 (x_1 + x_2) & \leq y_3 (7) \\
y_4 (x_1) & \leq y_4 (5)
\end{align*}
\]

\[
(y_1 + 2y_2 + y_3 + y_4)x_1 + (3y_1 - 4y_2 + y_3)x_2 \leq \ldots
\]

2. \( 5y_1 + 10y_2 + 7y_3 + 5y_4 \) is an upper bound, provided coefficients of \( x_i \) are same as in the objective function \( (4x_1 + 2x_2) \),

\[
y_1 + 2y_2 + y_3 + y_4 = 4 \quad 3y_1 - 4y_2 + y_3 = 2
\]

3. Subject to these constrains, the best upper bound is

\[
\text{min : } 5y_1 + 10y_2 + 7y_3 + 5y_4!
\]
Thus, the optimum value of program

\[
\begin{align*}
\text{maximize} & \quad 4x_1 + 2x_2 \\
\text{subject to} & \quad x_1 + 3x_2 \leq 5 \\
& \quad 2x_1 - 4x_2 \leq 10 \\
& \quad x_1 + x_2 \leq 7 \\
& \quad x_1 \leq 5
\end{align*}
\]

is upper bounded by the optimal value of the program

\[
\begin{align*}
\text{minimize} & \quad 5y_1 + 10y_2 + 7y_3 + 5y_4 \\
\text{subject to} & \quad y_1 + 2y_2 + y_3 + y_4 = 4 \\
& \quad 3y_1 - 4y_2 + y_3 = 2 \\
& \quad y_1, y_2 \geq 0
\end{align*}
\]
Given a linear program $\Pi$ in canonical form

$$\begin{align*}
\text{maximize} & \quad \sum_{j=1}^{d} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{d} a_{ij} x_j \leq b_i \quad i = 1, 2, \ldots n
\end{align*}$$

the dual $\text{Dual}(\Pi)$ is given by

$$\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} b_i y_i \\
\text{subject to} & \quad \sum_{i=1}^{n} y_i a_{ij} = c_j \quad j = 1, 2, \ldots d \\
y_i \geq 0 \quad i = 1, 2, \ldots n
\end{align*}$$
Given a linear program $\Pi$ in canonical form

$$\begin{align*}
\text{maximize} & \quad \sum_{j=1}^{d} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{d} a_{ij} x_j \leq b_i \quad i = 1, 2, \ldots n
\end{align*}$$

the dual $\text{Dual}(\Pi)$ is given by

$$\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} b_i y_i \\
\text{subject to} & \quad \sum_{i=1}^{n} y_i a_{ij} = c_j \quad j = 1, 2, \ldots d \\
y_i & \geq 0 \quad i = 1, 2, \ldots n
\end{align*}$$

**Proposition**

$\text{Dual(Dual(\Pi))}$ is equivalent to $\Pi$
Duality Theorems

**Theorem (Weak Duality)**

If $x'$ is a feasible solution to $\Pi$ and $y'$ is a feasible solution to $\text{Dual}(\Pi)$ then $c \cdot x' \leq y' \cdot b$.

**Theorem (Strong Duality)**

If $x^*$ is an optimal solution to $\Pi$ and $y^*$ is an optimal solution to $\text{Dual}(\Pi)$ then $c \cdot x^* = y^* \cdot b$.

Many applications! Maxflow-Mincut theorem can be deduced from duality.
<table>
<thead>
<tr>
<th>Theorem (Weak Duality)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $x'$ is a feasible solution to $\Pi$ and $y'$ is a feasible solution to $\text{Dual}(\Pi)$ then $c \cdot x' \leq y' \cdot b$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem (Strong Duality)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $x^<em>$ is an optimal solution to $\Pi$ and $y^</em>$ is an optimal solution to $\text{Dual}(\Pi)$ then $c \cdot x^* = y^* \cdot b$.</td>
</tr>
</tbody>
</table>

Many applications! Maxflow-Mincut theorem can be deduced from duality.
Weak Duality

Theorem (Weak Duality)

*If* \( x \) *is a feasible solution to* \( \Pi \) *and* \( y \) *is a feasible solution to* \( \text{Dual}(\Pi) \) *then* \( c \cdot x \leq y \cdot b \).

