A Polynomial-Time Cutting-Plane Algorithm for Matchings

László Végh
London School of Economics

Joint work with Karthik Chandrasekaran & Santosh Vempala.
Slides thanks to Karthik.
Cutting Plane Method

\[ P = \{ x \in \mathbb{R}^n : Ax \leq b \} \]

\[ P_I = \text{conv-hull}(P \cap \mathbb{Z}^n) \]
Cutting Plane Method

\[ P = \{ x \in \mathbb{R}^n : Ax \leq b \} \]

\[ P_I = \text{conv-hull}(P \cap \mathbb{Z}^n) \]
Cutting Plane Method

\[ P = \{ x \in \mathbb{R}^n : A x \leq b \} \]

\[ P_I = \text{conv-hull}(P \cap \mathbb{Z}^n) \]
Cutting Plane Method

\[ P = \{ x \in \mathbb{R}^n : Ax \leq b \} \]

\[ P_I = \text{conv-hull}(P \cap \mathbb{Z}^n) \]
Cutting Plane Method

1. **Starting LP.** Start with the LP relaxation of the given IP to obtain basic optimal solution $x$

2. Repeat until $x$ is integral:
   a. **Add Cuts.** Find a linear inequality that is valid for the convex hull of integer solutions but violated by $x$ and add it to the LP
   b. **Re-solve LP.** Obtain basic optimal solution $x$
Cutting Plane Method

- Proposed by Dantzig-Fulkerson-Johnson (1954)
- Several cut-generation procedures
  - Gomory cuts [Gomory (1958)]
  - Intersection cuts [Balas (1971)]
  - Disjunctive cuts [Balas (1979)]
  - Split cuts [Cook-Kannan-Schrijver (1990)]
  - MIR Inequalities [Nemhauser-Wolsey (1990)]
  - Lift-and-project methods
    [Sherali-Adams (1990), Lovász-Schrijver (1991), Balas-Ceria-Cornuéjols (1993)]
- Closure properties of polytopes [Chvátal (1973)]
  - Chvatal-Gomory rank
- Cutting plane proof system
  [Chvátal-Cook-Hartmann (1989)]
- Implemented in commercial IP solvers
Requirements – Efficient and Correct

- Efficient cut-generation procedure [Gomory(1958)]
- Convergence to integral solution [Gomory(1958)]
- Fast convergence (number of cuts to reach integral solution)
  - At most $2^n$ cuts for 0/1-IP [Gomory(1958)]
  - No hope for faster theoretical convergence using Gomory cuts
  - Practical implementations seem to be efficient for certain problems
Requirements – Efficient and Correct

- Efficient cut-generation procedure [Gomory(1958)]
- Convergence to integral solution [Gomory(1958)]
- Fast convergence (number of cuts to reach integral solution)
  - At most \(2^n\) cuts for 0/1-IP [Gomory(1958)]
  - No hope for faster theoretical convergence using Gomory cuts
  - Practical implementations seem to be efficient for certain problems

Question: Can we explain the efficiency in practical implementations?
Minimum-Cost Perfect Matching

- **Problem**: Given an edge cost function in a graph, find a minimum cost perfect matching
Minimum-Cost Perfect Matching

- **Problem**: Given an edge cost function in a graph, find a minimum cost perfect matching

<table>
<thead>
<tr>
<th></th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edmonds (1965)</td>
<td>$O(n^2 m)$</td>
</tr>
<tr>
<td>Gabow (1973), Lawler (1976)</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>Galil-Micali-Gabow (1986)</td>
<td>$O(nm \log n)$</td>
</tr>
<tr>
<td>Gabow (1990)</td>
<td>$O(n(m + n \log n))$</td>
</tr>
<tr>
<td>Gabow-Tarjan (1991)</td>
<td>$O(m \log (n|c|_\infty) \sqrt{n \log n})$</td>
</tr>
</tbody>
</table>
Minimum-Cost Perfect Matching

