Constrained Geometric Programming

This Hypercard stack was prepared by:
Dennis L. Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
e-mail: dbricker@icaen.uiowa.edu

author

In GEOMETRIC PROGRAMMING, recall that the objective function, to be *minimized*, is a *posynomial*:

$$\boxed{g_0(x_1, x_2, \cdots x_m) = \sum_{i=1}^{T} c_i \prod_{j=1}^{m} x_j^{a_{ij}}}$$

where $c_i > 0$ *and* a_{ij} *are real numbers.*

In constrained Geometric Programming, the constraints must be of the form

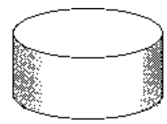
$$\boxed{\text{posynomial} \le 1}$$

This is not so restrictive an assumption as it might first appear, since many constraints may be transformed into this form:

$$x_1 x_2 \ge 25 \quad \Leftrightarrow \quad 25 x_1^{-1} x_2^{-1} \le 1$$
$$x_1 + x_2 \le 5x_3 \quad \Leftrightarrow \quad 0.2\, x_1 x_3^{-1} + 0.2\, x_2 x_3^{-1} \le 1$$
$$x_1 - x_2 \ge 10 \quad \Leftrightarrow \quad 10 x_1^{-1} + x_1^{-1} x_2 \le 1$$

...etc. (since x>0)

EXAMPLE



We wish to design an oil storage tank having volume 1000π m³, and minimum cost, where cost of materials (top, sides, & bottom) is $1/m².

Let the design variables be h=height (m), R=radius(m)

Cost function $\underbrace{2\pi R^2}_{\text{top+bottom}} + \underbrace{2\pi R h}_{\text{sides}}$

The cost function is a posynomial in R and h.

Constraint $\pi R^2 h = 1000\pi \text{ m}^3$

The constraint is an equation, not an inequality, but it seems clear that if "=" is replaced with "≥", the inequality would be tight at the optimum!

$$\pi R^2 h \ge 1000\pi \text{ m}^3 \quad \Leftrightarrow \quad 1000\, h^{-1} R^{-2} \le 1$$

To simplify the notation (avoiding the need for triple subscripts), we will assign each term a unique index from 1 (first term in the objective posynomial) to n (total # of terms in all posynomials), and define the index sets $[k]$

$$\bigcup_{k=0}^{p} [k] = \{1,2,\cdots n\}, \quad [k_1] \cap [k_2] = \varnothing \text{ if } k_1 \ne k_2$$

Constraints

$$\boxed{g_k(x_1, x_2, \cdots x_m) = \sum_{i \in [k]} c_i \prod_{j=1}^{m} x_j^{a_{ij}} \le 1}$$

k=1,2,... p

term #1 term #2

Minimize $2\pi R^2 + 2\pi R h$
subject to
$1000\, h^{-1} R^{-2} \le 1$
$h > 0 \;\&\; R > 0$

term #3

Index sets:
[0] = {1,2}
[1] = {3}

Geometric Programming Primal Problem

$$\begin{cases} \text{Minimize } g_0(x) \\ \text{subject to } g_k(x) \le 1, \; k=1,\cdots p \\ x_j > 0, \; j=1,2,\cdots m \end{cases}$$

where $g_k(x) = \sum_{i \in [k]} c_i \prod_{j=1}^{m} x_j^{a_{ij}}$ *posynomial*

$c_i > 0$

$$\bigcup_k [k] = \{1,2,\cdots n\} \;\&\; [k'] \cap [k''] = \varnothing \text{ for } k' \ne k''$$

Geometric Programming Dual Problem

The GP dual problem is linearly-constrained, and (if the negative of the log of the objective is minimized) has a convex objective function.

To each posynomial term #i, there corresponds a "weight" δ_i, and to each posynomial there corresponds a Lagrange multiplier λ_k.

DGP: Maximize $v(\delta,\lambda) = \prod_{k=0}^{p} \left\{ \lambda_k^{\lambda_k} \prod_{i\in[k]} \left(\frac{c_i}{\delta_i}\right)^{\delta_i} \right\}$

subject to

Geometric Programming Dual Problem

$\sum_{i\in[k]} \delta_i = \lambda_k$, $k=0,1,\cdots p$

$\sum_{i=1}^{n} a_{ij}\,\delta_i = 0$, $j=1,\cdots m$ *orthogonality constraints*