We already saw the proof by the way we derived it but we will do it again formally.

**Proof.**

Since \( y' \) is feasible in \( \text{Dual}(\Pi) \): \( y' \cdot A = c \)
**Weak Duality**

**Theorem (Weak Duality)**

*If* $\mathbf{x}$ *is a feasible solution to* $\Pi$ *and* $\mathbf{y}$ *is a feasible solution to* $\text{Dual}(\Pi)$ *then* $\mathbf{c} \cdot \mathbf{x} \leq \mathbf{y} \cdot \mathbf{b}$.

We already saw the proof by the way we derived it but we will do it again formally.

**Proof.**

Since $\mathbf{y}'$ is feasible in $\text{Dual}(\Pi)$: $\mathbf{y}' \mathbf{A} = \mathbf{c}$

Therefore $\mathbf{c} \cdot \mathbf{x}' = \mathbf{y}' \mathbf{A} \mathbf{x}'$
Weak Duality

Theorem (Weak Duality)

If $x$ is a feasible solution to $\Pi$ and $y$ is a feasible solution to Dual($\Pi$) then $c \cdot x \leq y \cdot b$.

We already saw the proof by the way we derived it but we will do it again formally.

Proof.

Since $y'$ is feasible in Dual($\Pi$): $y' A = c$

Therefore $c \cdot x' = y' A x'$

Since $x'$ is feasible in $\Pi$, $A x' \leq b$ and hence,

$$c \cdot x' = y' A x' \leq y' \cdot b$$
Strong Duality and Complementary Slackness

maximize: \( c \cdot x \)
subject to: \( Ax \leq b \)

\[ \text{Dual} \]

minimize: \( y \cdot b \)
subject to: \( yA = c \)
\( y \geq 0 \)

Definition (Complementary Slackness)

\( x \) feasible in \( \Pi \) and \( y \) feasible in \( \text{Dual}(\Pi) \), s.t.,
\( \forall i = 1..n, \ y_i > 0 \Rightarrow (Ax)_i = b_i \)
**Strong Duality and Complementary Slackness**

\[
\text{maximize} : \quad c \cdot x \\
\text{subject to} \quad Ax \leq b \\
\text{Dual} \quad \rightarrow \\
\text{minimize} : \quad y \cdot b \\
\text{subject to} \quad yA = c \\
y \geq 0
\]

**Definition (Complementary Slackness)**

\(x\) feasible in \(\Pi\) and \(y\) feasible in \(\text{Dual(}\Pi\text{)}\), s.t.,
\[
\forall i = 1..n, \quad y_i > 0 \Rightarrow (Ax)_i = b_i
\]

**Geometric Interpretation:** \(c\) is in the cone of the normal vectors of the tight hyperplanes at \(x\).
**Strong Duality and Complementary Slackness**

### Definition (Complementary Slackness)

\[ x \text{ feasible in } \Pi \text{ and } y \text{ feasible in } Dual(\Pi), \text{ s.t.,} \]
\[ \forall i = 1..n, \quad y_i > 0 \Rightarrow (Ax)_i = b_i \]

### Theorem

\((x^*, y^*)\) satisfies complementary Slackness if and only if strong duality holds, i.e., \(c \cdot x^* = y^* \cdot b\).  

### Proof.

\[
c \cdot x^* = (y^* A) \cdot x^* \\
= y^* \cdot (Ax^*)
\]

\((\Rightarrow)\)
**Definition (Complementary Slackness)**

\[
x \text{ feasible in } \Pi \text{ and } y \text{ feasible in } \text{Dual}(\Pi), \text{ s.t.,}
\forall i = 1..n, \quad y_i > 0 \Rightarrow (Ax)_i = b_i
\]

**Theorem**

\((x^*, y^*) \text{ satisfies complementary Slackness if and only if strong duality holds, i.e., } c \cdot x^* = y^* \cdot b.\)

**Proof.**

\[
c \cdot x^* = (y^* A) \cdot x^*
\]
\[
= y^* \cdot (Ax^*)
\]
\[
= \sum_{i=1}^{n} y_i^* (Ax^*)_i
\]
\[
= \sum_{i: y_i > 0} y_i^* (Ax^*)_i
\]
Strong Duality and Complementary Slackness

**Definition (Complementary Slackness)**

\[ x \text{ feasible in } \Pi \text{ and } y \text{ feasible in } \text{Dual}(\Pi), \text{ s.t.,} \]
\[ \forall i = 1..n, \quad y_i > 0 \Rightarrow (Ax)_i = b_i \]

**Theorem**

\((x^*, y^*)\) satisfies complementary Slackness if and only if strong duality holds, i.e., \(c \cdot x^* = y^* \cdot b\).

