Interior Point Cutting Plane and Column Generation Methods

John E. Mitchell

Department of Mathematical Sciences
RPI, Troy, NY 12180 USA

October 12, 2009

Collaborators: Luc Basescu, Brian Borchers, Mohammad Oskoorouchi, Srini Ramaswamy, Kartik Sivaramakrishnan, Mike Todd
Outline

1. Introduction
2. MaxCut
3. Interior point cutting plane methods
4. Warm starting
5. Theoretical results
6. Stabilization
7. Conclusions
The Traveling Salesman Problem

Lower bounds determined using cutting planes and branch-and-cut.

TSP Webpage
Bill Cook, Georgia Tech

Robert Bosch
Oberlin College
Want to solve the integer programming problem

\[
\begin{align*}
\max_{y \in \mathbb{R}^m} & \quad b^T y \\
\text{subject to} & \quad A^T y \leq c \\
& \quad y_i \quad \text{integer for } i \in I
\end{align*}
\]
LP Relaxation

Relax integrality restriction:

Dual problem:
\[
\max_{y \in \mathbb{R}^m} \quad b^T y \\
\text{s.t.} \quad A^T y \leq c \\
a_0^T y \leq c_0
\]

Primal problem:
\[
\min_{x \in \mathbb{R}^n, x_0} \quad c^T x + c_0 x_0 \\
\text{s.t.} \quad Ax + a_0 x_0 = b \\
\quad x \geq 0
\]

Add a cutting plane to the dual problem

Corresponds to column generation in the primal.
LP Relaxation

Relax integrality restriction:

Dual problem:

\[
\begin{align*}
\max_{y \in \mathbb{R}^m} & \quad b^T y \\
\text{s.t.} & \quad A^T y \leq c \\
& \quad a_0^T y \leq c_0
\end{align*}
\]

Primal problem:

\[
\begin{align*}
\min_{x \in \mathbb{R}^n, x_0} & \quad c^T x + c_0 x_0 \\
\text{s.t.} & \quad Ax + a_0 x_0 = b \\
& \quad x \geq 0
\end{align*}
\]

Add a cutting plane to the dual problem

Corresponds to column generation in the primal.
LP Relaxation

Relax integrality restriction:

Dual problem:

\[
\begin{align*}
\max_{y \in \mathbb{R}^m} & \quad b^T y \\
\text{s.t.} & \quad A^T y \leq c \\
& \quad a_0^T y \leq c_0
\end{align*}
\]

Primal problem:

\[
\begin{align*}
\min_{x \in \mathbb{R}^n, x_0} & \quad c^T x + c_0 x_0 \\
\text{s.t.} & \quad Ax + a_0 x_0 = b \\
& \quad x, \quad x_0 \geq 0
\end{align*}
\]

Add a cutting plane to the dual problem

Corresponds to column generation in the primal.
LP Relaxation

Relax integrality restriction:

Dual problem:

\[
\begin{align*}
\max_{y \in \mathbb{R}^m} & \quad b^T y \\
\text{s.t.} & \quad A^T y \leq c \\
& \quad a_0^T y \leq c_0
\end{align*}
\]

Primal problem:

\[
\begin{align*}
\min_{x \in \mathbb{R}^n, x_0} & \quad c^T x + c_0 x_0 \\
\text{s.t.} & \quad Ax + a_0 x_0 = b \\
& \quad x, x_0 \geq 0
\end{align*}
\]

Add a cutting plane to the dual problem

Corresponds to column generation in the primal.
Cutting plane algorithm illustration

\[ \{ y \in \mathbb{R}^m : A^T y \leq c \} \]

Solution to LP relaxation
Cutting plane algorithm illustration

\[ \{ y \in \mathbb{R}^m : A^T y \leq c \} \]
Cutting plane algorithm illustration

\[ \{ y \in \mathbb{R}^m : A^T y \leq c, \ a^T_1 y \leq c_1 \} \]
Cutting plane algorithm illustration

\[ \{ y \in \mathbb{R}^m : A^T y \leq c, \; a_1^T y \leq c_1 \} \]

Cutting plane

\[ a_2^T y = c_2 \]

Soln to LP relaxation
Cutting plane algorithm illustration

\[ \{ y \in \mathbb{R}^m : A^T y \leq c, \ a_1^T y \leq c_1, \ a_2^T y \leq b_2 \} \]

Solution to LP relaxation
Cutting plane algorithm illustration

\[ \{ y \in \mathbb{R}^m : A^T y \leq c, a_1^T y \leq c_1, a_2^T y \leq b_2 \} \]

Solution to LP relaxation

Solution is integral, so optimal to IP
Outline

1. Introduction
2. MaxCut
3. Interior point cutting plane methods
4. Warm starting
5. Theoretical results
6. Stabilization
7. Conclusions
Valid constraints for MaxCut

Find the maximum cut in a graph.


