In path-following methods for convex quadratic programming, one must solve systems of equations of the form:

\[
\begin{align*}
Ax - y &= b \\
-Qx + A'w + s &= c \\
XSe &= \mu e \\
WYe &= \mu e
\end{align*}
\]

This system consists of both linear and nonlinear equations, and are frequently solved using Newton's method.

**The Arithmetic-Geometric Mean Inequality**

**Simplest case:** Given two positive numbers \( a \) and \( b \), their arithmetic mean \( \frac{a+b}{2} \) is greater than or equal to their geometric mean \( \sqrt{ab} \).

I.e., \( \frac{a+b}{2} \geq \sqrt{ab} \)

with equality if and only if \( a = b \).

**Proof:** Let \( \alpha \) and \( \beta \) be real numbers such that \( a = \alpha^2 \geq 0 \) and \( b = \beta^2 \geq 0 \). Then:

\[
(a+b)^2 = \alpha^2 - 2\alpha\beta + \beta^2 \geq 0
\]

\[
\alpha^2 + \beta^2 \geq 2\alpha\beta
\]

\[
\frac{a+b}{2} = \sqrt{\alpha^2 + \beta^2} \geq \sqrt{2\alpha\beta} = \sqrt{ab}
\]

with equality if and only if \( a = b \).

**The General Case:** Let \( x_1, x_2, \ldots, x_n \geq 0 \) and \( \delta_1, \delta_2, \ldots, \delta_n \geq 0 \) and \( \sum_{i=1}^{n} \delta_i = 1 \). Then:

\[
\sum_{i=1}^{n} \delta_i x_i = \prod_{i=1}^{n} x_i
\]

with equality if and only if \( x_1 - x_2 - \ldots - x_n = 0 \).
The Arithmetic-Geometric Mean Inequality

If we let \( n=2 \), and \( \delta_1 = \frac{a}{b} \), then we obtain the earlier inequality,

\[
\frac{a}{2} + \frac{b}{2} \geq \sqrt{ab}
\]

The Arithmetic-Geometric Mean Inequality

Writing \( u_i = \delta x_i \), we get:

Equivalent form:

\[
\sum \frac{u_i}{\delta_i} \geq \prod \frac{u_i}{\delta_i}
\]

where \( \delta_1 \cdot \delta_2 \cdots \delta_n \geq 0 \) and \( \sum \delta_i = 1 \)

with equality if \& only if \( u_1/\delta_1 = u_2/\delta_2 \cdots = u_n/\delta_n \)

\( \square \)

Condensation of Posynomials

\[
g(x_1, x_2, \ldots, x_m) = \sum_{i=1}^{n} c_i \prod_{j=1}^{m} x_j^{a_{ij}}
\]

where \( c_i > 0 \) and \( a_{ij} \) are real numbers.

\( \square \)

That is, we obtain a monomial approximation (lower bound) of the posynomial.

\[
g(x) = \sum_{i} c_i \prod_{j} x_j^{\alpha_j} \geq C(\delta) \prod_{j} x_j^{\alpha_{j}(\delta)}
\]

where \( C(\delta) = \prod_{i} \frac{c_i^{\delta_i}}{\delta_i}, \alpha_{j}(\delta) = \sum_{i} a_{ij}\delta_i \)

which is exact when

\[
\frac{c_1 \prod_{j} x_j^{a_{1j}}}{\delta_1} = \frac{c_2 \prod_{j} x_j^{a_{2j}}}{\delta_2} = \cdots = \frac{c_n \prod_{j} x_j^{a_{nj}}}{\delta_n}
\]

Signomial Functions

\[
g(x_1, x_2, \ldots, x_m) = \sum_{i=1}^{n} c_i \prod_{j=1}^{m} x_j^{a_{ij}}
\]

Condensation has long been used in solving Signomial GP problems (which are essentially nonconvex) by means of a sequence of approximating Posynomial GP problems (which are essentially convex problems).