- **Problem**: Given an edge cost function in a graph, find a minimum cost perfect matching

- **IP formulation**:

\[
\begin{align*}
\min & \quad \sum_{uv \in E} c(uv)x(uv) \\
\text{s.t.} & \quad x(\delta(u)) = 1 \quad \forall u \in V \quad \text{(degree constraints)} \\
& \quad x \geq 0 \quad \text{(non-negativity constraints)} \\
& \quad x \in \mathbb{Z}^E
\end{align*}
\]

\[
x(\delta(u)) := \sum_{e : e = uv} x(e)
\]
LP Relaxation

\[
\min \sum_{uv \in E} c(uv)x(uv)
\]

\[
x(\delta(u)) = 1 \ \forall u \in V \quad \text{(degree constraints)}
\]

\[
x \geq 0 \quad \text{(non-negativity constraints)}
\]

\[
x \in \mathbb{Z}^E
\]
LP Relaxation  (Bipartite Relaxation)

\[
\min \sum_{uv \in E} c(uv)x(uv)
\]

\[
x(\delta(u)) = 1 \ \forall u \in V \quad \text{(degree constraints)}
\]

\[
x \geq 0 \quad \text{(non-negativity constraints)}
\]
LP Relaxation (Bipartite Relaxation)

\[
\min \sum_{uv \in E} c(uv)x(uv)
\]

\[
x(\delta(u)) = 1 \ \forall u \in V \quad \text{(degree constraints)}
\]

\[
x \geq 0 \quad \text{(non-negativity constraints)}
\]

- In bipartite graphs, the solution is the indicator vector of a perfect matching in the graph

- In non-bipartite graphs, the solution is not necessarily integral
  - Solution is half-integral and supported by a disjoint union of edges and odd cycles

\[x_e = \frac{1}{2}\]
Polyhedral Characterization [Edmonds(1965)]

\[
\begin{align*}
\min \sum_{uv \in E} c(uv)x(uv) \\
x(\delta(u)) &= 1 \forall u \in V \\
x &\geq 0 \\
x(\delta(S)) &\geq 1 \forall S \subseteq V, |S| \text{odd}
\end{align*}
\]

(degree constraints)
(non-negativity constraints)
(odd-set inequalities) — Exponentially many!

\[
x(\delta(S)) := \sum_{e=uv : u \in S, v \in V \setminus S} x(e)
\]

- Odd-set inequalities
  - Gomory cuts derived from bipartite relaxation
    [Chvátal (1973)]

- Efficient separation oracle exists
  - For any given point \( x \), we can efficiently verify if \( x \) satisfies all constraints and if not output a violated constraint
    [Padberg-Rao (1982)]
Cutting Plane Algorithm [Padberg – Rao(1982)]

1. **Starting LP.** Start with the **bipartite relaxation** to obtain basic optimal solution $\mathbf{x}$

2. Repeat until $\mathbf{x}$ is integral:
   a. **Add Cuts.** Find an **odd-set inequality** that is violated by $\mathbf{x}$ using the Padberg-Rao procedure and add it to the LP
   b. **Re-solve LP.** Obtain basic optimal solution $\mathbf{x}$

- Easy to find cuts in the first step
  - Starting LP optimum is half-integral and supported by a disjoint union of odd cycles and edges
- Half-integral structure is not preserved in later steps - need Padberg-Rao for cut generation

$c_e = 1 \forall e$
Cutting Plane Algorithm [Padberg – Rao (1982)]

- Discussed by Lovász-Plummer (1986)
  - Computationally efficient

**Result:** A cutting plane algorithm for minimum-cost perfect matching
  - Convergence guarantee: $O(n \log n)$ rounds of cuts for an $n$-vertex graph
  - Black-box LP solver after each round of cut addition
  - Intermediate LP optima are half-integral
Cutting Plane Algorithm [Padberg – Rao(1982)]

