Interior Point Methods for Mathematical Programming

Clóvis C. Gonzaga

Federal University of Santa Catarina, Florianópolis, Brazil

EURO - 2013 — Roma
Our heroes

Cauchy

Newton

Lagrange
Early results

Unconstrained minimization

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\end{align*}
\]

Inequality constraints

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g(x) \leq 0
\end{align*}
\]

- Cauchy (steepest descent): \( x^+ = x - \lambda \nabla f(x) \)
- Newton: \( x^+ = x - H(x)^{-1} \nabla f(x) \)
- For the constrained problem: Frisch, Fiacco, McCormick:

\[
F(x) = f(x) - \mu \sum \log(-g_i(x))
\]
Early results

<table>
<thead>
<tr>
<th>Unconstrained minimization</th>
<th>Inequality constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize $f(x)$</td>
<td>minimize $f(x)$ subject to $g(x) \leq 0$</td>
</tr>
</tbody>
</table>

- Cauchy (steepest descent): $x^+ = x - \lambda \nabla f(x)$
- Newton: $x^+ = x - H(x)^{-1} \nabla f(x)$

For the constrained problem: Frisch, Fiacco, McCormick:

$$F(x) = f(x) - \mu \sum \log(-g_i(x))$$
Early results

Unconstrained minimization

\[
\text{minimize } f(x)
\]

Inequality constraints

\[
\begin{align*}
\text{minimize } & f(x) \\
\text{subject to } & g(x) \leq 0
\end{align*}
\]

- Cauchy (steepest descent): \( x^+ = x - \lambda \nabla f(x) \)
- Newton: \( x^+ = x - H(x)^{-1} \nabla f(x) \)
- For the constrained problem: Frisch, Fiacco, McCormick:

\[
F(x) = f(x) - \mu \sum \log(-g_i(x))
\]
The barrier function
Near the boundary, the barrier function tends to $+\infty$. 
### The Linear Programming Problem

#### Inequality constraints

- **Minimize**: $c^T x$
- **Subject to**: $Ax \leq b$

#### Equality and non-negativity

- **Minimize**: $c^T x$
- **Subject to**: $Ax = b$, $x \geq 0$

- **Interior point**: $x > 0$.
- **Solution**: $Ax = b$.
- **Feasible solution**: $Ax = b$, $x \geq 0$.
- **Interior (feasible) solution**: $Ax = b$, $x > 0$. 
The Linear Programming Problem

<table>
<thead>
<tr>
<th>Inequality constraints</th>
<th>Equality and non-negativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize $c^T x$</td>
<td>minimize $c^T x$</td>
</tr>
<tr>
<td>subject to $Ax \leq b$</td>
<td>subject to $Ax = b$</td>
</tr>
<tr>
<td></td>
<td>$x \geq 0$</td>
</tr>
</tbody>
</table>

- Interior point: $x > 0$.
- Solution: $Ax = b$.
- Feasible solution: $Ax = b$, $x \geq 0$.
- Interior (feasible) solution: $Ax = b$, $x > 0$. 
The Linear Programming Problem

Inequality constraints

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b
\end{align*}
\]

Equality and non-negativity

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

- Interior point: \( x > 0 \).
- Solution: \( Ax = b \).
- Feasible solution: \( Ax = b, \ x \geq 0 \).
- Interior (feasible) solution: \( Ax = b, \ x > 0 \).
The Linear Programming Problem

**Inequality constraints**

- minimize $c^T x$
- subject to $Ax \leq b$

**Equality and non-negativity**

- minimize $c^T x$
- subject to $Ax = b$
- $x \geq 0$

- Interior point: $x > 0$.
- Solution: $Ax = b$.
- Feasible solution: $Ax = b$, $x \geq 0$.
- Interior (feasible) solution: $Ax = b$, $x > 0$. 
Simplex Method: Dantzig 1948

Exponential in the worst case
Good for usual problems.
Centers of gravity
Centers of gravity

Conclusions

- Worst case performance: $O(n \ln(1/\varepsilon))$ calls to the oracle.
Centers of gravity

Conclusions

- Worst case performance: \( O(n \ln(1/\epsilon)) \) calls to the oracle.
- The method is generally not realizable: too difficult to compute the center of gravity.
  Work with simpler shapes: ellipsoids.
The computation of each new ellipsoid needs $O(n^2)$ arithmetical operations.