Interactions known. Deduce spins.

\[ y_{uv} = \begin{cases} 1 & u, v \text{ opposite spins} \\ 0 & u, v \text{ same spin} \end{cases} \]

\( y \leftrightarrow \) incidence vector of cut.

Any cut and any cycle intersect in an even number of edges.

Valid constraint:
\[ y_e + y_f + y_g - y_h \leq 2 \]
Valid constraints for MaxCut

Find the maximum cut in a graph.


Interactions known. Deduce spins.

\[ y_{uv} = \begin{cases} 1 & u, v \text{ opposite spins} \\ 0 & u, v \text{ same spin} \end{cases} \]

\( y \leftrightarrow \text{incidence vector of cut.} \)

Any cut and any cycle intersect in an even number of edges.

Valid constraint:

\[ y_e + y_f + y_g - y_h \leq 2 \]
**Integer programming formulation**

\[
\text{max } \quad \sum_{e \in E} b_e y_e \\
\text{subject to } \quad y \text{ satisfies all cut-cycle inequalities} \quad (IP)
\]

\[
\text{max } \quad \sum_{e \in E} b_e y_e \\
\text{subject to } \quad y \text{ satisfies all cut-cycle inequalities} \\
\quad y \text{ satisfies additional linear constraints} \\
\quad 0 \leq y \leq 1 \quad (LP)
\]

Algorithm:

1. Initialize: just the box constraints \( 0 \leq y \leq 1 \).
2. Solve LP relaxation
3. Add violated constraints as required.
4. If not yet converged, return to Step 2.

Can typically solve 100x100 grids in about 5 minutes with \( b_e = \pm 1 \).
**Integer programming formulation**

$$\max \sum_{e \in E} b_e y_e$$
subject to

- $y$ satisfies all cut-cycle inequalities
- $y$ binary

$$\max \sum_{e \in E} b_e y_e$$
subject to

- $y$ satisfies all cut-cycle inequalities
- $y$ satisfies additional linear constraints
- $0 \leq y \leq 1$

**Algorithm:**

1. Initialize: just the box constraints $0 \leq y \leq 1$.
2. Solve LP relaxation
3. Add violated constraints as required.
4. If not yet converged, return to Step 2.

Can typically solve 100x100 grids in about 5 minutes with $b_e = \pm 1$. 
Integer programming formulation

\[
\begin{align*}
\text{max} & \quad \sum_{e \in E} b_e y_e \\
\text{subject to} & \quad y \text{satisfies all cut-cycle inequalities} \\
& \quad y \text{ binary}
\end{align*}
\]

\[
\begin{align*}
\text{max} & \quad \sum_{e \in E} b_e y_e \\
\text{subject to} & \quad y \text{satisfies all cut-cycle inequalities} \\
& \quad y \text{satisfies additional linear constraints} \\
& \quad 0 \leq y \leq 1
\end{align*}
\]  

Algorithm:

1. Initialize: just the box constraints \(0 \leq y \leq 1\).
2. Solve LP relaxation
3. Add violated constraints as required.
4. If not yet converged, return to Step 2.

Can typically solve 100x100 grids in about 5 minutes with \(b_e = \pm 1\).
Integer programming formulation

\[
\begin{align*}
\text{max} & \quad \sum_{e \in E} b_e y_e \\
\text{subject to} & \quad y \text{ satisfies all cut-cycle inequalities} \\
& \quad y \text{ binary}
\end{align*}
\]

\[
\begin{align*}
\text{max} & \quad \sum_{e \in E} b_e y_e \\
\text{subject to} & \quad y \text{ satisfies all cut-cycle inequalities} \\
& \quad y \text{ satisfies additional linear constraints} \\
& \quad 0 \leq y \leq 1
\end{align*}
\]

Algorithm:

1. Initialize: just the box constraints \(0 \leq y \leq 1\).
2. Solve LP relaxation
3. Add violated constraints as required.
4. If not yet converged, return to Step 2.

Can typically solve 100x100 grids in about 5 minutes with \(b_e = \pm 1\).
Computational results

Compare interior point and simplex cutting plane algorithms. Times in seconds (rounded). \( b_e = \pm 1 \). One problem of each size.