- Discussed by Lovász-Plummer (1986)
  - Computationally efficient

**Result:** A cutting plane algorithm for minimum-cost perfect matching
- Convergence guarantee: $O(n \log n)$ rounds of cuts for an $n$-vertex graph
- Black-box LP solver after each round of cut addition
- Intermediate LP optima are half-integral

- Bunch, PhD thesis in Chemical Engineering (1997): similar (weaker) result, using specifically implemented simplex rule.
Outline

- Cutting Plane Method for Integer Programs
- Perfect Matching and Linear Constraints
- Results
- New Cutting Plane Algorithm for Matching
  - Analysis
    - Half-integral Structure
    - Progress
      - Coupling with a *new Combinatorial Primal-Dual Algorithm for Matching*
- Future Directions
New Cutting Plane Algorithm

**Primal** - $P_F$

Min $\sum_{uv \in E} c_{uv} x_{uv}$

$x(\delta(u)) = 1 \ \forall u \in V$

$x(\delta(S)) \geq 1 \ \forall S \in F$

$x \geq 0$

1. Perturb the integral cost function by adding $\frac{1}{2i}$ to edge $i$

2. **Starting LP**: Bipartite relaxation ($F = \emptyset$)

3. Repeat until $x$ is integral
   (a) **Retain old cuts.**

(b) **Choose new cuts.**

(c) **Re-solve LP**: Find an optimal solution $x$ to $P_F$

To ensure uniqueness of intermediate optima

Final integral optimum to perturbed cost function is also optimum to original cost function
New Cutting Plane Algorithm

**Primal -** $P_F$

\[
\text{Min } \sum_{uv \in E} c_{uv} x_{uv} \\
x(\delta(u)) = 1 \ \forall u \in V \\
x(\delta(S)) \geq 1 \ \forall S \in F \\
x \geq 0
\]

**Dual -** $D_F$

\[
\text{Max } \sum_{S \in V \cup F} \Pi(S) \\
\sum_{S \in V \cup F: uv \in \delta(S)} \Pi(S) \leq c(uv) \ \forall uv \in E \\
\Pi(S) \geq 0 \ \forall S \in F
\]

1. Perturb the integral cost function by adding $\frac{1}{2i}$ to edge $i$
2. **Starting LP.** Bipartite relaxation ($F = \emptyset$)
3. Repeat until $x$ is integral
   - (a) **Retain old cuts.** Find a specific dual optimal solution $\Pi$ to $D_F$.
     \[
     H' = \{S \in F: \Pi(S) > 0\}
     \]
   - (b) **Choose new cuts.**
4. (c) **Re-solve LP.** Find an optimal solution $x$ to $P_F$
An Example

\[ F = \{S_1, S_2, S_3\} \]
An Example

\[ F = \{ S_1, S_2, S_3 \} \]
An Example

$F = \{S_1, S_2, S_3\}$

$H' = \{S_1, S_3\}$
New Cutting Plane Algorithm

Primal - $P_F$

Min $\sum_{uv \in E} c_{uv} x_{uv}$

$x(\delta(u)) = 1, \forall u \in V$

$x(\delta(S)) \geq 1, \forall S \in F$

$x \geq 0$

Dual - $D_F$

Max $\sum_{S \in V \cup F} \Pi(S)$

$\sum_{S \in V \cup F : uv \in \delta(S)} \Pi(S) \leq c(uv), \forall uv \in E$

$\Pi(S) \geq 0, \forall S \in F$

1. Perturb the integral cost function by adding $\frac{1}{2^i}$ to edge $i$
2. Starting LP. Bipartite relaxation ($F = \emptyset$)
3. Repeat until $x$ is integral
   (a) Retain old cuts. Find a specific dual optimal solution $\Pi$ to $D_F$.
      \[ H' = \{ S \in F : \Pi(S) > 0 \} \]
   (b) Choose new cuts.
   (c) Re-solve LP. Find an optimal solution $x$ to $P_F$
An Example