$\lambda_0 = 1$ *normality constraint*

$\delta_i \geq 0,\ \lambda_k \geq 0\ \ \forall i,k$

Note: $\bigcup_k [k] = \{1,2,\cdots N\}$ & $[k'] \cap [k''] = \varnothing$ for $k' \neq k''$

term #1 term #2 dual constraints

Minimize $2\pi R^2 + 2\pi R h$ → $\delta_1 + \delta_2 = \lambda_0 = 1$ "normality"

subject to

term #3 $1000\, h^{-1} R^{-2} \leq 1$ → $\delta_3 = \lambda_1$

$h > 0$ & $R > 0$

Exponents of R are: 2, 1, −2 and so the orthogonality constraint is: $2\delta_1 + \delta_2 - 2\delta_3 = 0$

Exponents of h are: 0, 1, −1 and so the orthogonality constraint is: $\delta_2 - \delta_3 = 0$

Maximize $\left(\frac{2\pi}{\delta_1}\right)^{\delta_1} \left(\frac{2\pi}{\delta_2}\right)^{\delta_2} \left(\frac{1000}{\delta_3}\right)^{\delta_3} \lambda_1^{\lambda_1}$

subject to $\delta_1 + \delta_2 = \lambda_0 = 1$ ← normality constraint

$\delta_3 = \lambda_1$

$2\delta_1 + \delta_2 - 2\delta_3 = 0$

$\delta_2 - \delta_3 = 0$ ← orthogonality constraints

$\delta_i \geq 0, i=1,2,3; \lambda_1 \geq 0$

$$\begin{aligned} \delta_1 + \delta_2 = \lambda_0 &= 1 \\ \delta_3 &= \lambda_1 \\ 2\delta_1 + \delta_2 - 2\delta_3 &= 0 \\ \delta_2 - \delta_3 &= 0 \end{aligned}$$

There are an equal number of variables and equations, which (assuming full rank) implies that there is a unique feasible solution:

$\delta_1 = 1/3,\ \delta_2 = 2/3,\ \delta_3 = \lambda_1 = 2/3$

Duality Theorem

If x is primal-feasible, and (δ,λ) is dual-feasible, then

$g_o(x) \geq v(\delta,\lambda)$

with equality if & only if

$$\boxed{\delta_i\, g_k(x) = \lambda_k c_i \prod_{j=1}^{m} x_j^{a_{ij}}} \quad \forall k=0,1,\ldots p \quad \forall i\in[k]$$

$$\lambda_k^* > 0\ \&\ \delta_i^* > 0 \Rightarrow \frac{c_i \prod_{j=1}^{m} x_j^{*\,a_{ij}}}{g_k(x^*)} = \frac{\delta_i^*}{\lambda_k^*}$$

If $k=0$ (objective function), $\lambda_k = 1$ and

$$\boxed{\delta_i^* = \frac{c_i \prod_{j=1}^{m} x_j^{*\,a_{ij}}}{g_o(x^*)}}$$

i.e., δ_i is the fraction of the minimum cost which is contributed by term # i

(same relationship as in unconstrained GP!)

If $k \geq 1$ and $g_k(x^*) < 1$ (slack constraint), then

$\lambda_k = 0$ & $\delta_i = 0\ \forall i \in [k]$

and no information about the primal solution is available from these dual variables

If $k \geq 1$ and $\lambda_k > 0$ then $g_k(x^*) = 1$ (tight constraint) and

$$\boxed{c_i \prod_{j=1}^{m} x_j^{*\,a_{ij}} = \frac{\delta_i^*}{\lambda_k^*}}$$

RECOVERY of PRIMAL VARIABLES from DUAL SOLUTION

Given optimal dual solution δ^* & λ^*,

$$\boxed{\delta_i^* g_k(x) = \lambda_k^* c_i \prod_{j=1}^{m} x_j^{a_{ij}}} \quad \forall\, k=0,1,\ldots p \quad \forall\, i \in [k]$$

$$\Rightarrow \prod_{j=1}^{m} x_j^{a_{ij}} = \frac{\delta_i^*}{\lambda_k^*} g_k(x) \Rightarrow \sum_{j=1}^{m} a_{ij} \ln x_j = \ln\left[\frac{\delta_i^*}{\lambda_k^*} g_k(x)\right] \ \forall\, i \in [k]$$

But we can determine $g_k(x)$ *since* $g_o(x) = v(\delta^*, \lambda^*)$ *and* $\lambda_k^* > 0 \Rightarrow g_k(x) = 1$ *for k=1,2,...p*

RECOVERY of PRIMAL VARIABLES from DUAL SOLUTION

Given optimal dual solution δ^* & λ^*, we evaluate the dual objective $v(\delta^*,\lambda^*)$, and solve

$$\boxed{\sum_{i=1}^{m} a_{ij} z_j = \ln\left[\delta_i^* v(\delta^*,\lambda^*)\right] \ \forall\, i \in [0]}$$

and if $\lambda_k^* \neq 0$, $k=1,2,\ldots p$

$$\boxed{\sum_{i=1}^{m} a_{ij} z_j = \ln\left[\frac{\delta_i^*}{\lambda_k^*}\right] \ \forall\, i \in [k]}$$

where $\boxed{z_j \equiv \ln x_j}$

a linear system of equations!