Worst case performance: $O(n^2 \ln(1/\varepsilon))$ calls to the oracle, and $O(n^4 \ln(1/\varepsilon))$ operations.
Khachiyan applied the ellipsoidal method to LP, and defined the number 
\( L = \text{number of bits in the input} \) for a problem with rational data.

A solution with precision \( 2^{-L} \) can be “purified” to an exact vertex solution of LP. 
Performance: \( O(n^4) L \) computations.
Time Magazine Joins In

Folding the Perfect Corner
A young Bell scientist makes a major math breakthrough

By John M. Nichols

For the past 10 years, mathematicians have been working on a problem that has been困扰ing them: how to fold a piece of paper into a perfect corner. The problem is not just a mathematical curiosity, but has practical applications in fields such as engineering and physics. In the past, mathematicians have struggled to find a solution, but a young scientist at Bell Labs has made a breakthrough. His work has been recognized by the Mathematical Association of America, and he has been invited to speak at a major conference. The scientist, who has been working on the problem for several years, is excited about the potential applications of his work. "This is just the beginning," he said. "I'm sure there are many more applications of this technology that we haven't even thought of yet."
Karmarkar 1984: projective IP method: $O(n^{3.5}L)$ computations.
Renegar 1986: path following algorithm: $O(n^{3.5}L)$ computations.
Vajdya and G. (independently) 1987: path following algorithms: $O(n^3L)$ computations.
Extensions

- Linear programming and quadratic programming:
  Jaceck Gonzio solved a Portfolio Optimization problem based on Markowitz analysis using 16 million scenarios, 1.10 billion variables, 353 million constraints in 53 iterations of an IP method in 50 minutes of a system of 1280 parallel processors.

- Second Order Cone Programming: extension of the Lorentz cone. Applications to Quadratically constrained quadratic programming, robust LP, robust Least Squares, Filter design, Portfolio Optimization, Truss design,...

- Semi-definite programming.
  A huge body of theory and applications to Structural Optimization, Optimal Control, Combinatorial Optimization, ...
Extensions

- **Linear programming and quadratic programming:**
  Jaceck Gonzio solved a Portfolio Optimization problem based on Markowitz analysis using 16 million scenarios, 1.10 billion variables, 353 million constraints in 53 iterations of an IP method in 50 minutes of a system of 1280 parallel processors.

- **Second Order Cone Programming:** extension of the Lorentz cone. Applications to Quadratically constrained quadratic programming, robust LP, robust Least Squares, Filter design, Portfolio Optimization, Truss design, ...

- **Semi-definite programming.**
  A huge body of theory and applications to Structural Optimization, Optimal Control, Combinatorial Optimization, ...
Extensions

- Linear programming and quadratic programming:
  Jaceck Gonzio solved a Portfolio Optimization problem based on Markowitz analysis using 16 million scenarios, 1.10 billion variables, 353 million constraints in 53 iterations of an IP method in 50 minutes of a system of 1280 parallel processors.

- Second Order Cone Programming: extension of the Lorentz cone. Applications to Quadratically constrained quadratic programming, robust LP, robust Least Squares, Filter design, Portfolio Optimization, Truss design, ...

- Semi-definite programming.
  A huge body of theory and applications to Structural Optimization, Optimal Control, Combinatorial Optimization, ...
The Affine-Scaling direction

Projection matrix
Given \( c \in \mathbb{R}^n \) and a matrix \( A \),
can be decomposed as
\[
c = P_A c + A^T y,
\]
where \( P_A c \in \mathcal{N}(A) \) is the projection of \( c \) into \( \mathcal{N}(A) \).
The Affine-Scaling direction

Linearly constrained problem:

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

Define \( c = \nabla f(x^0) \). The projected gradient (Cauchy) direction is

\[ c_P = P_A c, \]

and the steepest descent direction is \( d = -c_P \). It solves the trust region problem

\[
\text{minimize}\{c^T h \mid Ah = 0, \|d\| \leq \Delta\}. 
\]
Largest ball centered in $x^k$
Largest simple ellipsoid centered in $x^k$
The affine scaling direction
Largest ball centered in $e = (1, 1, \ldots, 1)$
The Affine-Scaling direction

Given a feasible point $x_0$, $X = \text{diag}(x_0)$ and $c = \nabla f(x_0)$

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0 \\
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad (Xc)^T \bar{x} \\
\text{subject to} & \quad AX\bar{x} = b \\
& \quad \bar{x} \geq 0
\end{align*}
\]

Scaled steepest descent direction: $x = Xe$ ($x$ is scaled into $e$):

\[
\begin{align*}
\bar{d} & = -P_{AX}Xc \\
d & = X\bar{d} = -XP_{AX}Xc
\end{align*}
\]