\[ F = \{S_1, S_2, S_3\} \]

\[ H' = \{S_1, S_3\} \]
An Example

\[ F = \{ S_1, S_2, S_3 \} \]

\[ H' = \{ S_1, S_3 \} \]
An Example

\[ F = \{S_1, S_2, S_3\} \]

\[ H' = \{S_1, S_3\} \]
An Example

\[ F = \{S_1, S_2, S_3\} \]
\[ H' = \{S_1, S_3\} \]
\[ H'' = \{\hat{C}\} \]
New Cutting Plane Algorithm

**Primal -** $P_F$

\[
\begin{align*}
\text{Min } & \sum_{uv \in E} c_{uv} x_{uv} \\
x(\delta(u)) &= 1 \ \forall u \in V \\
x(\delta(S)) &\geq 1 \ \forall S \in F \\
x &\geq 0
\end{align*}
\]

**Dual -** $D_F$

\[
\begin{align*}
\text{Max } & \sum_{S \in V \cup F} \Pi(S) \\
\sum_{S \in V \cup F: uv \in \delta(S)} \Pi(S) &\leq c(uv) \ \forall uv \in E \\
\Pi(S) &\geq 0 \ \forall S \in F
\end{align*}
\]

1. Perturb the integral cost function by adding $\frac{1}{2}i$ to edge $i$
2. **Starting LP.** Bipartite relaxation ($F = \emptyset$)
3. Repeat until $x$ is integral
   - (a) **Retain old cuts.** Find a specific dual optimal solution $\Pi$ to $D_F$.
     \[
     H' = \{S \in F: \Pi(S) > 0\}
     \]
   - (b) **Choose new cuts.**
New Cutting Plane Algorithm

**Primal** - $P_F$

\[
\text{Min } \sum_{uv \in E} c_{uv} x_{uv} \\
x(\delta(u)) = 1 \forall u \in V \\
x(\delta(S)) \geq 1 \forall S \in F \\
x \geq 0
\]

**Dual** - $D_F$

\[
\text{Max } \sum_{S \in V \cup F} \Pi(S) \\
\sum_{S \in V \cup F: uv \in \delta(S)} \Pi(S) \leq c(uv) \forall uv \in E \\
\Pi(S) \geq 0 \forall S \in F
\]

1. Perturb the integral cost function by adding $\frac{1}{2i}$ to edge $i$
2. **Starting LP.** Bipartite relaxation ($F = \emptyset$)
3. Repeat until $x$ is integral
   (a) **Retain old cuts.** Find a specific dual optimal solution $\Pi$ to $D_F$.
   \[
   H' = \{S \in F: \Pi(S) > 0\}
   \]
   (b) **Choose new cuts.** For each cycle $C \in \text{supp}(x)$, define $\hat{C}$ as the union of $V(C)$ and the inclusionwise maximal sets of $H'$ intersecting $V(C)$
   \[
   H'' = \{\hat{C}: C \in \text{supp}(x)\}
   \]
   (c) Set the new $F = H' \cup H''$
   (d) **Re-solve LP.** Find an optimal solution $x$ to $P_F$
Analysis Overview

- **Laminarity**: Intermediate LPs are defined by a laminar family $F$ of odd sets
  
  \[ \Rightarrow \text{At most } \frac{n}{2} \text{ odd-set inequalities in intermediate LPs} \]

(i) **Structural Guarantee**: Intermediate LP optima are half-integral and supported by a disjoint union of odd cycles and edges
  
  \[ \Rightarrow \text{Cut-generation in } O(n) \text{ time} \]

(ii) **Progress**: The number of odd cycles $\text{odd}(x)$ in the support of the intermediate LP optima $x$

  - Non-increasing
  - Decreases by one in at most $\frac{n}{2 \text{odd}(x)}$ rounds of cut addition

  \[ \Rightarrow \text{Number of rounds of cut addition is } O(n \log n) \]
Half-integral Structure