Storage Tank Example

$$\delta_1^* = 1/3,\ \delta_2^* = 2/3,\ \delta_3^* = \lambda_1^* = 2/3$$

$$v(\delta^*,\lambda^*) = \left(\frac{2\pi}{1/3}\right)^{1/3} \left(\frac{2\pi}{2/3}\right)^{2/3} \left(\frac{1000}{2/3}\right)^{2/3} \left(2/3\right)^{2/3} = 1187.45$$

This must also be the optimal value of $g_o(x^*)$

We can now use this information to compute the optimal dimensions!

optimal cost of oil storage tank

$$\boxed{\delta_i^* = \frac{c_i \prod_{j=1}^{m} x_j^{*\,a_{ij}}}{g_o(x^*)}}$$

Computing optimal dimensions of tank

$$\frac{2\pi R^2}{1187.45} = \frac{1}{3} \Rightarrow R^* = \sqrt{\frac{\frac{1}{3}\times 1187.45}{2\pi}} = \sqrt{62.99} = 7.937 \text{ m.}$$

$$\frac{2\pi R h}{1187.45} = \frac{2}{3} \Rightarrow \quad h^* = \frac{\frac{2}{3}\times 1187.45}{2\pi R} = 15.874 \text{ m.}$$

Note that $h^* = 2R^*$

Two equations were sufficient to compute the two primal variables, but a third equation is available as a check:

$$\boxed{c_i \prod_{j=1}^{m} x_j^{*\,a_{ij}} = \frac{\delta_i^*}{\lambda_k^*}} \qquad 1000 h^{-1} R^{-2} = \frac{2/3}{2/3} = 1$$

$$1000\left(\frac{1}{15.874}\right) \times \left(\frac{1}{7.937}\right)^2 = 1$$

Note that in this case (zero degree of difficulty), the optimal weights are determined only by the exponents of
the primal variables in the posynomials,
NOT by the coefficients.

$$\delta_1^* = 1/3,\ \delta_2^* = 2/3$$

Thus, independent of the cost coefficients, the first objective term (cost of top & bottom) will always contribute one-third of the total optimal cost, and the second term (cost of the sides) will contribute two-thirds!

Even if the cost of materials for the top & bottom were to cost not \$1 per square meter, but \$1000 per square meter, while the cost of materials for the sides remains \$1 per square meter, the cost of top & bottom will still optimally contribute one-third of the total cost!

(The total cost and the dimensions will change, of course.)

ILLUSTRATION

Let c = \$ per square meter for material for top & bottom, while \$1 = material cost for side.

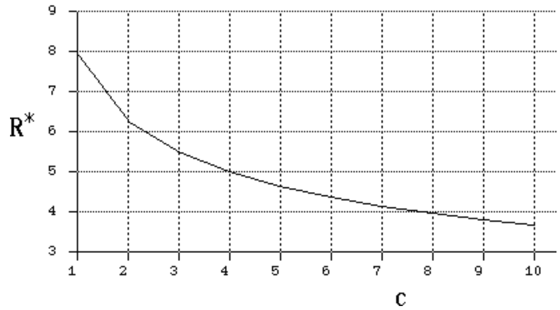
$$g_0(R,h) = 2\pi c R^2 + 2\pi R h$$

$$R^* = \sqrt{\frac{g_0(R^*,h^*)}{6\pi c}}$$

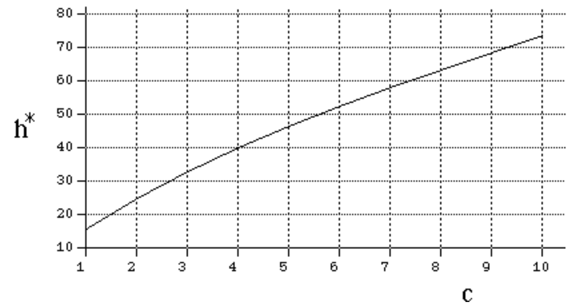
$$h^* = \frac{g_0(R^*,h^*)}{3\pi R^*}$$

computed using dual objective function

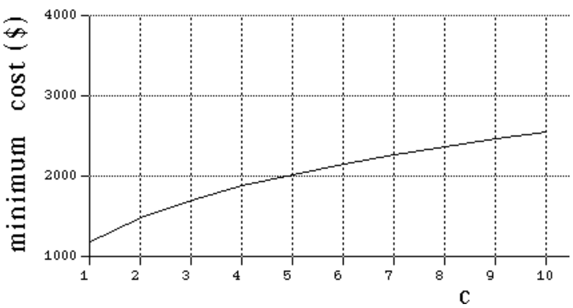
As cost of top & bottom increases, radius decreases...