**Proof.**

\[
\begin{align*}
    c \cdot x^* &= (y^* A) \cdot x^* \\
    &= y^* \cdot (Ax^*) \\
    &= \sum_{i=1}^{n} y_i^* (Ax^*)_i \\
    &= \sum_{i:y_i > 0} y_i^* (Ax^*)_i \\
    &= \sum_i y_i^* b_i = y^* \cdot b
\end{align*}
\]
Strong Duality and Complementary Slackness

**Definition (Complementary Slackness)**

$x$ feasible in $\Pi$ and $y$ feasible in $\text{Dual}(\Pi)$, s.t.,

$$\forall i = 1..n, \quad y_i > 0 \Rightarrow (Ax)_i = b_i$$

**Theorem**

$(x^*, y^*)$ satisfies complementary Slackness if and only if strong duality holds, i.e., $c \cdot x^* = y^* \cdot b$.

**Proof.**

$(\Leftarrow)$
**Strong Duality and Complementary Slackness**

**Definition (Complementary Slackness)**

\[ x \text{ feasible in } \Pi \text{ and } y \text{ feasible in } \text{Dual}(\Pi), \text{ s.t.,} \]
\[ \forall i = 1..n, \quad y_i > 0 \implies (Ax)_i = b_i \]

**Theorem**

\((x^*, y^*)\) satisfies complementary Slackness if and only if strong duality holds, i.e., \(c \cdot x^* = y^* \cdot b\).

**Proof.**

\((\leftarrow) \quad \textbf{Exercise}\)
We want

Theorem (Strong Duality)

If \( x^* \) is an optimal solution to \( \Pi \) and \( y^* \) is an optimal solution to \( \text{Dual}(\Pi) \) then \( c \cdot x^* = y^* \cdot b \).

We showed

Theorem

\((x^*, y^*)\) satisfies complementary slackness \(\Leftrightarrow\) Strong duality holds, i.e., \( c \cdot x^* = y^* \cdot b \).
Strong Duality $\equiv$ Complementary Slackness

We want

**Theorem (Strong Duality)**

If $x^*$ is an optimal solution to $\Pi$ and $y^*$ is an optimal solution to $\text{Dual}(\Pi)$ then $c \cdot x^* = y^* \cdot b$.

We showed

**Theorem**

$(x^*, y^*)$ satisfies complementary slackness $\Leftrightarrow$ Strong duality holds, i.e., $c \cdot x^* = y^* \cdot b$.

If $(x^*, y^*)$ optimum $\Rightarrow$ complementary slackness, then done.
Complementary Slackness: Geometric View

\[
\begin{align*}
\text{maximize} & : \quad c \cdot x \\
\text{subject to} & : \quad Ax \leq b \\
\text{Dual} & \quad \rightarrow
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & : \quad y \cdot b \\
\text{subject to} & : \quad yA = c \\
& : \quad y \geq 0
\end{align*}
\]

\( y^* \) satisfies complementary slackness: \( \forall i = 1..n, \quad y_i^* > 0 \Rightarrow (Ax)_i \leq b_i \)
Complementary Slackness: Geometric View

\[
\begin{align*}
\text{maximize : } & \quad c \cdot x \\
\text{subject to } & \quad Ax \leq b \\
\text{Dual } & \quad \rightarrow \\
\text{minimize : } & \quad y \cdot b \\
\text{subject to } & \quad yA = c \\
& \quad y \geq 0
\end{align*}
\]

\( y^* \) satisfies complementary slackness: \( \forall i = 1..n, \quad y_i^* > 0 \Rightarrow (Ax^*)_i = b \\)