Dikin’s direction:

\[
\begin{align*}
\bar{d} & = -P_{AX}Xc \\
d & = -X\bar{d}/\|\bar{d}\|.
\end{align*}
\]
Dikin’s algorithm

Problem P1
Dikin’s algorithm

Problem P1
Dikin’s algorithm
Dikin’s algorithm

Problem P1
Affine scaling algorithm
Affine scaling algorithm
Affine scaling algorithm

Problem P1
Affine scaling algorithm
The logarithmic barrier function

\[ x \in \mathbb{R}^n_+ \mapsto p(x) = -\sum_{i=1}^{n} \log x_i. \]

Scaling: for a diagonal matrix \( D > 0 \)

\[
\begin{align*}
    p(Dx) &= p(x) + \text{constant}, \\
    p(Dx) - p(Dy) &= p(x) - p(y).
\end{align*}
\]

Derivatives:

\[
\begin{align*}
    \nabla p(x) &= -x^{-1}, & \nabla p(e) &= -e, \\
    \nabla^2 p(x) &= X^{-2}, & \nabla^2 p(e) &= I.
\end{align*}
\]

At \( x = e \), the Hessian matrix is the identity, and hence the Newton direction coincides with the Cauchy direction.
The logarithmic barrier function

\[ x \in \mathbb{R}^n_+ \mapsto p(x) = - \sum_{i=1}^n \log x_i. \]

Scaling: for a diagonal matrix \( D > 0 \)

\[ p(Dx) = p(x) + \text{constant}, \]
\[ p(Dx) - p(Dy) = p(x) - p(y). \]

Derivatives:

\[ \nabla p(x) = -x^{-1}, \quad \nabla p(e) = -e \]
\[ \nabla^2 p(x) = X^{-2}, \quad \nabla^2 p(e) = I. \]

At \( x = e \), the Hessian matrix is the identity, and hence the Newton direction coincides with the Cauchy direction.

At any \( x > 0 \), the affine scaling direction coincides with the Newton direction.
The penalized function in linear programming

For \( x > 0, \mu > 0 \) and \( \alpha = 1/\mu \),

\[
 f_\alpha(x) = \alpha c^T x + p(x) \quad \text{or} \quad f_\mu(x) = c^T x + \mu p(x)
\]

minimize \( c^T x \) \hspace{2cm} \text{minimize} \quad c^T x + \mu p(x)

subject to \( Ax = b \) \hspace{2cm} \text{subject to} \quad Ax = b

\( x \geq 0 \) \hspace{2cm} x > 0

- For \( \alpha \geq 0 \), \( f_\alpha \) is strictly convex and grows indefinitely near the boundary of the feasible set.
- Whenever the minimizers exist, they are defined uniquely by

\[
 x_\alpha = \text{argmin}_{x \in \Omega} f_\alpha(x).
\]

- In particular, if \( \Omega \) is bounded, \( x_0 \) is the analytic center of \( \Omega \)
- If the optimal face of the problem is bounded, then the curve

\[
 \alpha > 0 \mapsto x_\alpha
\]

is well defined and is called the primal central path.
The penalized function in linear programming

For $x > 0$, $\mu > 0$ and $\alpha = 1/\mu$,

$$f_\alpha(x) = \alpha c^T x + p(x) \quad \text{or} \quad f_\mu(x) = c^T x + \mu p(x)$$

minimize $c^T x$
subject to $Ax = b$
$x \geq 0$

minimize $c^T x + \mu p(x)$
subject to $Ax = b$
$x > 0$

- For $\alpha \geq 0$, $f_\alpha$ is strictly convex and grows indefinitely near the boundary of the feasible set.
- Whenever the minimizers exist, they are defined uniquely by
  $$x_\alpha = \arg\min_{x \in \Omega} f_\alpha(x).$$

- In particular, if $\Omega$ is bounded, $x_0$ is the analytic center of $\Omega$
- If the optimal face of the problem is bounded, then the curve
  $$\alpha > 0 \mapsto x_\alpha$$
  is well defined and is called the primal central path.
The penalized function in linear programming

For $x > 0$, $\mu > 0$ and $\alpha = 1/\mu$,

$$f_{\alpha}(x) = \alpha c^T x + p(x) \quad \text{or} \quad f_{\mu}(x) = c^T x + \mu p(x)$$

minimize $c^T x$
subject to $Ax = b$
$x \geq 0$

minimize $c^T x + \mu p(x)$
subject to $Ax = b$
$x > 0$

For $\alpha \geq 0$, $f_{\alpha}$ is strictly convex and grows indefinitely near the boundary of the feasible set.