As cost of top & bottom increases, height increases...



As cost of top & bottom increases, total cost increases....



As cost of top & bottom increases, the ratio of the 2 cost components is unchanged!

c	dimensions (m.)		Cost ($)		
	R	h	top&bottom	sides	Total
1	7.93700526	15.87401052	395.81	791.63	1187.44
2	6.299605249	25.198421	498.69	997.39	1496.09
3	5.503212081	33.01927249	570.86	1141.73	1712.59
4	5	40	628.31	1256.63	1884.95
5	4.641588834	46.41588834	676.83	1353.67	2030.50
6	4.367902324	52.41482788	719.24	1438.49	2157.73
7	4.149132667	58.08785734	757.16	1514.33	2271.50
8	3.96850263	63.49604208	791.63	1583.26	2374.89
9	3.815714142	68.68285455	823.33	1646.66	2469.99
10	3.684031499	73.68062997	852.75	1705.51	2558.27

$$v(\delta,\lambda) = \prod_{k=0}^{p} \left\{ \lambda_k^{\lambda_k} \prod_{i\in[k]} \left(\frac{c_i}{\delta_i}\right)^{\delta_i} \right\}$$

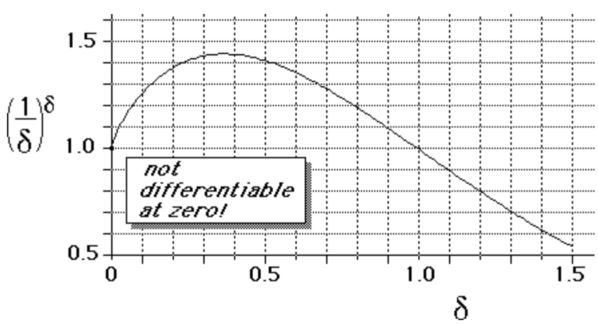
dual objective

Computation of the factor $\left(\frac{c}{\delta}\right)^{\delta}$ at $\delta = 0$ appears problematic...

But $$\lim_{\delta \to 0} \left(\frac{c}{\delta}\right)^{\delta} = 1$$

we therefore "define" $\left(\frac{c}{0}\right)^{0} \equiv 1$

δ	$\frac{1}{\delta}$	$\left(\frac{1}{\delta}\right)^{\delta}$
10	1.00E⁻1	0.000000000100000
1	1.00E0	1.000000000000000
0.1	1.00E1	1.258925411794167
0.01	1.00E2	1.047128548050900
0.001	1.00E3	1.006931668851804
0.0001	1.00E4	1.000921458319296
0.00001	1.00E5	1.000115135882277
0.000001	1.00E6	1.000013815605993
0.0000001	1.00E7	1.000001611810864
0.00000001	1.00E8	1.000000184206825
0.000000001	1.00E9	1.000000020723266
0.0000000001	1.00E10	1.000000002302585
0.00000000001	1.00E11	1.000000000253284
0.000000000001	1.00E12	1.000000000027631
0.0000000000001	1.00E13	1.000000000002993
0.00000000000001	1.00E14	1.000000000000322
0.000000000000001	1.00E15	1.000000000000035



$$\boxed{\text{Maximize} \quad v(\delta,\lambda) = \prod_{k=0}^{p} \left\{ \lambda_k^{\lambda_k} \prod_{i\in[k]} \left(\frac{c_i}{\delta_i}\right)^{\delta_i} \right\}}$$

is equivalent to

$$\boxed{\text{Max} \quad \ln v(\delta,\lambda) = \sum_{i=1}^{n} \left\{ \delta_i \ln c_i - \delta_i \ln \delta_i \right\} + \sum_{k=0}^{K} \lambda_k \ln \lambda_k}$$

since the logarithm function is monotonically increasing.
This objective has the advantage that it is separable, i.e., each term contains only 1 variable!

$$\text{Max} \quad \ln v(\delta,\lambda) = \sum_{i=1}^{n} \left\{\delta_i \ln c_i - \delta_i \ln \delta_i\right\} + \sum_{k=0}^{p} \lambda_k \ln \lambda_k$$