\( \square \)

Example

Minimize \( x_1 \)
subject to
\[
\delta_1 (x_1 - 2)^2 + (x_2 - 4)^2 \geq 4 \quad \text{\( X \) is outside a circle centered at \((2,4)\) with radius 2}
\]
\[
\delta_2 (x_1 - 3)^2 + (x_2 - 3)^2 \leq 4 \quad \text{\( X \) is within a circle centered at \((3,3)\) with radius 2}
\]

Minimize \( x_1 \)
subject to
\[
(x_1 - 2)^2 + (x_2 - 4)^2 \geq 4 \quad \text{\( X \) is outside a circle centered at \((2,4)\) with radius 2}
\]
\[
(x_1 - 3)^2 + (x_2 - 3)^2 \leq 4 \quad \text{\( X \) is within a circle centered at \((3,3)\) with radius 2}
\]

Feasible region
Reformulation as a GP problem

\[
\begin{align*}
(X_1 - 3)^2 + (X_2 - 3)^2 & \leq 4 \\
\Rightarrow \quad [x_1^2 - 6x_1 + 9] + [x_2^2 - 6x_2 + 9] & \leq 4 \\
\Rightarrow \quad x_1^2 - 6x_1 + x_2^2 + 14 & \leq 6x_2 \\
\text{The constraint becomes the signomial constraint} \\
\Rightarrow \quad \frac{x_1^2}{6} + \frac{x_2^2}{6} + \frac{7x_1^2}{3} - x_1x_2 & \leq 1
\end{align*}
\]

Signomial Geometric Program

\[
\begin{align*}
\text{Minimize} \quad & X_1 \\
\text{subject to} \\
\frac{x_1}{4} + \frac{x_2}{2} - \frac{x_1^2}{16} - \frac{x_2^2}{16} & \leq 1 \\
\frac{x_1^2x_2^2}{6} + \frac{x_1^2}{6} + \frac{7x_2^2}{3} - x_1x_2 & \leq 1 \\
x_1 > 0, \quad x_2 > 0
\end{align*}
\]

To condense the signomial constraint:
\[
\frac{x_1}{4} + \frac{x_2}{2} \leq 1 + \frac{x_1^2}{16} + \frac{x_2^2}{16}
\]

We next condense the denominator of
\[
\frac{0.25x_1 + 0.5x_2}{1 + 0.0625x_1^2 + 0.0625x_2^2} \leq 1
\]

into a single term. Let's use the point \(x_0 = (4, 5)\) at which the terms of the denominator are
\[
1 + 1 + 1.5625 = 3.5625
\]

Then
\[
\delta_1 = \delta_2 = \frac{1}{3.5625} = 0.2807 \quad \text{and} \quad \delta_3 = \frac{1.5625}{3.5625} = 0.4386
\]

\[
\begin{align*}
\delta_1 = \delta_2 = 0.2807, \quad & \delta_3 = 0.4386 \\
\text{Coefficient:} \\
C(\delta) &= \frac{1}{\delta_1 \delta_2} \\
C(\delta) &= \begin{bmatrix} 0.2807 & 0.2807 \\ 0.0625 & 0.0625 \\ 0.4386 & 0.4386 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
C(\delta) &= \frac{1}{0.3967} \\
= 0.2527
\end{align*}
\]

\[
\begin{align*}
\delta_1 = \delta_2 = 0.2807, \quad & \delta_3 = 0.4386 \\
\text{Exponents:} \\
a_1(\delta) &= \sum_{i=1}^{3} a_i \delta_i \\
a_1 &= a_1(0.2807 + 0.2807 + 0.0625) = 0.5614 \\
a_2 &= a_2(0.2807 + 0.2807 + 0.4386) = 0.8772
\end{align*}
\]
Condensed denominator is

\[ C(\delta) = 0.3987 \]
\[ a_1 = 0.5614 \]
\[ a_2 = 0.8772 \]

**monomial**

\[ \frac{0.25X_1 + 0.5X_2}{0.3987 X_1^{0.5614} X_2^{0.8772}} = \frac{0.25}{0.3987} X_1^{1 - 0.5614} X_2^{-0.8772} + \frac{0.5}{0.3987} X_1^{0.5614} X_2^{0.8772} \]
\[ = 0.627 X_1^{0.406} X_2^{-0.8772} + 1.254 X_1^{-0.5614} X_2^{0.128} \]

which is a posynomial!