- Conjecture 0: All intermediate solutions are Half-integral
Half-integral Structure

- Conjecture 0: All intermediate solutions are Half-integral
- Conjecture 1: Half-integral if odd-set inequalities correspond to a laminar family
Half-integral Structure

- Conjecture 0: All intermediate solutions are Half-integral
- Conjecture 1: Half-integral if odd-set inequalities correspond to a laminar family

\[ c_e = 1 \ \forall e \]
Half-integral Structure

- Conjecture 0: All intermediate solutions are Half-integral
- Conjecture 1: Half-integral if odd-set inequalities correspond to a laminar family

Hey look – the optima are not unique!
Half-integral Structure

- Conjecture 2: Half-integral if the optimum is unique and odd-set inequalities correspond to a laminar family
Half-integral Structure

- Conjecture 2: Half-integral if the optimum is unique and odd-set inequalities correspond to a laminar family
### Half-integral Structure

**Lemma:** If $x$ is unique, $F$ is laminar, and $D_F$ has an **F-critical** dual optimal solution, then $x$ is half-integral.

- For simplicity, say the sets in $F$ are disjoint.

**Definition.** $\Pi$ is a **F-critical** dual solution to $D_F$ if $\forall S \in F: \Pi(S) > 0$, the induced graph over $S$ using the tight edges wrt $\Pi$ is factor-critical.

- For every $u \in S$, there exists a matching $M_u$
  - $M_u$ covers all vertices in $S \setminus u$
Half-integral Structure

**Primal -** $P_F$

- Min $\sum_{uv \in E} c_{uv} x_{uv}$
- $x(\delta(u)) = 1 \forall u \in V$
- $x(\delta(S)) \geq 1 \forall S \in F$
- $x \geq 0$

**Dual -** $D_F$

- Max $\sum_{S \in V \cup F} \Pi(S)$
- $\sum_{S \in V \cup F: uv \in \delta(S)} \Pi(S) \leq c(uv) \forall uv \in E$
- $\Pi(S) \geq 0 \forall S \in F$

- **Lemma:** If $x$ is unique, $F$ is laminar, and $D_F$ has an **F-critical** dual optimal solution, then $x$ is half-integral.

- For simplicity, say the sets in $F$ are disjoint.

- **Definition.** $\Pi$ is a **F-critical** dual solution to $D_F$ if $\forall S \in F: \Pi(S) > 0$, the induced graph over $S$ using the tight edges wrt $\Pi$ is factor-critical.

  - For every $u \in S$, there exists a matching $M_u$
    - $M_u$ covers all vertices in $S \setminus u$
  - Same notion appears in Edmonds’ blossom algorithm.
New Cutting Plane Algorithm

**Primal** - $P_F$

\[
\begin{align*}
\text{Min} & \quad \sum_{uv \in E} c_{uv} x_{uv} \\
x(\delta(u)) &= 1 \ \forall u \in V \\
x(\delta(S)) &\geq 1 \ \forall S \in F \\
x &\geq 0
\end{align*}
\]

**Dual** - $D_F$

\[
\begin{align*}
\text{Max} & \quad \sum_{S \in V \cup F} \Pi(S) \\
\sum_{S \in V \cup F \cup uv \in \delta(S)} \Pi(S) &\leq c(uv) \ \forall uv \in E \\
\Pi(S) &\geq 0 \ \forall S \in F
\end{align*}
\]

1. Perturb the integral cost function by adding $\frac{1}{2i}$ to edge $i$
2. **Starting LP.** Bipartite relaxation $(F = \emptyset)$
3. Repeat until $x$ is integral
   (a) **Retain old cuts.** Find a specific dual optimal solution $\Pi$ to $D_F$.
      \[H' = \{ S \in F : \Pi(S) > 0 \}\]
   (b) **Choose new cuts.** For each cycle $C \in \text{supp}(x)$, define $\hat{C}$ as the union of $V(C)$ and the inclusionwise maximal sets of $H'$ intersecting $V(C)$
      \[H'' = \{ \hat{C} : C \in \text{supp}(x) \}\]
   (c) Set the new $F = H' \cup H''$
   (d) **Re-solve LP.** Find an optimal solution $x$ to $P_F$
Analysis Overview

(i) **Half-integral Structure:** Intermediate LP optima are half-integral and supported by a disjoint union of odd cycles and edges.

