Primal-Dual LP and column generation

Given a vector $\vec{a}_1$ and a value $b_1 \in \mathbb{R}$, the points $\{ \vec{x} \in \mathbb{R} \mid \vec{x} \cdot \vec{a}_1 = b_1 \}$ is a line. The halfplane $\{ \vec{x} \in \mathbb{R} \mid \vec{x} \cdot \vec{a}_1 \leq b_1 \}$ is on the opposite side from the direction that $\vec{a}_1$ points to.

\[
\vec{a}_1 = (0.88, 0.47)
\]

Primal LP Problem:
Given matrix $A = -\vec{a}_1 - \vec{a}_2 - \ldots - \vec{a}_n$
and \(\vec{b} = (b_1, \ldots, b_n)\) \(\in \mathbb{R}^n\), \(\vec{c} \in \mathbb{R}^d\),
Find $\vec{x} \in \mathbb{R}^d$, max $\vec{c} \cdot \vec{x}$ s. t. $A \vec{x} \leq \vec{b}$, $\vec{x} \geq 0$

Dual Problem:
Find $\vec{y} \in \mathbb{R}^n$, min $\vec{b} \cdot \vec{y}$ s. t. $\vec{y} A \geq \vec{c}$, $\vec{y} \geq 0$

Solving Art Gallery with ILP
- Given a polygon, find a subset of the vertices that sees every other vertex
- Let $F(i)$ be the set of vertices that vertex $i$ sees.
- For a vertex $v_i$, we set $x_i = 1$ if we place a guard at $v_i$.
- As usual, $x_i$ are integers between 0 to 1.

Cutting stocks
- Number of boards cut by pattern $I$: $x_I$

Credit: Sergiy Butenko
Cutting stocks

Need to produce
80 4-ft boards,
50 6-ft boards, and
100 7-ft boards.

- $n_i$ number of boards cut by pattern $i$.
- In this example, we have 6 patterns.
- Each pattern has a board length.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>1 2 3 4 5 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>1 2 3 4 5 6</td>
</tr>
<tr>
<td>$b_i$</td>
<td>2 0 2 0 1 1</td>
</tr>
<tr>
<td>$c_i$</td>
<td>0 0 2 1 0 1</td>
</tr>
<tr>
<td>$d_i$</td>
<td>0 0 2 1 0 1</td>
</tr>
</tbody>
</table>

First raw of M refers to 4-ft boards.
Second raw of M refers to 6-ft boards.
Third raw of M refers to 7-ft boards.

Produce min # of stocks, each of length $l$.

Calculate this matrix $M$.

For each stock:
- $\sum x_i M_{ij} \geq 80$
- $\sum x_i M_{ij} \geq 50$
- $\sum x_i M_{ij} \geq 70$

Conclusion:
- If $x_i \cdot y_i = \min$ for all patterns, then $x_i, y_i$ are both optimum solutions.
Application 1: Network flow

Primal: Max Flow

\[ \text{Maximize} \sum_{(u,v) \in E} x_{u,v} \]
\[ \text{s.t.} \]
\[ \sum_{v \in \delta^-(u)} x_{u,v} - \sum_{v \in \delta^+(u)} x_{v,u} = \begin{cases} 1 & \text{if } u = s \\ 0 & \text{otherwise} \end{cases} \]
\[ \sum_{v \in \delta^-(v)} x_{v,u} - \sum_{v \in \delta^+(v)} x_{u,v} = \begin{cases} 0 & \text{if } v = t \\ 1 & \text{if } v = t \end{cases} \]
\[ x_{u,v} \geq 0 \]

Dual: Min Cut

\[ \text{Minimize} \sum_{(u,v) \in E} c_{u,v} \cdot \bar{x}_{u,v} \]
\[ \text{s.t.} \]
\[ \sum_{u \in V} \bar{x}_{u,v} = 1 \quad \forall v \in \delta^+(v) \]
\[ \sum_{v \in V} \bar{x}_{u,v} = 1 \quad \forall u \in \delta^-(u) \]
\[ \bar{x}_{u,v} \geq 0 \]