**posynomial**

\[ \frac{0.25X_1 + 0.5X_2}{0.3987 X_1^{0.5614} X_2^{0.8772}} \]

**posynomial**

\[ \frac{0.25X_1 + 0.5X_2}{0.3987 X_1^{0.5614} X_2^{0.8772}} \]

Geometric Inequality implies

\[ 1 + 0.0625X_1^2 + 0.0625X_2^2 \leq \frac{0.3987 X_1^{0.5614} X_2^{0.8772}}{0.25X_1 + 0.5X_2} \]

and so

\[ \frac{0.25X_1 + 0.5X_2}{1 + 0.0625X_1^2 + 0.0625X_2^2} \leq \frac{0.3987 X_1^{0.5614} X_2^{0.8772}}{0.25X_1 + 0.5X_2} \]

If we constrain this posynomial so as to be \( \leq 1 \), then by the geometric inequality, the original signomial should also be \( \leq 1 \).

That is, any \( X \) feasible in the posynomial constraint derived by condensation will also be feasible in the signomial constraint:

\[ \frac{0.25X_1 + 0.5X_2}{1 + 0.0625X_1^2 + 0.0625X_2^2} \leq 1 \]

The second signomial constraint may be condensed in a similar fashion:

\[ \frac{X_1 X_2^{-1}}{6} + \frac{X_2}{6} + \frac{7X_2^{-1}}{3} - X_1X_2^{-1} \leq 1 \]

\[ \frac{X_1 X_2^{-1}}{6} + \frac{X_2}{6} + \frac{7X_2^{-1}}{3} \leq 1 + X_1X_2^{-1} \]

\[ \frac{X_1 X_2^{-1}}{6} + \frac{X_2}{6} + \frac{7X_2^{-1}}{3} \leq 1 \]

The signomial GP problem is therefore approximated by the posynomial problem:

Minimize \( X_1 \)

subject to

\[ 0.627 X_1^{0.406} X_2^{-0.8772} + 1.254 X_1^{-0.5614} X_2^{0.128} \leq 1 \]

\[ 0.08385X_1^{1.555} X_2^{-0.555} + 0.08385X_1^{1.555} X_2^{-0.555} + 1.174X_1^{-0.444} X_2^{0.555} \leq 1 \]

\[ X_1 > 0, X_2 > 0 \]

**Monomial Method**

We wish to find a (positive) solution of the following system of nonlinear (signomial) equations:

\[ g_k(x) = \sum_j \sigma_{jk} \prod_i x_j^{\delta_{ijk}} = 0, k=1, \ldots, N \]

where \( \sigma_{jk} \in \{+1, -1\} \), \( \delta_{ijk} > 0 \)

**Example:**

\[ \begin{align*}
2.5 x_1^{1.5} + 15 x_1^{0.5} x_2^{0.5} - 30x_2 &= 0 \\
77 + 9x_2^{1.5} - 28x_2 - 4x_2^{0.5} &= 0
\end{align*} \]
Define the index sets of the positive & negative terms of each equation:

\[ T^+_k = \{ i \mid \sigma_{ik} > 0 \} \quad \text{&} \quad T^-_k = \{ i \mid \sigma_{ik} < 0 \} \]

Then separate each signomial into positive & negative parts:

\[ g_k(x) = P_k(x) - Q_k(x) \]

where

\[ P_k(x) = \sum_{i \in T^+_k} c_{ik} \prod_j x_j^\sigma_{ij} \quad \text{&} \quad Q_k(x) = \sum_{i \in T^-_k} c_{ik} \prod_j x_j^\sigma_{ij} \]

\[ g_k(x) = P_k(x) - Q_k(x) = 0 \]

\[ \Rightarrow \quad P_k(x) = Q_k(x) \]

\[ \Rightarrow \quad P_k(x) = 1 \quad \text{&} \quad Q_k(x) \]

Each of the posynomials \( P_k(x) \) and \( Q_k(x) \) are then condensed into monomial approximations \( \tilde{P}_k(x) \) and \( \tilde{Q}_k(x) \), respectively, and the ratio of the two monomials is also a monomial!