Suppose first \( d \) inequalities are tight at \( x^* \).
Complementary Slackness: Geometric View

\[ \begin{align*}
\text{maximize : } & \quad c \cdot x \\
\text{subject to } & \quad Ax \leq b \\
\text{Dual } & \quad \text{minimize : } \quad y \cdot b \\
\text{subject to } & \quad yA = c \\
& \quad y \geq 0
\end{align*} \]

\( y^* \) satisfies complementary slackness: \( \forall i = 1..n, \quad y_i^* > 0 \Rightarrow (Ax_i^*) \)

Suppose first \( d \) inequalities are tight at \( x^* \).

\[ c \] is in the cone of
$x^*$: Optimum vertex.
Optimality implies Complementary Slackness

\( x^* : \) Optimum vertex. First \( d \) inequalities tight at \( x^* \).

\[
Ax^* \leq b \quad \text{splits into} \quad \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}
\]
Optimality implies Complementary Slackness

$x^*$: Optimum vertex. First $d$ inequalities tight at $x^*$.

$$Ax^* \leq b$$ splits into

$$\hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}$$

Suppose $c$ is NOT in the cone of rows of $\hat{A}$. 

Also known as Farkas’ Lemma
Optimality implies Complementary Slackness

$x^*$: Optimum vertex. First $d$ inequalities tight at $x^*$.

$$Ax^* \leq b$$ splits into $$\hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}$$

Suppose $c$ is NOT in the cone of rows of $\hat{A}$.

![Diagram showing hyperplanes and vectors](image-url)
Optimality implies Complementary Slackness

$x^*$: Optimum vertex. First $d$ inequalities tight at $x^*$.

$$Ax^* \leq b \quad \text{splits into} \quad \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}$$

Suppose $c$ is NOT in the cone of rows of $\hat{A}$.

⇒ There exists a hyperplane separating $c$ from the cone.
Optimality implies Complementary Slackness

$x^*$: Optimum vertex. First $d$ inequalities tight at $x^*$.

\[ Ax^* \leq b \quad \text{splits into} \quad \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b} \]

Suppose $c$ is NOT in the cone of rows of $\hat{A}$.

\[ \Rightarrow \text{There exists a hyperplane separating } c \text{ from the cone.} \]

Suppose cone is on the negative side, and $c$ on the positive size. If the $d$ is the normal vector of the hyperplane, then formally,

\[ \hat{A}d < 0, \quad c \cdot d > 0 \]
Optimality implies Complementary Slackness

$x^*$: Optimum vertex. First $d$ inequalities tight at $x^*$.

$Ax^* \leq b$ splits into $\hat{A}x^* = \hat{b}$, $\tilde{A}x^* < \tilde{b}$

Suppose $c$ is NOT in the cone of rows of $\hat{A}$.

$\Rightarrow$ There exists a hyperplane separating $c$ from the cone. Suppose cone is on the negative side, and $c$ on the positive size. If the $d$ is the normal vector of the hyperplane, then formally,

$\hat{A}d < 0, \quad c \cdot d > 0$ Also known as Farkas’ Lemma
Optimality implies Complementary Slackness

$x^*$: Optimum vertex.
Optimality implies Complementary Slackness

\( x^* \): Optimum vertex. First \( d \) inequalities tight at \( x^* \).

\[ Ax^* \leq b \] splits into \( \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b} \]

Suppose \( c \) is NOT in the cone of rows of \( \hat{A} \).

\[ \implies \text{There exists a hyperplane separating } c \text{ from the cone.} \]

\[ \hat{A}d < 0, \quad c \cdot d > 0 \] Also known as Farkas’ Lemma
Optimality implies Complementary Slackness

\(x^*\): Optimum vertex. First \(d\) inequalities tight at \(x^*\).

\[Ax^* \leq b\] splits into \[\hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}\]

Suppose \(c\) is NOT in the cone of rows of \(\hat{A}\).

\[\Rightarrow\] There exists a hyperplane separating \(c\) from the cone.

\[\hat{A}d < 0, \quad c \cdot d > 0\] Also known as Farkas’ Lemma

Choose v. v. tiny \(\epsilon > 0\) such that \[\tilde{A}(x^* + \epsilon d) \leq \tilde{b}\].

\[\hat{A}(x^* + \epsilon d) = \hat{A}x^* + \epsilon \hat{A}d < \hat{b} \Rightarrow\]
Optimality implies Complementary Slackness

\( x^* : \) Optimum vertex. First \( d \) inequalities tight at \( x^* \).

\[
Ax^* \leq b \quad \text{splits into} \quad \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}
\]

Suppose \( c \) is NOT in the cone of rows of \( \hat{A} \).