Whenever the minimizers exist, they are defined uniquely by

$$x_{\alpha} = \arg\min_{x \in \Omega} f_{\alpha}(x).$$

In particular, if $\Omega$ is bounded, $x_0$ is the analytic center of $\Omega$

If the optimal face of the problem is bounded, then the curve

$$\alpha > 0 \mapsto x_{\alpha}$$

is well defined and is called the primal central path.
The central path
Equivalent definitions of the central path

There are four equivalent ways of defining central points:

- Minimizers of the penalized function:
  \[
  \arg\min_{x \in \Omega} f_\alpha(x).
  \]

- Analytic centers of constant cost slices
  \[
  \arg\min_{x \in \Omega} \{ p(x) \mid c^T x = K \}
  \]

- Renegar centers: Analytic centers of \( \Omega \) with an extra constraint \( c^T x \leq \).
  \[
  \arg\min_{x \in \Omega} \{ p(x) - \log(K - c^T x) \mid c^T x < K \}
  \]

- Primal-dual central points (seen ahead).
Constant cost slices

Enter the new cut position (one point) and then the initial point
Constant cost slices

Enter the new cut position (one point) and then the initial point
Constant cost slices

Enter the new cut position (one point) and then the initial point
Renegar cuts
Renegar cuts
Renegar cuts

Problem Ptemp
Renegar cuts
Renegar cuts

Problem Ptemp
The most important problem in interior point methods is the following:

**Centering problem**

Given a feasible interior point \( x^0 \) and a value \( \alpha \geq 0 \), solve approximately the problem

\[
\text{minimize}_{x \in \Omega^0} \alpha c^T x + p(x).
\]

The Newton direction from \( x^0 \) coincides with the affine-scaling direction, and hence is the best possible. It is given by

\[
\begin{align*}
  d &= X \tilde{d}, \\
  \tilde{d} &= -P_{AX}X(\alpha c - x^{-1}) = -\alpha P_{AX}Xc + P_{AX}e.
\end{align*}
\]
The most important problem in interior point methods is the following:

Centering problem

Given a feasible interior point \( x^0 \) and a value \( \alpha \geq 0 \), solve approximately the problem

\[
\min_{x \in \Omega^0} \alpha c^T x + p(x).
\]

The Newton direction from \( x^0 \) coincides with the affine-scaling direction, and hence is the best possible. It is given by

\[
\begin{align*}
    d &= X\bar{d}, \\
    \bar{d} &= -P_{AX}X(\alpha c - x^{-1}) = -\alpha P_{AX}Xc + P_{AX}e.
\end{align*}
\]
Efficiency of the Newton step for centering

Newton direction:

\[
\begin{align*}
  d &= X\bar{d}, \\
  \bar{d} &= -P_{AX}X(\alpha c - x^{-1}) = -\alpha P_{AX}Xc + P_{AX}e.
\end{align*}
\]

We define the Proximity to the central point as

\[
\delta(x, \alpha) = \|\bar{d}\| = \| -\alpha P_{AX}Xc + P_{AX}e \|.
\]

The following important theorem says how efficient it is:

**Theorem**

Consider a feasible point \(x\) and a parameter \(\alpha\). Let \(x^+ = x + d\) be the point resulting from a Newton centering step. If \(\delta(x, \alpha) = \delta < 1\), then \(\delta(x^+, \alpha) < \delta^2\).

If \(\delta(x, \alpha) \leq 0.5\), then this value is a very good approximation to the euclidean distance between \(e\) and \(X^{-1}x_\alpha\), i.e., between \(x\) and \(x_\alpha\) in the scaled space.
Efficiency of the Newton step for centering

Newton direction:

\[ d = X\bar{d}, \]
\[ \bar{d} = -P_{AX}X(\alpha c - x^{-1}) = -\alpha P_{AX}Xc + P_{AX}e. \]

We define the Proximity to the central point as

\[ \delta(x, \alpha) = \|\bar{d}\| = \| -\alpha P_{AX}Xc + P_{AX}e \|. \]

The following important theorem says how efficient it is:

**Theorem**

Consider a feasible point \( x \) and a parameter \( \alpha \). Let \( x^+ = x + d \) be the point resulting from a Newton centering step. If \( \delta(x, \alpha) = \delta < 1 \), then \( \delta(x^+, \alpha) < \delta^2 \).