The above objective is *concave* if we make the substitution $\boxed{\sum_{i\in[k]} \delta_i = \lambda_k}$

$$\text{Max} \ \ln V(\delta) = \sum_{i=1}^{n}\left\{\delta_i \ln c_i - \delta_i \ln \delta_i\right\} + \sum_{k=0}^{p}\left[\sum_{i\in[k]} \delta_i\right] \ln \left[\sum_{i\in[k]} \delta_i\right]$$

(but no longer separable!)

$$\text{DGP}': \text{Max} \sum_{i=1}^{n}\left\{\delta_i \ln c_i - \delta_i \ln \delta_i\right\} + \sum_{k=0}^{p}\left[\sum_{i\in[k]} \delta_i\right] \ln \left[\sum_{i\in[k]} \delta_i\right]$$

Geometric Programming Dual Problem

subject to

$$\sum_{i\in[k]}^{n} \delta_i = 1 \qquad \textit{normality}$$

$$\sum_{i=1} a_{ij}\, \delta_i = 0, \ \ j=1,\cdots m \qquad \textit{orthogonality}$$

$$\delta_i \geq 0, \quad \forall i$$

DGP' has several noteworthy properties:

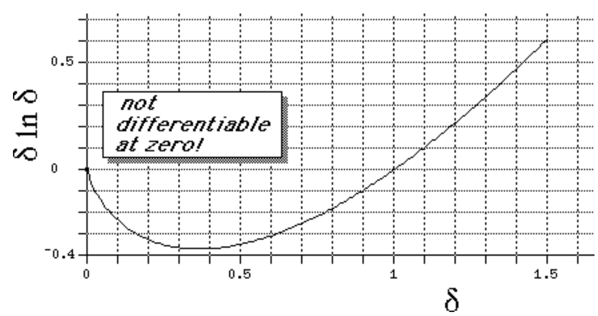
- objective is concave
- constraints are linear
- if primal constraint k is slack

$$\Rightarrow \lambda_k = 0 \Rightarrow \sum_{i\in[k]} \delta_i = 0 \quad \Rightarrow \boxed{\delta_i = 0 \ \forall i \in [k]}$$

- terms $\delta \ln \delta$ are difficult to compute for small positive
- while we may define $\boxed{0 \ln 0 = \lim_{\delta\to 0} \delta \ln \delta = 0}$

  $\delta \ln \delta$ is not differentiable at 0
- the objective is infinitely differentiable at positive δ

δ	$\ln \delta$	$\delta \ln \delta$
10	2.3025	23.025850929940461
1	0.0000	0.000000000000000
0.1	¯2.3025	¯0.230258509299405
0.01	¯4.6051	¯0.046051701859881
0.001	¯6.9077	¯0.006907755278982
0.0001	¯9.2103	¯0.000921034037198
0.00001	¯11.5129	¯0.000115129254650
0.000001	¯13.8155	¯0.000013815510558
0.0000001	¯16.1180	¯0.000001611809565
0.00000001	¯18.4206	¯0.000000184206807
0.000000001	¯20.7232	¯0.000000020723266
0.0000000001	¯23.0258	¯0.000000002302585
0.00000000001	¯25.3284	¯0.000000000253284
0.000000000001	¯27.6310	¯0.000000000027631
0.0000000000001	¯29.9336	¯0.000000000002993
0.00000000000001	¯32.2361	¯0.000000000000322
0.000000000000001	¯34.5387	¯0.000000000000035



Degrees of Difficulty

The "degree of difficulty" of a dual GP is the difference between
n = # of variables
(= # of primal terms)
m+1 = # dual equality constraints
(normality + orthogonality)

$$\boxed{\text{Degrees of Difficulty} = n - (m+1)}$$

If DGP has zero degree of difficulty, then no optimization is necessary... there is a single feasible solution.

EXPONENTIAL FORM OF PRIMAL GP PROBLEM

While posynomials are not in general convex (e.g., $x^{1/2} = \sqrt{x}$ is a concave function), a change of variables yields an equivalent problem which is both convex and separable!

