Randomized Algorithms

Randomized Rounding

- **Brief Introduction to Linear Programming and Its Usage in Combinatorial Optimization**
- Randomized Rounding for Cut Problems
- Randomized Rounding for Satisfiability Problems
- Randomized Rounding for Covering Problems
- Randomized Rounding and Semi-definite Programming

Approximate Sampling and Counting

- ...

©Hung Q. Ngo (SUNY at Buffalo)
Some Combinatorial Optimization Problems

- MAXFLOW and MINCUT problems
- MULTIWAY CUT problem
- MAX-2SAT, MAX-E3SAT, MAX-SAT problems
- SET COVER, VERTEX COVER problems

They can all be formulated as (integer) linear programs
Cornerstone problems in combinatorial optimization

Many non-trivial applications/reductions: airline scheduling, data mining, bipartite matching, image segmentation, network survivability, many many many more ...

Simple Example: on the Internet with error-free transmission, what is the maximum data rate that a router \( s \) can send to a router \( t \) (assuming no network coding is allowed), given that each link has limited capacity

More examples and applications to come
A flow network is a directed graph \( G = (V, E) \) where each edge \( e \) has a capacity \( c(e) > 0 \).

Also, there are two distinguished nodes: the source \( s \) and the sink \( t \).
Cuts

- An $s, t$-cut is a partition $(A, B)$ of $V$ where $s \in A$, $t \in B$
- Let $[A, B] = \text{set of edges } (u, v) \text{ with } u \in A, v \in B$
- The capacity of the cut $(A, B)$ is defined by

$$\text{cap}(A, B) = \sum_{e \in [A, B]} c(e)$$

### Capacity Example

In the given graph, the capacity is calculated as follows:

- $s \rightarrow A$: edge capacities $5$ and $15$.
- $A \rightarrow 3$: edge capacities $4$ and $15$.
- $3 \rightarrow 6$: edge capacity $8$.
- $6 \rightarrow t$: edge capacity $10$.
- $4 \rightarrow 7$: edge capacity $30$.

The total capacity is $5 + 15 + 4 + 15 + 8 + 10 + 30 = 97$.
An $s, t$-cut is a partition $(A, B)$ of $V$ where $s \in A$, $t \in B$

Let $[A, B] = \text{set of edges (}u, v\text{)}$ with $u \in A, v \in B$

The capacity of the cut $(A, B)$ is defined by

$$\text{cap}(A, B) = \sum_{e \in [A,B]} c(e)$$

![Graph with cut](image)
Minimum Cut - Problem Definition

Given a flow network, find an $s, t$-cut with minimum capacity

![Flow Network Diagram]

- Capacity = $10 + 8 + 10 = 28$
Flows

- An \( s,t \)-flow is a function \( f : E \rightarrow \mathbb{R} \) satisfying:
  - Capacity constraint: \( 0 \leq f(e) \leq c(e), \forall e \in E \)
  - Flow Conservation constraint: \( \sum_{e=(u,v) \in E} f(e) = \sum_{e=(v,w) \in E} f(e) \)

- The value of \( f \): \( \text{val}(f) = \sum_{e=(s,v) \in E} f(e) \)

\[\begin{array}{c}
\text{Values on edges:} \\
4 & 0 & 9 \\
10 & 0 & 15 \\
4 & 4 \\
5 & \text{cap: 15} \\
\text{flow: 0} \\
\end{array}\]
An \( s, t \)-flow is a function \( f : E \rightarrow \mathbb{R} \) satisfying:

- **Capacity constraint:** \( 0 \leq f(e) \leq c(e), \ \forall e \in E \)
- **Flow Conservation constraint:** \( \sum_{e=(u,v) \in E} f(e) = \sum_{e=(v,w) \in E} f(e) \)

The value of \( f \): \( \text{val}(f) = \sum_{e=(s,v) \in E} f(e) \)

![Graph](image-url)
Given a flow network, find a flow $f$ with maximum capacity.
First Linear Program for Maximum Flow

\[
\begin{align*}
\text{max} & \quad \sum_{e \in E} f_e \\
\text{subject to} & \quad f_e \leq c_e, \quad \forall e \in E, \\
& \quad \sum_{uv \in E} f_{uv} - \sum_{vw \in E} f_{vw} = 0, \quad \forall v \neq s, t \\
& \quad f_e \geq 0, \quad \forall e \in E
\end{align*}
\]
Let $\mathcal{P}$ be the set of all $s, t$-paths.

$f_P$ denote the flow amount sent along $P$

$$\begin{align*}
\text{max} \quad & \sum_{P \in \mathcal{P}} f_P \\
\text{subject to} \quad & \sum_{P: e \in P} f_P \leq c_e, \quad \forall e \in E, \\
& f_P \geq 0, \quad \forall P \in \mathcal{P}.
\end{align*}$$

(2)
What are Linear Programs?