If \( \delta(x, \alpha) \leq 0.5 \), then this value is a very good approximation to the euclidean distance between \( e \) and \( X^{-1}x_\alpha \), i.e., between \( x \) and \( x_\alpha \) in the scaled space.
Short steps
Primal results as we saw are important to give a geometrical meaning to the procedures, and to develop the intuition. Also, these results can be generalized to a large class of problems, by generalizing the idea of barrier functions.

From now on we shall deal with primal-dual results, which are more efficient for linear and non-linear programming.
The Linear Programming Problem

<table>
<thead>
<tr>
<th>LP</th>
<th>LD</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize</td>
<td>maximize</td>
</tr>
<tr>
<td>( c^T x )</td>
<td>( b^T y )</td>
</tr>
<tr>
<td>subject to</td>
<td>subject to</td>
</tr>
<tr>
<td>( Ax = b )</td>
<td>( A^T y \leq c )</td>
</tr>
<tr>
<td>( x \geq 0 )</td>
<td></td>
</tr>
</tbody>
</table>
## The Linear Programming Problem

<table>
<thead>
<tr>
<th>LP</th>
<th>LD</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize $c^T x$</td>
<td>maximize $b^T y$</td>
</tr>
<tr>
<td>subject to $Ax = b$</td>
<td>subject to $A^T y + s = c$</td>
</tr>
<tr>
<td>$x \geq 0$</td>
<td>$s \geq 0$</td>
</tr>
</tbody>
</table>
The Linear Programming Problem

**LP**

minimize \( c^T x \)

subject to \( Ax = b \)

\( x \geq 0 \)

**KKT: multipliers \( y, s \)**

\( A^T y + s = c \)

\( Ax = b \)

\( xs = 0 \)

\( x, s \geq 0 \)

**LD**

maximize \( b^T y \)

subject to \( A^T y + s = c \)

\( s \geq 0 \)
The Linear Programming Problem

**LP**

minimize \( c^T x \)
subject to \( Ax = b \)
\( x \geq 0 \)

KKT: multipliers \( y, s \)

\[ A^T y + s = c \]
\[ Ax = b \]
\[ xs = 0 \]
\[ x, s \geq 0 \]

**LD**

maximize \( b^T y \)
subject to \( A^T y + s = c \)
\( s \geq 0 \)

KKT: multipliers \( x \)

\[ A^T y + s = c \]
\[ Ax = b \]
\[ xs = 0 \]
\[ x, s \geq 0 \]
The Linear Programming Problem

**LP**

<table>
<thead>
<tr>
<th>minimize</th>
<th>( c^T x )</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject to</td>
<td>( Ax = b )</td>
</tr>
<tr>
<td></td>
<td>( x \geq 0 )</td>
</tr>
</tbody>
</table>

**LD**

<table>
<thead>
<tr>
<th>maximize</th>
<th>( b^T y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject to</td>
<td>( A^T y + s = c )</td>
</tr>
<tr>
<td></td>
<td>( s \geq 0 )</td>
</tr>
</tbody>
</table>

**Primal-dual optimality**

| \( A^T y + s = c \) |
| \( Ax = b \) |
| \( xs = 0 \) |
| \( x, s \geq 0 \) |

**Duality gap**

For \( x, y, s \) feasible,

\[
c^T x - b^T y = x^T s \geq 0
\]
The Linear Programming Problem

<table>
<thead>
<tr>
<th>LP</th>
<th>LD</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize $c^T x$</td>
<td>maximize $b^T y$</td>
</tr>
<tr>
<td>subject to $Ax = b$</td>
<td>subject to $A^T y + s = c$</td>
</tr>
<tr>
<td>$x \geq 0$</td>
<td>$s \geq 0$</td>
</tr>
</tbody>
</table>

**Primal-dual optimality**

$$A^T y + s = c$$
$$Ax = b$$
$$xs = 0$$
$$x, s \geq 0$$

**Duality gap**

For $x, y, s$ feasible,

$$c^T x - b^T y = x^T s \geq 0$$

(LP) has solution $x$ and (LD) has solution $y, s$ if and only if the optimality conditions have solution $x, y, s$. 
Primal-dual centering

Let us write the KKT conditions for the centering problem (now using $\mu$ instead of $\alpha = 1/\mu$).

\[
\begin{align*}
\text{minimize} & \quad c^T x - \mu \sum \log x_i \\
\text{subject to} & \quad Ax = b \\
& \quad x > 0
\end{align*}
\]