<table>
<thead>
<tr>
<th>Grid</th>
<th>Interior</th>
<th>Simplex</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>50x50</td>
<td>4</td>
<td>18</td>
<td>4.87</td>
</tr>
<tr>
<td>60x60</td>
<td>31</td>
<td>271</td>
<td>8.90</td>
</tr>
<tr>
<td>70x70</td>
<td>54</td>
<td>786</td>
<td>14.75</td>
</tr>
<tr>
<td>80x80</td>
<td>51</td>
<td>639</td>
<td>12.72</td>
</tr>
<tr>
<td>90x90</td>
<td>223</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>100x100</td>
<td>187</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

**Interior point:**
one core of Mac Pro with 2x2.8 GHz Quad-Core Intel Xeon. Personal code.

**Simplex** (spin glass server):
Intel(R) Celeron(R) M CPU 440 @ 1.86GHz. CPLEX 9.1.

*Interior* is faster, with the ratio getting better as problem size increases.
Outline

1. Introduction
2. MaxCut
3. Interior point cutting plane methods
4. Warm starting
5. Theoretical results
6. Stabilization
7. Conclusions
Interior point cutting plane methods

Solve the LP relaxations using **interior point methods**: good for large scale LPs.

These methods work well when a lot of violated cuts can be added at once.

The LP relaxations need only be solved approximately, so the **resulting cutting planes can be more centered**.

**Combining interior point and simplex methods** is especially effective: use the interior point method initially; switch to simplex once sufficiently close to optimality.
Comparing the strength of simplex and interior point cutting planes

Simplex:

- Optimal vertex found by simplex
- Added cutting plane when using simplex

Interior point method:

- Central trajectory
- Optimal face
- Added cutting plane when using interior point method
Comparing the strength of simplex and interior point cutting planes

**Simplex:**
- Optimal vertex found by simplex
- Added cutting plane when using simplex

**Interior point method:**
- Interior point iterate
- Optimal face
- Central trajectory
- Added cutting plane when using interior point method
Linear ordering problems

Ratio: Combined time / Simplex time

Different problem classes:
- 0% zeroes
- 10% zeroes
- 20% zeroes
- Type B
- Type C

Simplex time (secs)
Outline

1. Introduction
2. MaxCut
3. Interior point cutting plane methods
4. Warm starting
5. Theoretical results
6. Stabilization
7. Conclusions
Warm starting an interior point method

Can’t be done
Warm starting an interior point method

Can be done to some degree!

Can reduce iteration counts by 50%, perhaps more.
Warm starting an interior point method

Can be done to some degree!

Can reduce iteration counts by 50%, perhaps more.
Warm starting an interior point method

Can be done to some degree!

Can reduce iteration counts by 50%, perhaps more.
Warm starting an interior point method

Can be done to some degree!

Can reduce iteration counts by 50%, perhaps more.
Warm starting an interior point method

Can be done to some degree!

Can reduce iteration counts by 50%, perhaps more.

Need to make some effort to recover centrality instead of immediately aiming for new optimal solution.

Useful to store potential restart points: earlier iterates, or earlier approximate analytic centers (Gondzio), or problem-specific points.
Warm starting an interior point method

Can be done to some degree!

Can reduce iteration counts by 50%, perhaps more.

Need to make some effort to recover centrality instead of immediately aiming for new optimal solution.

Useful to store potential restart points: earlier iterates, or earlier approximate analytic centers (Gondzio), or problem-specific points.

Want to get to final straighter portion of trajectory.
WARM STARTING

Using Dikin ellipsoid to restart

Current relaxation and iterate

\[ A^T y \leq c \]
Using Dikin ellipsoid to restart

Dikin ellipsoid: inscribed ellipsoid centered at $\bar{y}$

$$\sum \frac{(s_i - \bar{s}_i)^2}{\bar{s}_i^2} \leq 1$$

$$1 \leq \sum (s_i - \bar{s}_i)/(\bar{s}_i)$$
Using Dikin ellipsoid to restart

Add cutting plane

\[ a_0^T y = a_0^T \bar{y} \]
Warm starting

Using Dikin ellipsoid to restart

Zoom in . . .