Optimize linear objective subject to linear equalities/inequalities

Example 1:

$$\begin{align*}
\text{min} & \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
\text{subject to} & \quad a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n = b_1 \\
& \quad a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n = b_2 \\
& \quad \vdots \quad \vdots \quad \vdots \quad = \quad \vdots \\
& \quad a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n = b_m \\
\end{align*}$$

$$x_i \geq 0, \forall i = 1, \ldots, n,$$

Or simply: $\text{min}\{c^T x \mid Ax = b, x \geq 0\}$
What are Linear Programs?

Optimize linear objective subject to linear equalities/inequalities

Example 2:

\[
\begin{align*}
\text{max} & \quad c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
\text{subject to} & \quad a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n \leq b_1 \\
& \quad a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n \leq b_2 \\
& \quad \vdots \quad \vdots \quad \vdots \\
& \quad a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n \leq b_m \\
& \quad x_i \geq 0, \forall i = 1, \ldots, n,
\end{align*}
\]

Or simply: \( \max \{ c^T x \mid Ax \leq b, x \geq 0 \} \)
Certainly, constraints may be mixed: $=, \leq, \geq$, some variables may not need to be non-negative, etc.

Example 3:

$$\min / \max \quad a^T x + b^T y + c^T z$$
subject to
$$A_{11} x + A_{12} y + A_{13} z = d$$
$$A_{21} x + A_{22} y + A_{23} z \leq e$$
$$A_{31} x + A_{32} y + A_{33} z \geq f$$
$$x \geq 0, y \leq 0.$$ 

Note that $A_{ij}$ are matrices and $a, b, c, d, e, f, x, y, z$ are vectors.

Fortunately, easy to “convert” any LP into any one of the following:

- The min and the max versions of the standard form:

$$\min \{ c^T x \mid Ax = b, x \geq 0 \}, \quad \text{and} \quad \max \{ c^T x \mid Ax = b, x \geq 0 \}.$$ 

- The min and the max versions of the canonical form:

$$\min \{ c^T x \mid Ax \geq b, x \geq 0 \}, \quad \text{and} \quad \max \{ c^T x \mid Ax \leq b, x \geq 0 \}.$$ 

Solving Linear Programs

- **Simplex Method** (Dantzig, 1948): worst-case exponential time, but runs very fast on most practical inputs
- **Ellipsoid Method** (Khachian, 1979): worst-case polynomial time, but quite slow in practice. Can even solve some LP with an exponential number of constraints if a separation oracle exists
- **Interior Point Method** (Karmarkar, 1984): worst-case polynomial time, quite fast in practice, not as popular as the simplex method
Linear Programming Duality

To each LP (called the **primal LP**) there corresponds another LP called the **dual LP** satisfying the following:

<table>
<thead>
<tr>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasible</td>
<td>Feasible</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
</tr>
<tr>
<td>Infeasible</td>
<td>Unbounded</td>
</tr>
<tr>
<td>Unbounded</td>
<td>Infeasible</td>
</tr>
</tbody>
</table>

(X = Possible, O = Impossible)

If the primal is a \( \min \{ \ldots \} \), then the dual is a \( \max \{ \ldots \} \) and vice versa.

**Theorem (Strong duality)**

*If both the primal and the dual LPs are feasible, then their optimal objective values are the same.*
## Rules for Writing Down the Dual LP

<table>
<thead>
<tr>
<th>Maximization problem</th>
<th>Minimization problem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Constraints</strong></td>
<td><strong>Variables</strong></td>
</tr>
<tr>
<td>$i$th constraint $\leq$</td>
<td>$i$th variable $\geq 0$</td>
</tr>
<tr>
<td>$i$th constraint $\geq$</td>
<td>$i$th variable $\leq 0$</td>
</tr>
<tr>
<td>$i$th constraint $=$</td>
<td>$i$th variable unrestricted</td>
</tr>
<tr>
<td><strong>Variables</strong></td>
<td><strong>Constraints</strong></td>
</tr>
<tr>
<td>$j$th variable $\geq 0$</td>
<td>$j$th constraint $\geq$</td>
</tr>
<tr>
<td>$j$th variable $\leq 0$</td>
<td>$j$th constraint $\leq$</td>
</tr>
<tr>
<td>$j$th variable unrestricted</td>
<td>$j$th constraint $=$</td>
</tr>
</tbody>
</table>

**Table:** Rules for converting between primals and duals.
In **standard form**, the primal and dual LPs are

\[
\begin{align*}
\text{min} \quad & c^T x & \quad \text{(primal program)} \\
\text{subject to} \quad & Ax = b \\
& x \geq 0 \\
\text{max} \quad & b^T y & \quad \text{(dual program)} \\
\text{subject to} \quad & A^T y \leq c & \text{no non-negativity restriction!}
\end{align*}
\]
In **canonical form**, the primal and dual LPs are

\[
\begin{align*}
\text{min} & \quad c^T x \quad \text{(primal program)} \\
\text{subject to} & \quad Ax \geq b \\
& \quad x \geq 0 \\
\text{max} & \quad b^T y \quad \text{(dual program)} \\
\text{subject to} & \quad A^T y \leq c \\
& \quad y \geq 0.
\end{align*}
\]
Weak Duality and Strong Duality

Primal LP: \( \min \{ c^T x \mid Ax \geq b, x \geq 0 \} \)
Dual LP: \( \max \{ b^T y \mid A^T y \leq c, y \geq 0 \} \).