Primal-dual center: KKT conditions

\[
\begin{align*}
x s &= \mu e \\
A^T y + s &= c \\
Ax &= b \\
x, s &> 0
\end{align*}
\]
Primal-dual centering

Let us write the KKT conditions for the centering problem (now using $\mu$ instead of $\alpha = 1/\mu$).

minimize $c^T x - \mu \sum \log x_i$
subject to $Ax = b$
$x > 0$

Primal-dual center: KKT conditions

$x s = \mu e$
$A^T y + s = c$
$Ax = b$
$x, s > 0$
Generalization

Let us write the KKT conditions for the convex quadratic programming problem

\[
\begin{align*}
\text{minimize} & \quad c^T x + \frac{1}{2} x^T H x \\
\text{subject to} & \quad Ax = b \\
& \quad x > 0
\end{align*}
\]

The first KKT condition is written as

\[
c + Hx - A^T y - s = 0
\]

To obtain a symmetrical formulation for the problem, we may multiply this equation by a matrix \( B \) whose rows form a basis for the null space of \( A \). Then \( BA^T y = 0 \), and we obtain the following conditions:

\[
\begin{align*}
x s & = 0 \\
-BHx + Bs & = Bc \\
Ax & = b \\
x, s & \geq 0
\end{align*}
\]
Horizontal linear complementarity problem

In any case, the problem can be written as

\[
\begin{align*}
xs & = 0 \\
Qx + Rs & = b \\
x, s & \geq 0
\end{align*}
\]

This is a linear complementarity problem, which includes linear and quadratic programming as particular problems. The techniques studied here apply to these problems, as long as the following monotonicity condition holds:

For any feasible pair of directions \((u, v)\) such that \(Qu + Rv = 0\), we have \(u^Tv \geq 0\).

The optimal face: the optimal solutions must satisfy \(x_is_i = 0\) for \(i = 1, \ldots, n\). This is a combinatorial constraint, responsible for all the difficulty in the solution.
Horizontal linear complementarity problem

In any case, the problem can be written as

\[
\begin{align*}
xs & = 0 \\
Qx + Rs & = b \\
x, s & \geq 0
\end{align*}
\]

This is a linear complementarity problem, which includes linear and quadratic programming as particular problems. The techniques studied here apply to these problems, as long as the following monotonicity condition holds:

For any feasible pair of directions \((u, v)\) such that \(Qu + Rv = 0\), we have \(u^T v \geq 0\).

The optimal face: the optimal solutions must satisfy \(x_i s_i = 0\) for \(i = 1, \ldots, n\). This is a combinatorial constraint, responsible for all the difficulty in the solution.
Primal-dual centering: the Newton step

Given $x, s$ feasible and $\mu > 0$, find

\[
\begin{align*}
x^+ &= x + u \\
s^+ &= s + v
\end{align*}
\]

such that

\[
\begin{align*}
x^+s^+ &= \mu e \\
Qx^+ + Rs^+ &= b \\
xs + su + xv + uv &= \mu e \\
Qu + Rv &= 0
\end{align*}
\]

Newton step

\[
\begin{align*}
Xv + Su &= -xs + \mu e \\
Qu + Rv &= 0
\end{align*}
\]

Solving this linear system is all the work. In the case of linear programming one should keep the multipliers $y$ and simplify the resulting system of equations.
Primal-dual centering: the Newton step

Given $x, s$ feasible and $\mu > 0$, find

$$
\begin{align*}
    x^+ &= x + u \\
    s^+ &= s + v
\end{align*}
$$

such that

$$
\begin{align*}
    x^+ s^+ &= \mu e \\
    Qx^+ + Rs^+ &= b \\
    xs + su + xv + uv &= \mu e \\
    Qu + Rv &= 0
\end{align*}
$$

Newton step

$$
\begin{align*}
    Xv + Su &= -xs + \mu e \\
    Qu + Rv &= 0
\end{align*}
$$

Solving this linear system is all the work. In the case of linear programming one should keep the multipliers $y$ and simplify the resulting system of equations.
Primal-dual centering: the Newton step

Given \( x, s \) feasible and \( \mu > 0 \), find

\[
\begin{align*}
x^+ & = x + u \\
s^+ & = s + v
\end{align*}
\]

such that

\[
\begin{align*}
x^+s^+ & = \mu e \\
Qx^+ + Rs^+ & = b \\
xs + su + xv + uv & = \mu e \\
Qu + Rv & = 0
\end{align*}
\]