\( \Rightarrow \) There exists a hyperplane separating \( c \) from the cone.

\[
\hat{A}d < 0, \quad c \cdot d > 0 \quad \text{Also known as Farkas’ Lemma}
\]

Choose v. v. tiny \( \epsilon > 0 \) such that \( \tilde{A}(x^* + \epsilon d) \leq \tilde{b} \).

\[
\hat{A}(x^* + \epsilon d) = \hat{A}x^* + \epsilon \hat{A}d < \hat{b} \quad \Rightarrow \quad (x^* + \epsilon d) \text{ is feasible in } \Pi
\]
Optimality implies Complementary Slackness

\( x^* \): Optimum vertex. First \( d \) inequalities tight at \( x^* \).

\[ Ax^* \leq b \] splits into \( \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b} \)

Suppose \( c \) is NOT in the cone of rows of \( \hat{A} \).

⇒ There exists a hyperplane separating \( c \) from the cone.

\[ \hat{A}d < 0, \quad c \cdot d > 0 \quad \text{Also known as Farkas’ Lemma} \]

Choose v. v. tiny \( \epsilon > 0 \) such that \( \tilde{A}(x^* + \epsilon d) \leq \tilde{b} \).

\[ \hat{A}(x^* + \epsilon d) = \hat{A}x^* + \epsilon \hat{A}d < \hat{b} \quad \Rightarrow \quad (x^* + \epsilon d) \text{ is feasible in } \Pi \]

\[ c \cdot (x^* + \epsilon d) = c \cdot x^* + \epsilon (c \cdot d) \]
**Optimality implies Complementary Slackness**

\( x^* \): Optimum vertex. First \( d \) inequalities tight at \( x^* \).

\[
Ax^* \leq b \quad \text{splits into} \quad \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}
\]

**Suppose \( c \) is NOT in the cone of rows of \( \hat{A} \).**

\( \Rightarrow \) There exists a hyperplane separating \( c \) from the cone.

\[
\hat{A}d < 0, \quad c \cdot d > 0 \quad \text{Also known as Farkas’ Lemma}
\]

Choose \( \epsilon > 0 \) such that \( \tilde{A}(x^* + \epsilon d) \leq \tilde{b} \).

\[
\hat{A}(x^* + \epsilon d) = \hat{A}x^* + \epsilon \hat{A}d < \hat{b} \quad \Rightarrow \quad (x^* + \epsilon d) \text{ is feasible in } \Pi
\]

\[
c \cdot (x^* + \epsilon d) = c \cdot x^* + \epsilon (c \cdot d) > c \cdot x^* \quad \Rightarrow
\]
Optimality implies Complementary Slackness

$x^*$: Optimum vertex. First $d$ inequalities tight at $x^*$.

$$Ax^* \leq b \quad \text{splits into} \quad \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}$$

Suppose $c$ is NOT in the cone of rows of $\hat{A}$.

$$\Rightarrow \text{There exists a hyperplane separating } c \text{ from the cone.}$$

$$\hat{A}d < 0, \quad c \cdot d > 0 \quad \text{Also known as Farkas’ Lemma}$$

Choose v. v. tiny $\epsilon > 0$ such that $\tilde{A}(x^* + \epsilon d) \leq \tilde{b}$.

$$\hat{A}(x^* + \epsilon d) = \hat{A}x^* + \epsilon \hat{A}d < \hat{b} \quad \Rightarrow \quad (x^* + \epsilon d) \text{ is feasible in } \Pi$$

$$c \cdot (x^* + \epsilon d) = c \cdot x^* + \epsilon (c \cdot d) > c \cdot x^* \Rightarrow x^* \text{ is NOT optimum!}$$
Proof of Strong Duality

\( x^* \): Optimum vertex. First \( d \) inequalities tight at \( x^* \).

\[ Ax^* \leq b \text{ splits into } \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b} \]

Then \( c \) IS in the cone of rows of \( \hat{A} \).
Proof of Strong Duality

\(x^*\): Optimum vertex. First \(d\) inequalities tight at \(x^*\).

\[Ax^* \leq b\] splits into

\[\hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}\]

Then \(c\) IS in the cone of rows of \(\hat{A}\).