More applications: Column generation

- Consider the LP in the primal where number of variables is exponential.
- Let's solve the dual - now the number of constraints is huge. \( \min y \cdot \bar{b} \) s.t. \( \bar{A} y \geq \bar{c} \).
- Pick only the first \( d \) constraints, let \( A' \) be the matrix with these rows (primal).
- Solve LP when \( A \) is replaced by \( A' \). Let \( y^* \) be the optimal solution.
- Note: if \( A'y^* \geq \bar{c} \), we are done.
- Find (one) \( \bar{y} \) violating \( A'y < \bar{c} \). If such a row exists, we are done.
- Add this row to \( A' \) and repeat.

Finding this row could, in many cases, be done much faster than checking all exponential rows explicitly.

Proving optimality

- Say we have an NP-hard problem, (say TSP, or Vertex Cover). Via heuristics we found a solution - a route that visits all cities. How could we prove that it is optimal?
- Note we did not find an algorithm that always finds an optimal solution to every graph. We found a solution to one specific graph.
- We could look at the dual problem, and if the costs are equal, then they are optimum.

Cutting stocks

Need to buy min \# of stocks, each of length 15.

Need to produce

- 80 4-ft boards,
- 50 6-ft boards, and
- 100 7-ft boards.

- \( x_i \) number of boards cut by pattern \( i \).
- Define \( \bar{c} = (1,1,1) \).

A stock of length 15

<table>
<thead>
<tr>
<th>Board of length 4</th>
<th>Board of length 4</th>
<th>Board of length 4</th>
<th>Waste</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

problem: M has exponentially many columns

Applications to graphs

Primal LP Problem:

Find \( \bar{y} \in \mathbb{R}^n \), one \( \bar{y} \) per edge \( \bar{c} \cdot \bar{y} \) where \( \bar{c} = (1,1,1,1,1) \). \( \bar{c} \cdot \bar{y} \).

Dual LP Problem:

Find \( \bar{x} \in \mathbb{R}^m \), one \( \bar{x} \) per vertex \( \bar{x} \cdot \bar{b} \) where \( \bar{b} = (1,1,1) \). \( \bar{x} \cdot \bar{b} \).

An integer solution picks edges, so for every vertex \( v \), only one of the edges connecting to it is picked.

Recall: Any feasible solution to primal \( \leq \) any solution to dual.

Conclusion: max-matching \( \leq \) min vertex cover.
Another Application: Sensors Scheduling

We have a sensor in each vertex of a polygon \( P \).

Each sensor has a battery for 1 hour.

Determine when to activate/deactivate each one so overall cover time is max

\[
\sum_{i=1}^{n} x_i \quad \text{s.t.} \quad \sum_{k \in V_i} x_k \geq 1 \quad \forall 1 \leq i \leq n
\]

- Given a polygon, find a subset of the vertices that
  sees every other vertex.
- For a vertex \( v \) we set \( x = 1 \) if we place a guard at \( v \).
- As usual, \( x \) are integers between 0 to 1.

\[
\begin{align*}
x_1 + x_2 + x_4 + x_6 & \geq 1 \quad \text{guarantees } v_1 \text{ is seen} \\
x_1 + x_2 + x_4 & \geq 1 \quad \text{guarantees } v_2 \text{ is seen} \\
x_2 + x_3 + x_5 & \geq 1 \quad \text{guarantees } v_3 \text{ is seen} \\
x_1 + x_2 + x_4 + x_5 + x_6 & \geq 1 \quad \text{guarantees } v_4 \text{ is seen} \\
x_1 + x_2 + x_5 + x_6 & \geq 1 \quad \text{guarantees } v_5 \text{ is seen} \\
x_1 + x_2 + x_4 + x_5 & \geq 1 \quad \text{guarantees } v_6 \text{ is seen}
\end{align*}
\]