Newton step

\[
\begin{align*}
Xv + Su & = -xs + \mu e \\
Qu + Rv & = 0
\end{align*}
\]

Solving this linear system is all the work. In the case of linear programming one should keep the multipliers \( y \) and simplify the resulting system of equations.
Primal-dual centering: Proximity measure

Given \( x, s \) feasible and \( \mu > 0 \), we want

\[
x s = \mu e \text{ or equivalently } \frac{x s}{\mu} - e = 0
\]

The actual error in this equation gives the proximity measure:

Proximity measure

\[
x, s, \mu \mapsto \delta(x, s, \mu) = \left\| \frac{x s}{\mu} - e \right\|.
\]

Theorem

Given a feasible pair \((x, s)\) and a parameter \(\mu\), Let \(x^+ = x + u\) and \(s^+ = s + v\) be the point resulting from a Newton centering step. If \(\delta(x, s, \mu) = \delta < 1\), then

\[
\delta(x^+, s^+, \mu) < \frac{1}{\sqrt{8}} \frac{\delta^2}{1 - \delta}.
\]

In particular, if \(\delta \leq 0.7\), then \(\delta(x^+, s^+, \mu) < \delta^2\).
Primal-dual centering: Proximity measure

Given $x, s$ feasible and $\mu > 0$, we want

$$xs = \mu e \text{ or equivalently } \frac{x_s}{\mu} - e = 0$$

The actual error in this equation gives the proximity measure:

<table>
<thead>
<tr>
<th>Proximity measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x, s, \mu \mapsto \delta(x, s, \mu) = | \frac{x_s}{\mu} - e |$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a feasible pair $(x, s)$ and a parameter $\mu$, Let $x^+ = x + u$ and $s^+ = s + v$ be the point resulting from a Newton centering step. If $\delta(x, s, \mu) = \delta &lt; 1$, then</td>
</tr>
<tr>
<td>$\delta(x^+, s^+, \mu) &lt; \frac{1}{\sqrt{8}} \frac{\delta^2}{1 - \delta}$.</td>
</tr>
<tr>
<td>In particular, if $\delta \leq 0.7$, then $\delta(x^+, s^+, \mu) &lt; \delta^2$.</td>
</tr>
</tbody>
</table>
Primal-dual path following: Traditional approach

- Assume that we have \( x, s, \mu \) such that \((x, s)\) is feasible and \( \delta(x, s, \mu) \leq \alpha < 1 \)
- Choose \( \mu^+ = \gamma \mu \), with \( \gamma < 1 \).
- Use Newton’s algorithm (with line searches to avoid infeasible points) to find \((x^+, s^+)\) such that \( \delta(x^+, s^+, \mu^+) \leq \alpha \)

Neighborhood of the central path

Given \( \beta \in (0, 1) \), we define the neighborhood \( \eta(\alpha) \) as the set of all feasible pairs \((x, s)\) such that for some \( \mu > 0 \)

\[
\delta(x, s, \mu) \leq \beta
\]

The methods must ensure that all points are in such a neighborhood, using line searches.
Primal-dual path following: Traditional approach

- Assume that we have $x, s, \mu$ such that $(x, s)$ is feasible and $\delta(x, s, \mu) \leq \alpha < 1$
- Choose $\mu^+ = \gamma \mu$, with $\gamma < 1$.
- Use Newton’s algorithm (with line searches to avoid infeasible points) to find $(x^+, s^+)$ such that $\delta(x^+, s^+, \mu^+) \leq \alpha$

Neighborhood of the central path

Given $\beta \in (0, 1)$, we define the neighborhood $\eta(\alpha)$ as the set of all feasible pairs $(x, s)$ such that for some $\mu > 0$

$$\delta(x, s, \mu) \leq \beta$$

The methods must ensure that all points are in such a neighborhood, using line searches.
Primal-dual path following: Traditional approach

- Assume that we have \( x, s, \mu \) such that \((x, s)\) is feasible and \( \delta(x, s, \mu) \leq \alpha < 1 \)
- Choose \( \mu^+ = \gamma \mu \), with \( \gamma < 1 \).
- Use Newton’s algorithm (with line searches to avoid infeasible points) to find \((x^+, s^+)\) such that \( \delta(x^+, s^+, \mu^+) \leq \alpha \)

Neighborhood of the central path
Given \( \beta \in (0, 1) \), we define the neighborhood \( \eta(\alpha) \) as the set of all feasible pairs \((x, s)\) such that for some \( \mu > 0 \)

\[
\delta(x, s, \mu) \leq \beta
\]