\(\iff y^*\) feasible in \(\text{Dual}(\Pi)\) such that \((x^*, y^*)\) satisfies complementary slackness.
Proof of Strong Duality

$x^*$: Optimum vertex. First $d$ inequalities tight at $x^*$.

$$Ax^* \leq b$$ splits into $$\hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}$$

Then $c$ IS in the cone of rows of $\hat{A}$.

$\Leftrightarrow y^*$ feasible in $\text{Dual}(\Pi)$ such that $(x^*, y^*)$ satisfies complementary slackness.

$\Leftrightarrow (x^*, y^*)$ satisfies strong duality, $c \cdot x^* = y^* \cdot b$. 

Ruta (UIUC)
**Proof of Strong Duality**

\( x^* \): Optimum vertex. First \( d \) inequalities tight at \( x^* \).

\[ Ax^* \leq b \text{ splits into } \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b} \]

Then \( c \) IS in the cone of rows of \( \hat{A} \).

\( \Leftrightarrow \) \( y^* \) feasible in \( \text{Dual}(\Pi) \) such that \( (x^*, y^*) \) satisfies complementary slackness.

\( \Leftrightarrow (x^*, y^*) \) satisfies strong duality, \( c \cdot x^* = y^* \cdot b \). (implies \( y^* \) optimal in \( \text{Dual}(\Pi) \)).
Proof of Strong Duality

$x^*$: Optimum vertex. First $d$ inequalities tight at $x^*$.

$$Ax^* \leq b \quad \text{splits into} \quad \hat{A}x^* = \hat{b}, \quad \tilde{A}x^* < \tilde{b}$$

Then $c$ IS in the cone of rows of $\hat{A}$.

$\iff y^*$ feasible in $\text{Dual}(\Pi)$ such that $(x^*, y^*)$ satisfies complementary slackness.

$\iff (x^*, y^*)$ satisfies strong duality, $c \cdot x^* = y^* \cdot b$. (implies $y^*$ optimal in $\text{Dual}(\Pi)$).

Theorem (Strong Duality)

*If $x^*$ is an optimal solution to $\Pi$ and $y^*$ is an optimal solution to $\text{Dual}(\Pi)$ then $c \cdot x^* = y^* \cdot b$.***
Duality for another canonical form

Compactly, for the primal LP $\Pi$

\[
\begin{align*}
\text{max} & \quad c \cdot x \\
\text{subject to} & \quad Ax \leq b, \ x \geq 0
\end{align*}
\]

the dual LP is $\text{Dual}(\Pi)$

\[
\begin{align*}
\text{min} & \quad y \cdot b \\
\text{subject to} & \quad yA \geq c, \ y \geq 0
\end{align*}
\]

**Definition (Complementary Slackness)**

$x$ feasible in $\Pi$ and $y$ feasible in $\text{Dual}(\Pi)$, s.t.,

\[
\begin{align*}
\forall i = 1, \ldots, n, \quad y_i > 0 & \implies (Ax)_i = b_i \\
\forall j = 1, \ldots, d, \quad x_j > 0 & \implies (yA)_j = c_j
\end{align*}
\]
### In General...