Use Dikin ellipsoid as proxy for old constraints
Warm starting

**Dikin Ellipsoid**

Find restart direction $d$

$E^D(\bar{y}, \bar{s})$

$a_0^T \bar{y} = a_0^T \bar{\bar{y}}$

Move as far off the added constraint as possible, while staying within the Dikin ellipsoid.
Another Dikin Ellipsoid

Find restart direction $d$
when adding multiple constraints

\[ E^D(\bar{y}, \bar{s}) \]

\[ \text{Max } \pi \text{ (new slacks)} \]
\[ \text{r.t. old slacks in Dikin ellipsoid.} \]

Minimize the potential function of the new slack variables, $-\sum \ln s_i$,
while staying within the Dikin ellipsoid. (Goffin and Vial)
Primal restart

\[
\begin{align*}
\min_{x \in \mathbb{R}^n, x_0} & \quad c^T x + c_0 x_0 \\
\text{s.t.} & \quad A x + a_0 x_0 = b \\
& \quad x, x_0 \geq 0
\end{align*}
\]

Restart from a point \( x = \bar{x}, x_0 = 0 \).

Using a scaling matrix \( D \), get direction

\[
\Delta x := -D^2 A^T (A D^2 A^T)^{-1} a_0
\]

This can be derived from the Dikin ellipsoid.

Can be generalized to the addition of multiple columns.
Primal restart

\[ (x - \bar{x})^T D^{-2} (x - \bar{x}) \leq 1. \]

\[
\begin{align*}
\min_{x \in \mathbb{R}^n, x_0} & \quad c^T x + c_0 x_0 \\
\text{s.t.} & \quad A x + a_0 x_0 = b \\
& \quad x, x_0 \geq 0
\end{align*}
\]

Restart from a point \( x = \bar{x}, x_0 = 0 \).

Using a scaling matrix \( D \), get direction

\[
\Delta x := -D^2 A^T (A D^2 A^T)^{-1} a_0
\]

This can be derived from the Dikin ellipsoid.

Can be generalized to the addition of multiple columns.
Outline

1. Introduction
2. MaxCut
3. Interior point cutting plane methods
4. Warm starting
5. Theoretical results
6. Stabilization
7. Conclusions
Theoretical results for the convex feasibility problem

Convex feasibility problem:

Given convex set $C$ and a polyhedral outer approximation to $\mathbb{Y}$, either find a point in $\mathbb{Y}$ or determine $\mathbb{Y}$ is empty.
Theoretical results for the convex feasibility problem

Convex feasibility problem:

Given convex set $C$ and a polyhedral outer approximation to $Y$, either find a point in $Y$ or determine $Y$ is empty.
Theoretical results for the convex feasibility problem

Convex feasibility problem:

Given convex set $C$ and a polyhedral outer approximation to $Y$, either find a point in $Y$ or determine $Y$ is empty.
Theoretical results

Theoretical results for the convex feasibility problem

Convex feasibility problem:

Given convex set $C$ and a polyhedral outer approximation to $Y$, either find a point in $Y$ or determine $Y$ is empty.
Theoretical results

Convergence theorem (Goffin and Vial)

Use interior point cutting plane algorithm.

Use approximate analytic centers as iterates.

Add up to \( p \) cuts at each call to oracle. Never drop cuts.

\[
\text{max } \sum_{i \in S} \ln x_i \\
\text{st. } A^2 y + s = c \\
(5 \geq 0)
\]

\[
\text{F}
\]

\[
\text{FULLY POLYNOMIAL.}
\]

\[
\text{(not polynomial)}
\]

\[
\text{O} - \text{dimension of } g \text{.}
\]

\[
\text{m: dimension of } g. \\
\text{Total } \# \text{ Newton steps}.
\]

Theorem

If \( C \) contains a ball of radius \( \varepsilon \) then a certain interior point cutting plane method converges after adding no more than \( O\left(\frac{m^2 p^2}{\varepsilon^2}\right) \) cutting planes.

Each recentering step requires at most \( O(p \ln p) \) Newton steps.
Convergence theorems (Atkinson and Vaidya)

Also allow dropping of constraints. Add at most one cut at a time.

**Theorem**

If $C$ contains a ball of radius $\varepsilon$ then an interior point variant converges after adding no more than $O(m \ln(\frac{1}{\varepsilon})^2)$ iterations, when extra conditions are used to determine constraints to add and drop.

Volumetric center cutting plane methods:

Add or drop one constraint at a time.

**Theorem**

If $C$ contains a ball of radius $\varepsilon$ then an interior point variant stops in $O(m \ln(\frac{1}{\varepsilon}))$ calls to the oracle and $O(m \ln(\frac{1}{\varepsilon}))$ approximate Newton steps.
Convergence theorems (Atkinson and Vaidya)

Also allow **dropping of constraints**. Add at most one cut at a time.

**Theorem**

*If C contains a ball of radius $\varepsilon$ then an interior point variant converges after adding no more than $O(m \ln(\frac{1}{\varepsilon}^2))$ iterations, when extra conditions are used to determine constraints to add and drop.*

Volumetric center cutting plane methods:

Add or drop one constraint at a time.