The methods must ensure that all points are in such a neighborhood, using line searches
A neighborhood of the central path
Short steps

Using $\gamma$ near 1, we obtain short steps. With $\gamma = 0.4/\sqrt{n}$, only one Newton step is needed at each iteration, and the algorithm is polynomial: it finds a solution with precision $2^{-L}$ in $O(\sqrt{nL})$ iterations.
Short steps
Large steps

Using $\gamma$ small, say $\gamma = 0.1$, we obtain large steps. This uses to work well in practice, but some sort of line search is needed, to avoid leaving the neighborhood. Predictor-corrector methods are better, as we shall see.
Large steps
Adaptive methods

Assume that \((x, s)\) feasible is given in \(\eta(\beta)\), but no value of \(\mu\) is given. Then we know:

- if \((x, s)\) is a central point, then \(xs = \mu e\) implies \(x^T s = n\mu\). Hence the best choice for \(\mu\) is \(\mu = x^T s / n\).
- If \((x, s)\) is not a central point, the value \(\mu(x, s) = x^T s / n\) gives a parameter value which in a certain sense is the best possible.
- An adaptive algorithm does not use a value of \(\mu\) coming from a former iteration: it computes \(\mu(x, s)\) and then chooses a value \(\gamma \mu(x, s)\) as new target.
- The target may be far. Compute a direction \((u, v)\) and follow it until

\[
\delta(x + \lambda u, s + \lambda v, \mu(x + \lambda u, s + \lambda v)) = \beta
\]
Predictor-corrector methods

Alternate two kinds of iterations:

- **Predictor**: An iteration starts with \((x, s)\) near the central path, and computes a Newton step \((u, v)\) with goal \(\gamma \mu(x, s)\), \(\gamma\) small.

  Follow it until
  \[
  \delta(x + \lambda u, s + \lambda v, \mu(x + \lambda u, s + \lambda v)) = \beta
  \]

- **Corrector**: Set \(x^+ = x + \lambda u, s^+ = s + \lambda v\), compute \(\mu = \mu(x^+, s^+)\) and take a Newton step with target \(\mu\)

If the predictor uses \(\gamma = 0\), it is called the affine scaling step. It has no centering, and tries to solve the original problem in one step.

Using a neighborhood with \(\beta = 0.5\), the resulting algorithm (the Mizuno-Todd-Ye algorithm) converges quadratically to an optimal solution, keeping the complexity at its best value of \(O(\sqrt{nL})\) iterations.
Predictor-corrector methods

Alternate two kinds of iterations:

- **Predictor**: An iteration starts with \((x, s)\) near the central path, and computes a Newton step \((u, v)\) with goal \(\gamma \mu(x, s)\), \(\gamma\) small.
- Follow it until

  \[ \delta(x + \lambda u, s + \lambda v, \mu(x + \lambda u, s + \lambda v)) = \beta \]

- **Corrector**: Set \(x^+ = x + \lambda u\), \(s^+ = s + \lambda v\), compute \(\mu = \mu(x^+, s^+)\) and take a Newton step with target \(\mu\)

- If the predictor uses \(\gamma = 0\), it is called the **affine scaling step**. It has no centering, and tries to solve the original problem in one step.

- Using a neighborhood with \(\beta = 0.5\), the resulting algorithm (the Mizuno-Todd-Ye algorithm) converges quadratically to an optimal solution, keeping the complexity at its best value of \(O(\sqrt{nL})\) iterations.
Predictor-corrector
Mehrotra Predictor-corrector method: second order

When computing the Newton step, we eliminated the nonlinear term $uv$ in the equation

\begin{align*}
xs + su + xv + uv &= \mu e \\
Qu + Rv &= 0
\end{align*}

The second order method corrects the values of $u, v$ by estimating the value of the term $uv$ by a predictor step.

- **Predictor**: An iteration starts with $(x, s)$ near the central path, and computes a Newton step $(u, v)$ with goal $\mu^+$, small. The first equation is

\[ xv + su = -xs + \mu^+ e \]

- **Compute a correction** $(\Delta u, \Delta v)$ by

\[ x\Delta v + s\Delta u = -uv. \]

- **Line search**: Set $x^+ = x + \lambda u + \lambda^2 \Delta u$, $s^+ = s + \lambda v + \lambda^2 \Delta v$, by a line search so that $\delta(x^+, s^+, \mu(x^+, s^+)) = \beta$. 
Mehrotra Predictor-corrector
Thank you