from Jeff’s notes

<table>
<thead>
<tr>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{max } c \cdot x )</td>
<td>( \text{min } y \cdot b )</td>
</tr>
<tr>
<td>( \sum_j a_{ij} x_j \leq b_i )</td>
<td>( y_i \geq 0 )</td>
</tr>
<tr>
<td>( \sum_j a_{ij} x_j \geq b_i )</td>
<td>( y_i \leq 0 )</td>
</tr>
<tr>
<td>( \sum_j a_{ij} x_j = b_i )</td>
<td>( - )</td>
</tr>
<tr>
<td>( x_j \geq 0 )</td>
<td>( \sum_i y_i a_{ij} \geq c_j )</td>
</tr>
<tr>
<td>( x_j \leq 0 )</td>
<td>( \sum_i y_i a_{ij} \leq c_j )</td>
</tr>
<tr>
<td>( - )</td>
<td>( \sum_i y_i a_{ij} = c_j )</td>
</tr>
<tr>
<td>( x_j = 0 )</td>
<td>( - )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{min } c \cdot x )</td>
<td>( \text{max } y \cdot b )</td>
</tr>
<tr>
<td>( \sum_j a_{ij} x_j \leq b_i )</td>
<td>( y_i \leq 0 )</td>
</tr>
<tr>
<td>( \sum_j a_{ij} x_j \geq b_i )</td>
<td>( y_i \geq 0 )</td>
</tr>
<tr>
<td>( \sum_j a_{ij} x_j = b_i )</td>
<td>( - )</td>
</tr>
<tr>
<td>( x_j \leq 0 )</td>
<td>( \sum_i y_i a_{ij} \geq c_j )</td>
</tr>
<tr>
<td>( x_j \geq 0 )</td>
<td>( \sum_i y_i a_{ij} \leq c_j )</td>
</tr>
<tr>
<td>( - )</td>
<td>( \sum_i y_i a_{ij} = c_j )</td>
</tr>
</tbody>
</table>

**Figure H.4.** Constructing the dual of an arbitrary linear program.
Part III

Examples of Duality
Network flow

$s$-$t$ flow in directed graph $G = (V, E)$ with capacities $c$. Assume for simplicity that no incoming edges into $s$.

$$\text{max} \sum_{(s,v) \in E} x(s, v)$$

$$\sum_{(u,v) \in E} x(u, v) - \sum_{(v,w) \in E} x(v, w) = 0 \quad \forall v \in V \setminus \{s, t\}$$

$$x(u, v) \leq c(u, v) \quad \forall (u, v) \in E$$

$$x(u, v) \geq 0 \quad \forall (u, v) \in E.$$
Network flow: Equivalent formulation

Directed graph $G = (V, E)$, with capacities on edges. Add a $t$ to $s$ edge with infinite capacity. Now maximize flow on this edge.

$$\max \quad x(t, s)$$

$$\sum_{(u,v) \in E} x(u, v) - \sum_{(v,w) \in E} x(v, w) = 0 \quad \forall v \in V$$

$$x(u, v) \leq c(u, v) \quad \forall (u, v) \in E \setminus (t, s)$$

$$x(u, v) \geq 0 \quad \forall (u, v) \in E.$$
Linear Programming is a useful and powerful (modeling) problem.
Summary

1. Linear Programming is a useful and powerful (modeling) problem.

2. Can be solved in polynomial time. Practical solvers available commercially as well as in open source. Whether there is a strongly polynomial time algorithm is a major open problem.
1. Linear Programming is a useful and powerful (modeling) problem.

2. Can be solved in polynomial time. Practical solvers available commercially as well as in open source. Whether there is a strongly polynomial time algorithm is a major open problem.

3. Geometry and linear algebra are important to understand the structure of LP and in algorithm design. Vertex solutions imply that LPs have poly-sized optimum solutions. This implies that LP is in $\text{NP}$. 

Duality is a critical tool in the theory of linear programming. Duality implies the Linear Programming is in $\text{co-NP}$. Do you see why?
1 Linear Programming is a useful and powerful (modeling) problem.

2 Can be solved in polynomial time. Practical solvers available commercially as well as in open source. Whether there is a strongly polynomial time algorithm is a major open problem.

3 Geometry and linear algebra are important to understand the structure of LP and in algorithm design. Vertex solutions imply that LPs have poly-sized optimum solutions. This implies that LP is in \( \text{NP} \).