(ii) **Cut Retention:** If \( \text{odd}(x) \) remains the same in iterations \( i, i+1, \ldots, j \) then all cuts added in iterations \( i, i+1, \ldots, j \) are retained up to the \( j \)'th iteration.

**Proof of Cut Retention:** Coupling with the intermediate solutions of a new Half-Integral Primal-Dual Algorithm for matching.

- The choice of specific dual optimal solution to retain cuts comes from this coupling.
Half-Integral Primal-Dual Algorithm

Edmonds’ Algorithm

- Intermediate Primal – integral
- Unique way to augment primal
- Build an alternating tree using tight edges and repeatedly attempt to augment primal and change dual values until there are no more exposed nodes
- Deshrink if dual value on a set decreases to zero
- Shrink if a set forms a blossom

Half-Integral Algorithm

- Intermediate Primal – half-integral
- 3 ways to augment primal
- Build an alternating tree using tight edges and repeatedly attempt to augment primal and change dual values until there are no more exposed nodes
- Deshrink if dual value on a set decreases to zero
- Augment primal if a set forms a blossom
Half-Integral Primal-Dual Algorithm

Primal Augmentations
Half-Integral Primal-Dual Algorithm

**Edmonds’ Algorithm**

- Intermediate Primal – integral
- Unique way to augment primal
- Build an alternating tree using tight edges and repeatedly attempt to augment primal and change dual values until there are no more exposed nodes
- Deshrink if dual value on a set decreases to zero
- Shrink if a set forms a blossom

**Half-Integral Algorithm**

- Intermediate Primal – half-integral
- 3 ways to augment primal
- Build an alternating tree using tight edges and repeatedly attempt to augment primal and change dual values until there are no more exposed nodes
- Deshrink if dual value on a set decreases to zero
- **Augment primal** if a set forms a blossom
Half-Integral Primal-Dual Algorithm

**Edmonds’ Algorithm**

- Intermediate Primal – integral
- Unique way to augment primal
- Build an alternating tree using tight edges and repeatedly attempt to augment primal and change dual values until there are no more exposed nodes
- Deshrink if dual value on a set decreases to zero
- Shrink if a set forms a blossom

**Half-Integral Algorithm**

- Intermediate Primal – half-integral
- 3 ways to augment primal
- Build an alternating tree using tight edges and repeatedly attempt to augment primal and change dual values until there are no more exposed nodes
- Deshrink if dual value on a set decreases to zero
- **Augment primal** if a set forms a blossom
- After covering all exposed nodes, shrink all odd cycles to exposed nodes and proceed again until no more odd cycles
Summary

- Main ingredients
  - Drop cuts with zero dual values
    - Common in implementations of cutting plane method to ensure LPs do not blow up in size
  - Add cuts to maintain laminarity

- Tools from the analysis
  - New polyhedral results about the matching polytope
    - Solution to the LP with some odd-set inequalities is half-integral provided certain conditions are satisfied
  - New combinatorial algorithm for matching
    - An alternate primal-dual algorithm for matching where the intermediate solutions are half-integral
Future Directions

- Implications of the Dual-based cut-retention procedure for other poly-time solvable combinatorial problems
  - Combinatorial polytopes with Chvátal rank one (Edmonds-Johnson matrices)

- Efficient cutting plane algorithms for optimization over
  - intersection of two matroid polytopes
  - subtour elimination polytope

Happy Birthday, András!