Consider a change of variables

$$\boxed{u_j = \ln x_j, \quad \text{i.e., } x_j = e^{u_j}}$$

$$g_k(x) = \sum_{i\in[k]} c_i \prod_{j=1}^{m} x_j^{a_{ij}} \qquad \boxed{\textit{posynomial}}$$

becomes $$g_k(u) = \sum_{i\in[k]} c_i \prod_{j=1}^{m} (e^{u_j})^{a_{ij}} = \sum_{i\in[k]} c_i e^{\sum_j a_{ij}u_j}$$

$$= \sum_{i\in[k]} c_i e^{z_i} \text{ where } z_i = \sum_{j=1}^{m} a_{ij}u_j$$

The primal GP problem becomes

$$\text{Minimize } \sum_{i\in[0]} c_i e^{z_i}$$

subject to

$$\sum_{i\in[k]} c_i e^{z_i} \le 1,\ k=1,2,\cdots p$$

$$z_i = \sum_{j=1}^{m} a_{ij}u_j \ \forall\ i=1,2,\cdots n$$

$$\text{i.e., } z = Au$$

e^{z_i} is a convex function of z_i, and so

$\sum_{i\in[k]} c_i e^{z_i}$ is a convex function of z,

Hence this nonlinear programming problem is convex, and has desirable properties such as the sufficiency of the K-K-T conditions, etc.

The functions are also separable!

Storage Tank Example

posynomial formulation

$$\text{Minimize } 2\pi R^2 + 2\pi Rh$$

subject to

$$1000\ h^{-1} R^{-2} \le 1$$

$$h > 0 \ \&\ R > 0$$

exponential formulation

$$\text{Minimize } 2\pi e^{z_1} + 2\pi e^{z_2}$$

$$\text{subject to } 100\ e^{z_3} \le 1$$

$$\left.\begin{aligned} 2u_1 &= z_1 \\ u_1 + u_2 &= z_2 \\ -u_1 - 2u_2 &= z_3 \end{aligned}\right\}$$

(no sign restrictions!)

Subsidiary Problems

Sometimes the primal variables cannot be determined from the dual solution!

primal

$$\text{Minimize } x_1x_2 + x_1^{-1}x_2^{-1}$$

$$\text{subject to } \frac{1}{4} x_1^{1/2} + x_2 \le 1$$

$$x_1 > 0,\ x_2 > 0$$

dual

$$\text{Max } -\delta_1 \ln \delta_1 - \delta_2 \ln \delta_2 + d_3 \ln 1/4 - \delta_3 \ln \delta_3 - \delta_4 \ln \delta_4 + \lambda_1 \ln \lambda_1$$

subject to

$$\delta_1 + \delta_2 = \lambda_0 = 1$$

$$\delta_3 + \delta_4 = \lambda_1$$

$$\delta_1 - \delta_2 + 1/2\ \delta_3 = 0$$

$$\delta_1 - \delta_2 + \delta_4 = 0$$

$$\delta_i \ge 0,\ i = 1,2,3,4$$

$$\lambda_1 \ge 0$$

Example

The GP dual has the *(unique!)* optimal solution

$$\delta_1^* = \delta_2^* = 1/2\ ;\quad \delta_3^* = \delta_4^* = \lambda_1^* = 0,\quad v(\delta^*,\lambda^*) = 2$$

$$\Rightarrow \begin{cases} x_1x_2 = 1/2 \times v(\delta^*,\lambda^*) \\ x_1^{-1}x_2^{-1} = 1/2 \times v(\delta^*,\lambda^*) \end{cases}$$

$$\Rightarrow \begin{cases} \ln x_1 + \ln x_2 = \ln 1 = 0 \\ -\ln x_1 - \ln x_2 = \ln 1 = 0 \end{cases}$$

which has non-unique solution set:

$$\{ (x_1, x_2) \mid x_1x_2 = 1,\ x_1 \ge 0,\ x_2 \ge 0 \}$$

For all x in this solution set, i.e.,

$$\{ (x_1, x_2) \mid x_1x_2 = 1,\ x_1 \ge 0,\ x_2 \ge 0 \}$$

$g_0(x) = 2$, but in general, $g_1(x) \not\le 1$

E.g., $g_1(1,1) = 1.25$

In addition, the optimal Lagrange multipliers for the orthogonality constraints are not unique, leading to the same result!

Subsidiary Problems

When the usual procedure fails to determine an optimal primal solution, we may solve a "subsidiary" problem:

Minimize z

subject to

$$c_i \prod_{j=1}^{m} x_j^{a_{ij}} \le \frac{\delta_i^* g_k^*}{\lambda_k^*} \text{ if } \lambda_k^* > 0$$

$$g_k(x) \le z \quad \text{if } \lambda_k^* = 0$$

$$x_j > 0,\ j=1,\cdots m$$

where

$$g_0^* = v(\delta^*,\lambda^*)$$

$$g_k^* = 1 \text{ if } k = 1,2,\cdots p$$

Subsidiary Problems

That is, the values of terms which were determined by the original dual solution are fixed, and the maximum of the posynomials in the slack constraints is minimized, so that at least one additional constraint becomes tight!