4 Duality is a critical tool in the theory of linear programming. Duality implies the Linear Programming is in \( \text{co-NP} \). Do you see why?
Part IV

Integer Linear Programming
Integer Linear Programming

Problem

Find a vector $x \in \mathbb{Z}^d$ (integer values) that

maximize $\sum_{j=1}^{d} c_j x_j$

subject to $\sum_{j=1}^{d} a_{ij} x_j \leq b_i$ for $i = 1 \ldots n$

Input is matrix $A = (a_{ij}) \in \mathbb{R}^{n \times d}$, column vector $b = (b_i) \in \mathbb{R}^n$, and row vector $c = (c_j) \in \mathbb{R}^d$
Factory Example

maximize \( x_1 + 6x_2 \)
subject to \( x_1 \leq 200 \quad x_2 \leq 300 \quad x_1 + x_2 \leq 400 \)
\( x_1, x_2 \geq 0 \)

Suppose we want \( x_1, x_2 \) to be integer valued.
1. Feasible values of $x_1$ and $x_2$ are integer points in shaded region.

2. Optimization function is a line; moving the line until it just leaves the final integer point in feasible region, gives optimal values.
Feasible values of $x_1$ and $x_2$ are integer points in shaded region.

Optimization function is a line; moving the line until it just leaves the final integer point in feasible region, gives optimal values.
1. Feasible values of $x_1$ and $x_2$ are integer points in shaded region.

2. Optimization function is a line; moving the line until it just leaves the final integer point in feasible region, gives optimal values.
Integer Programming

Can model many difficult discrete optimization problems as integer programs!

Therefore integer programming is a hard problem. NP-hard.
Can model many difficult discrete optimization problems as integer programs!

Therefore integer programming is a hard problem. NP-hard.

Can relax integer program to linear program and approximate.
Can model many difficult discrete optimization problems as integer programs!

Therefore integer programming is a hard problem. NP-hard.

Can relax integer program to linear program and approximate.

Practice: integer programs are solved by a variety of methods

1. branch and bound
2. branch and cut
3. adding cutting planes
4. linear programming plays a fundamental role
Example: Maximum Independent Set

**Definition**

Given undirected graph $G = (V, E)$ a subset of nodes $S \subseteq V$ is an independent set (also called a stable set) if for there are no edges between nodes in $S$. That is, if $u, v \in S$ then $(u, v) \notin E$.

**Input** Graph $G = (V, E)$

**Goal** Find maximum sized independent set in $G$
Example: Dominating Set

**Definition**

Given undirected graph $G = (V, E)$ a subset of nodes $S \subseteq V$ is a dominating set if for all $v \in V$, either $v \in S$ or a neighbor of $v$ is in $S$.

**Input**

Graph $G = (V, E)$, weights $w(v) \geq 0$ for $v \in V$

**Goal**

Find minimum weight dominating set in $G$
Example: s-t minimum cut and implicit constraints

**Input**  
Graph $G = (V, E)$, edge capacities $c(e), e \in E$. $s, t \in V$.

**Goal**  
Find minimum capacity $s$-$t$ cut in $G$. 
Suppose we know that for a linear program all vertices have integer coordinates. All of above problems can hence be solved efficiently.
Suppose we know that for a linear program all vertices have integer coordinates. Then solving linear program is same as solving integer program. We know how to solve linear programs efficiently (polynomial time) and hence we get an integer solution for free!
Suppose we know that for a linear program all vertices have integer coordinates. Then solving linear program is same as solving integer program. We know how to solve linear programs efficiently (polynomial time) and hence we get an integer solution for free!

**Luck or Structure:**

1. Linear program for flows with integer capacities have integer vertices
2. Linear program for matchings in bipartite graphs have integer vertices
3. A complicated linear program for matchings in general graphs have integer vertices.

All of above problems can hence be solved efficiently.
Meta Theorem: A combinatorial optimization problem can be solved efficiently if and only if there is a linear program for problem with integer vertices.

Consequence of the Ellipsoid method for solving linear programming.

In a sense linear programming and other geometric generalizations such as convex programming are the most general problems that we can solve efficiently.
Linear Programming is a useful and powerful (modeling) problem.

Can be solved in polynomial time. Practical solvers available commercially as well as in open source. Whether there is a strongly polynomial time algorithm is a major open problem.

Geometry and linear algebra are important to understand the structure of LP and in algorithm design. Vertex solutions imply that LPs have poly-sized optimum solutions. This implies that LP is in \( \text{NP} \).

Duality is a critical tool in the theory of linear programming. Duality implies the Linear Programming is in \( \text{co-NP} \). Do you see why?

Integer Programming in \( \text{NP-Complete} \). LP-based techniques critical in heuristically solving integer programs.