Theorem (Weak Duality)
Suppose \( x \) is primal feasible, and \( y \) is dual feasible, then \( c^T x \geq b^T y \).
In particular, if \( x^* \) is primal-optimal and \( y^* \) is dual-optimal, then
\[
c^T x^* \geq b^T y^*.
\]

Theorem (Strong Duality)
If the primal LP has an optimal solution \( x^* \), then the dual LP has an optimal solution \( y^* \) such that
\[
c^T x^* = b^T y^*.
\]
Given the following programs

**Primal LP:** \( \min \{ c^T x \mid Ax \geq b, x \geq 0 \} \),

**Dual LP:** \( \max \{ b^T y \mid A^T y \leq c, y \geq 0 \} \).

Let \( x^* \) and \( y^* \) be feasible for the primal and the dual programs, respectively. Then, \( x^* \) and \( y^* \) are optimal for their respective LPs if and only if

\[
(c - A^T y^*)^T x^* = 0, \quad and \quad (b - Ax)^T y^* = 0.
\]
Intuition: for a cut \((A, B)\), set \(x_v = 1\) if \(v \in A\) and \(x_v = 0\) otherwise.

\[
\begin{align*}
\text{min} \quad & \sum_{e \in E} c_e z_e \\
\text{subject to} \quad & z_e \geq x_u - x_v \quad \forall e = uv \in E, \\
& z_e \geq x_v - x_u \quad \forall e = uv \in E, \\
& x_s = 1 \\
& x_t = 0 \\
& z_e, x_v \in \{0, 1\}, \quad \forall v \in V, e \in E
\end{align*}
\]
Second ILP for Mincut

Let $\mathcal{P}$ be the collection of all $s,t$-paths

\[
\min \sum_{e \in E} c_e y_e \\
\text{subject to } \sum_{e \in P} y_e \geq 1, \quad \forall P \in \mathcal{P}, \\
y_e \in \{0, 1\}, \quad \forall e \in E.
\]
Multiway Cut

**MULTIWAY CUT:**

Given an edge weighted graph $G = (V, E)$ ($w : E \to \mathbb{R}^+$) and $k$ terminals $\{t_1, \ldots, t_k\}$. Find a min-weight subset of edges whose removal disconnects the terminals from one another.

Let $\mathcal{P}$ be the collection of all $s_i, s_j$-paths

$$
\begin{align*}
\min & \quad \sum_{e \in E} w_e x_e \\
\text{subject to} & \quad \sum_{e \in P} x_e \geq 1, \quad \forall P \in \mathcal{P}, \\
& \quad x_e \in \{0, 1\}, \quad \forall e \in E.
\end{align*}
$$

(5)
Vertex Cover

**Weighted Vertex Cover**

Given a graph $G = (V, E)$, $|V| = n$, $|E| = m$, a weight function $w : V \to \mathbb{R}$. Find a vertex cover $C \subseteq V$ for which $\sum_{i \in C} w(i)$ is minimized.

An equivalent integer linear program (ILP) is

\[
\begin{align*}
\text{min} & \quad w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \\
\text{subject to} & \quad x_i + x_j \geq 1, \quad \forall ij \in E, \\
& \quad x_i \in \{0, 1\}, \quad \forall i \in V.
\end{align*}
\]
**Weighted Set Cover**

*Given a collection $S = \{S_1, \ldots, S_n\}$ of subsets of $[m] = \{1, \ldots, m\}$, and a weight function $w : S \rightarrow \mathbb{R}$. Find a cover $C = \{S_j \mid j \in J\}$ with minimum total weight.*

Use a 01-variable $x_j$ to indicate the inclusion of $S_j$ in the cover. The corresponding ILP is thus

$$
\begin{align*}
\text{min} & \quad w_1 x_1 + \cdots + w_n x_n \\
\text{subject to} & \quad \sum_{j : S_j \ni i} x_j \geq 1, \quad \forall i \in [m], \\
& \quad x_j \in \{0, 1\}, \quad \forall j \in [n].
\end{align*}
$$
Max-SAT

**Weighted max-sat:**

Given a CNF formula $\varphi$ with $m$ weighted clauses on $n$ variables, find a truth assignment maximizing the total weight of satisfied clauses.

Say, clause $C_j$ has weight $w_j \in \mathbb{R}^+$. Here’s an ILP

$$\begin{align*}
\text{max} & \quad w_1 z_1 + \cdots + w_m z_n \\
\text{subject to} & \quad \sum_{i : x_i \in C_j} y_i + \sum_{i : \bar{x}_i \in C_j} (1 - y_i) \geq z_j, \quad \forall j \in [m], \\
& \quad y_i, z_j \in \{0, 1\}, \quad \forall i \in [n], j \in [m]
\end{align*}$$