Each nonlinear equation is then approximated by a monomial equation

\[ \frac{P_k(x)}{Q_k(x)} = \frac{P_k(\delta)}{Q_k(\delta)} = \frac{C_k(\delta)}{} \prod_j x_j^{\sigma_{ij}(\delta)} = 1 \]

for some choice of the weights (\( \delta \))

By taking the logarithms of both sides and making the change of variable \( z_j = \ln x_j \) we get the linear equation

\[ \sum_j \alpha_{jk}(\delta) z_j = - C_k(\delta) \]

It can be shown that the "Monomial" Method is equivalent to Newton's Method applied to

\[ \ln \left( \frac{P_k(x)}{Q_k(x)} \right) = 0, \quad k = 1, \ldots, N \]

Select an initial starting point \( x^0 \).

1. Evaluate the weights of all the terms:

\[ \alpha_{ik}(\delta) = \frac{\prod_j (x_j^\sigma)^{\delta_{ij}}}{P_k(x^0)} \quad \forall i \in T^+_k \quad \text{&} \quad \delta_{ik} = \frac{\prod_j (x_j^\sigma)^{-\delta_{ij}}}{Q_k(x^0)} \quad \forall i \in T^-_k \]

2. Evaluate \( C_k(\delta) \) and \( \alpha_{ik}(\delta) \)

3. Solve the linear system of equations in \( z \).

4. Exponentiate \( z \) to obtain \( x' \) (yielding \( x' > 0 \))

5. Test for convergence, e.g.,

\[ \| x^0 - x' \| \leq \varepsilon \]

If the test fails, replace \( x^0 \) with \( x' \) and return to step 1.

---

**Example**

A "toy" LCP:

\[ y = Mx + q, \quad xy = 0 \]

\[ \begin{cases} 
xy = \mu \\
y = x + 1 
\end{cases} \]

i.e., one "complementarity" equation

one linear equation

---

---

\[ \begin{cases} 
xy = \mu = 0.75 \\
y = x + 1 
\end{cases} \]

In general, the solution is

\[ x(\mu) = \frac{1}{2} + \sqrt{\frac{1}{4} + \mu} \]

\[ y(\mu) = \frac{1}{2} + \sqrt{\frac{1}{4} + \mu} \]
Monomial Method

Note that the "complementarity" equation is already in monomial form.

The linear equation is approximated by a monomial as follows:
\[
P - Q = (x+1) - y = 0
\Rightarrow P = \frac{x+1}{y}
\]

\[
x + 1 \approx \left( \frac{x}{x} \right) \frac{1}{y} = \delta x \frac{1}{x+1} = 1
\]

where the weights are:
\[
\delta_x = \frac{1}{x+1}, \quad \delta_y = \frac{1}{x+1}
\]

In the Monomial Method, then, we solve

\[
\begin{bmatrix}
1 & -1 \\
x^y & y^x + 1
\end{bmatrix}
\begin{bmatrix}
z_x \\
z_y
\end{bmatrix} = \begin{bmatrix}
\ln \mu \\
\mu
\end{bmatrix}
\]

and update \(x^y \leftarrow \exp(z_x)\) & \(y^x \leftarrow \exp(z_y)\)

while in Newton's Method, we solve

\[
\begin{bmatrix}
y^x & x^y \\
1 & -1
\end{bmatrix}
\begin{bmatrix}
\Delta_x \\
\Delta_y
\end{bmatrix} = \begin{bmatrix}
-\mu \cdot x^y \\
y^x \cdot x^y
\end{bmatrix}
\]

and update \(x^y \leftarrow x^y + \Delta_x\) & \(y^x \leftarrow y^x + \Delta_y\)