Minimize z
subject to

$$c_i \prod_{j=1}^{m} x_j^{a_{ij}} \le \frac{\delta_i^* g_k^*}{\lambda_k^*} \text{ if } \lambda_k^* > 0$$

$$g_k(x) \le z \quad \text{if } \lambda_k^* = 0$$

$$x_j > 0,\ j=1, \cdots m$$

This then provides additional information about X.*

Example:

primal

Minimize $x_1x_2 + x_1^{-1}x_2^{-1}$
subject to $\frac{1}{4}x_1^{1/2} + x_2 \le 1$
$x_1 > 0,\ x_2 > 0$

solution:

$\delta_1^* = \delta_2^* = 1/2$;
$\delta_3^* = \delta_4^* = \lambda_1^* = 0$
$v(\delta^*, \lambda^*) = 2$

$\Rightarrow$

subsidiary problem

Minimize x_3
subject to $x_1x_2 \le 1$
$x_1^{-1}x_2^{-1} \le 1$
$1/4\, x_1^{1/2}x_3^{-1} + x_2x_3^{-1} \le 1$
$x_1>0,\ x_2>0,\ x_3>0$

$1/4\, x_1^{1/2} + x_2 \le x_3$

Suppose that each posynomial of a GP problem consists of a *single* term:

Minimize $c_1 \prod_{j=1}^{m} x_j^{a_{1j}}$
subject to

$$c_2 \prod_{j=1}^{m} x_j^{a_{2j}} \le 1$$
$$\vdots$$
$$c_p \prod_{j=1}^{m} x_j^{a_{pj}} \le 1$$
$$x > 0$$

Solving Posynomial GP by Condensation

Taking logarithms everywhere yields the equivalent problem

Minimum $\ln c_1 + \sum_{j=1}^{m} a_{1j} \ln x_j$
subject to

$$\begin{cases} \ln c_2 + \sum_{j=1}^{m} a_{2j} \ln x_j \le 0 \\ \vdots \\ \ln c_p + \sum_{j=1}^{m} a_{pj} \ln x_j \le 0 \end{cases}$$

... which, by the change of variable

$$u_j = \ln x_j \Leftrightarrow x_j = e^{u_j}$$

becomes the LP:

$\ln c_1 +$ Minimum $\sum_{j=1}^{m} a_{1j}\, i_j$
subject to

$$\begin{cases} \sum_{j=1}^{m} a_{2j} u_j \le -\ln c_2 \\ \vdots \\ \sum_{j=1}^{m} a_{pj}\, u_j \le -\ln c_p \end{cases}$$

(u_j unrestricted in sign)

Solving GP Problem via Condensation

Let u_i = term #i of a posynomial and apply the A-G inequality:

Arithmetic-Geometric Mean Inequality

$$\sum_i u_i \ge \prod_i \left(\frac{u_i}{\delta_i}\right)^{\delta_i}$$

for all δ satisfying $\sum_i \delta_i = 1,\ \delta_i \ge 0$

$$\sum_i c_i \prod_{j=1}^{m} x_j^{a_{ij}} \ge \prod_i \left(\frac{c_i \prod_{j=1}^{m} x_j^{a_{ij}}}{\delta_i}\right)^{\delta_i}$$

For a given δ, the expression on the right is a monomial, i.e., a single-term posynomial!

$$\sum_i c_i \prod_{j=1}^{m} x_j^{a_{ij}} \ge \prod_i \left(\frac{c_i \prod_{j=1}^{m} x_j^{a_{ij}}}{\delta_i}\right)^{\delta_i} = \left(\prod_i \left(\frac{c_i}{\delta_i}\right)^{\delta_i}\right) \prod_{j=1}^{m} x_j^{\sum_i a_{ij}\delta_i}$$

$$= C(\delta) \prod_{j=1}^{m} x_j^{a_i(\delta)}$$

Each choice of δ yields a monomial underestimate of the posynomial!

where $C(\delta) = \left(\prod_i \left(\frac{c_i}{\delta_i}\right)^{\delta_i}\right)$

$a_i(\delta) = \sum_i a_{ij}\delta_i$

If $\left(\frac{u_i}{\delta_i}\right)$ is the same for all i, i.e., $\left(\frac{u_i}{\delta_i}\right) = r$,

then the A-G Mean Inequality is tight, i.e.,

$$\sum_i c_i \prod_{j=1}^{m} x_j^{a_{ij}} = C(\delta) \prod_{j=1}^{m} x_j^{a_i(\delta)}$$

What value of δ will yield this equality?