**Theorem**

*If C contains a ball of radius $\varepsilon$ then an interior point variant stops in $O(m \ln(\frac{1}{\varepsilon}))$ calls to the oracle and $O(m \ln(\frac{1}{\varepsilon}))$ approximate Newton steps.*
Outline

1. Introduction
2. MaxCut
3. Interior point cutting plane methods
4. Warm starting
5. Theoretical results
6. Stabilization
7. Conclusions
Links with Stabilization

Maximize concave function $f(y)$.

Subgradient inequality: $f(y) \leq f(\bar{y}) + \xi^T(y - \bar{y})$
Links with Stabilization

Maximize concave function $f(y)$.

Subgradient inequality: $f(y) \leq f(\bar{y}) + \xi^T(y - \bar{y})$
Links with Stabilization

Maximize concave function $f(y)$.

Subgradient inequality: $f(y) \leq f(\bar{y}) + \xi^T(y - \bar{y})$

Maximize piecewise linear overestimator
**Links with Stabilization**

Maximize concave function $f(y)$.  
Subgradient inequality: $f(y) \leq f(\bar{y}) + \xi^T(y - \bar{y})$
Links with Stabilization

Maximize concave function $f(y)$.
Subgradient inequality: $f(y) \leq f(\bar{y}) + \xi^T(y - \bar{y})$
**Links with Stabilization**

Maximize concave function $f(y)$.  
Subgradient inequality: $f(y) \leq f(\bar{y}) + \xi^T(y - \bar{y})$
Maximize concave function $f(y)$.
Subgradient inequality: $f(y) \leq f(\bar{y}) + \xi^T(y - \bar{y})$
Stabilization / regularization

Iterates jump around if solve piecewise linear overestimator to optimality.

Try to stabilize the process.

One possibility: subtract proximal term $\frac{u}{2}||y - \bar{y}||^2$ from objective. Used in bundle methods.

Alternative: use interior point method and only solve the relaxations approximately.
Stabilization with interior point methods
Stabilization with interior point methods

\[ \max \{ f(y^2), f(y') \} \]

lower bound
Stabilization with interior point methods

Find approximate analytic center

lower bound
Stabilization with interior point methods

Find approximate analytic center
Stabilization with interior point methods

Find approximate analytic center
Outline

1. Introduction
2. MaxCut
3. Interior point cutting plane methods
4. Warm starting
5. Theoretical results
6. Stabilization
7. Conclusions
Conclusions for Cutting Plane Methods

- Interior point methods provide an excellent choice for stabilizing a column generation approach.
- The strength of interior point methods for linear programming means that these column generation approaches scale well, with theoretical polynomial or fully polynomial convergence depending on the variant.
- Combining interior point and simplex column generation methods has proven especially effective, with the interior point methods used early on and the simplex method used later. The development of methods that automatically combine the two linear programming approaches would be useful.
- Theoretical development of a more intuitive polynomial interior point cutting plane algorithm is desirable.
Conclusions for Cutting Plane Methods

- Interior point methods provide an excellent choice for **stabilizing** a column generation approach.
- The strength of interior point methods for linear programming means that these column generation approaches **scale well**, with theoretical **polynomial** or fully polynomial convergence depending on the variant.
- Combining interior point and simplex column generation methods has proven especially effective, with the interior point methods used early on and the simplex method used later. The development of methods that **automatically combine** the two linear programming approaches would be useful.
- Theoretical development of a **more intuitive polynomial** interior point cutting plane algorithm is desirable.
Conclusions for Cutting Plane Methods

- Interior point methods provide an excellent choice for stabilizing a column generation approach.

- The strength of interior point methods for linear programming means that these column generation approaches scale well, with theoretical polynomial or fully polynomial convergence depending on the variant.

- Combining interior point and simplex column generation methods has proven especially effective, with the interior point methods used early on and the simplex method used later. The development of methods that automatically combine the two linear programming approaches would be useful.

- Theoretical development of a more intuitive polynomial interior point cutting plane algorithm is desirable.
Conclusions for Cutting Plane Methods

- Interior point methods provide an excellent choice for stabilizing a column generation approach.
- The strength of interior point methods for linear programming means that these column generation approaches scale well, with theoretical polynomial or fully polynomial convergence depending on the variant.
- Combining interior point and simplex column generation methods has proven especially effective, with the interior point methods used early on and the simplex method used later. The development of methods that automatically combine the two linear programming approaches would be useful.
- Theoretical development of a more intuitive polynomial interior point cutting plane algorithm is desirable.