Still another algorithm may be obtained by applying Newton's Method after making the logarithmic transformation:

\[
\begin{bmatrix}
z_x + z_y = \ln \mu \\
\mu \cdot x^y - y^x = -1
\end{bmatrix}
\]

which requires solving

\[
\begin{bmatrix}
1 & 1 \\
\ln x^y - y^x
\end{bmatrix}
\begin{bmatrix}
dz_x \\
dz_y
\end{bmatrix} = \begin{bmatrix}
-\mu \cdot x^y \\
y^x \cdot x^y
\end{bmatrix}
\]

and updating \(z_x^\prime \leftarrow z_x + dz_x\) & \(z_y^\prime \leftarrow z_y + dz_y\)

Newton's Method

\[
\mu = 10^{-8}
\]

Stopping criterion: \(|\mu \cdot x^y| + |y^x - 1| \leq 10^{-8}\)

Starting point: \((100, 100)\)

An Infeasible Path-Following Algorithm using the Newton-Central Method

Equations to be approximately solved at each iteration

\[
\begin{cases}
Ax - y = b \\
-Qx + A^T w + s = c \\
X^e s = \mu e
\end{cases}
\]

The logarithmic transformation is made, so that the complementarity equations are linearized, and the linear equations become nonlinear:

\[
P(e^\mu) - Q(e^\mu) = 0
\]

Monomial Method

\[
\mu = 10^{-8}
\]

Stopping criterion: \(|\mu \cdot x^y| + |y^x - 1| \leq 10^{-8}\)

Starting point: \((100, 100)\)

An Infeasible Path-Following Algorithm using the Monomial Method

Equations to be approximately solved at each iteration

\[
\begin{cases}
Ax - y = b \\
-Qx + A^T w + s = c \\
W^\mu e = \mu e
\end{cases}
\]

The linear equations are approximated by monomial equations, and the logarithmic transformation is then made to linearize all the constraints.
0. Start with any interior solution \((x_b, y_b, x_c, w) > 0\) set \(k = 0\), and choose 3 tolerances \(\epsilon_1, \epsilon_2, \epsilon_3 > 0\). 

1. Compute \(\mu^k = \sigma_{\frac{\kappa^2 + y^k w^k}{m+n}}\) for \(0 < \sigma < 1\). 
   \[ x_{b}^k = b + y^k A x^k, \quad y_{c}^k = Q x^k + c - A^T w^k - s^k \]

2. If \(\mu^k \leq \epsilon_1, \frac{|x_{b}^k|}{b + 1} \leq \epsilon_2, \frac{|y_{c}^k|}{Q x^k + c + 1} \leq \epsilon_3\) then stop & accept the current iterate as optimal.

3. Evaluate the weights

4. Compute coefficients & rhs of linear system

5. Solve linear system & return to step 1.

Properties of the sequence generated by this algorithm:
- exactly on the central trajectory
- strictly positive
- converges if bounded and the algorithm does not fail

Computational Experience

<table>
<thead>
<tr>
<th>Number of Subproblems</th>
<th>Separable Problems</th>
<th>Nonseparable Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>O</td>
</tr>
<tr>
<td>M=2</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>M=4</td>
<td>25%</td>
<td>12.5%</td>
</tr>
<tr>
<td>M=8</td>
<td>12.5%</td>
<td>6.25%</td>
</tr>
<tr>
<td>M=12</td>
<td>8.33%</td>
<td>4.17%</td>
</tr>
</tbody>
</table>

\# variables \# constraints 2M 3M

Adjustment of factor \(\sigma^k\)

Standard Newton Algorithm:
\[ \sigma^{k+1} = \begin{cases} 
\text{min} \left( 0.95, 1.3 \sigma^k \right) & \text{if } \frac{\mu^k}{\sigma^k} < 1 \\
\text{max} \left( 0.2, 0.7 \sigma^k \right) & \text{otherwise}
\end{cases} \]

Newton-Central & Monomial Algorithms:
\[ \sigma^{k+1} = \begin{cases} 
\text{min} \left( 0.95, 1.3 \sigma^k \right) & \text{if } \text{error}^k < 1 \\
\text{max} \left( 0.2, 0.7 \sigma^k \right) & \text{otherwise}
\end{cases} \]
\[ \text{error}^k = \frac{|x_{b}^k|}{b + 1} + \frac{|y_{c}^k|}{Q x^k + c + 1} \]

Iterations vs \# subproblems

CPU vs \# subproblems