$$\left(\frac{u_i}{\delta_i}\right) = r \Rightarrow \left(\frac{u_i}{r}\right) = \delta_i \Rightarrow \quad \sum_i \left(\frac{u_i}{r}\right) = \sum_i \delta_i = 1 \quad \Rightarrow r = \sum_i u_i$$

$$\Rightarrow \delta_i = \frac{u_i}{\sum_{j\in[k]} u_j}$$

A-G Mean Inequality $\Rightarrow$ $$\boxed{\sum_i c_i \prod_{j=1}^{m} x_j^{a_{ij}} \geq C(\delta) \prod_{j=1}^{m} x_j^{a_i(\delta)}}$$

where $C(\delta) = \left(\prod_i \left(\frac{c_i}{\delta_i}\right)^{\delta_i}\right)$ & $a_i(\delta) = \sum_i a_{ij}\delta_i$

with equality if & only if $\delta_i = \dfrac{u_i}{\sum_{j\in[k]} u_j}$

i.e., if & only if δ is the fraction of the posynomial contributed by term #i.

Therefore,

- condensing the objective posynomial gives us an underestimate of the cost
- condensing a constraint posynomial gives us an approximation to the feasible region which contains the original feasible region
- the approximations are exact at a point $\hat{x}$ if

$$\delta_i = \frac{\text{value of term \#i at } \hat{x}}{\text{value of posynomial at } \hat{x}}$$

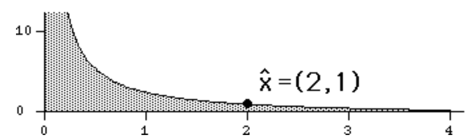
Example Find the condensation of

$$g(x) = \tfrac{1}{3}x_1x_2 + \tfrac{1}{6}x_1 \leq 1 \quad \text{at } \hat{x} = (2,1)$$

$$g(\hat{x}) = \tfrac{1}{3}\hat{x}_1\hat{x}_2 + \tfrac{1}{6}\hat{x}_1 = \tfrac{2}{3} + \tfrac{1}{3} = 1$$

$$\Rightarrow \begin{cases} \delta_1 = \dfrac{\frac{1}{3}\hat{x}_1\hat{x}_2}{g(\hat{x})} = \dfrac{2}{3} \\ \delta_2 = \dfrac{\frac{1}{6}\hat{x}_1}{g(\hat{x})} = \dfrac{1}{3} \end{cases}$$

$$g(x) = \tfrac{1}{3}x_1x_2 + \tfrac{1}{6}x_1 \leq 1$$



$\delta = \left(\frac{2}{3}, \frac{1}{3}\right)$

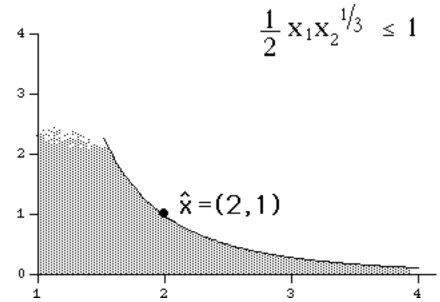
$$C(\delta) = \left(\frac{c_1}{\delta_1}\right)^{\delta_1} \times \left(\frac{c_2}{\delta_2}\right)^{\delta_2} = \left(\frac{1/3}{2/3}\right)^{2/3} \times \left(\frac{1/6}{1/3}\right)^{1/3} = \left(\frac{1}{2}\right)^{2/3} \times \left(\frac{1}{2}\right)^{1/3} = \frac{1}{2}$$

$$a_1(\delta) = a_{11}\delta_1 + a_{21}\delta_2 = 1 \times 2/3 + 1 \times 1/3 = 1$$

$$a_2(\delta) = a_{12}\delta_1 + a_{22}\delta_2 = 0 \times 2/3 + 1 \times 1/3 = 1/3$$

$$C(\delta) \prod_{j=1}^{m} x_j^{a_i(\delta)} = \frac{1}{2} x_1 x_2^{1/3}$$

$$\frac{1}{2} x_1 x_2^{1/3} \leq 1$$



Solving GP via LP

Step 0 Choose an initial $\hat{x}$

Step 1 Evaluate all terms & all posynomials, and compute, for each term i,

$$\delta_i = \frac{\text{value of term \#i at } \hat{x}}{\text{value of posynomial at } \hat{x}}$$

Step 2 Condense all posynomials into monomials, using weights δ_i.

Solving GP via LP

Step 3 Take logarithms of monomial objective & constraints to get an LP

Step 4 Solve the LP and exponentiate the optimal value of ln x^* to get x^*.

Step 5 If $x^* \approx \hat{x}$, STOP; otherwise, let $\hat{x} = x^*$ and return to step 1.