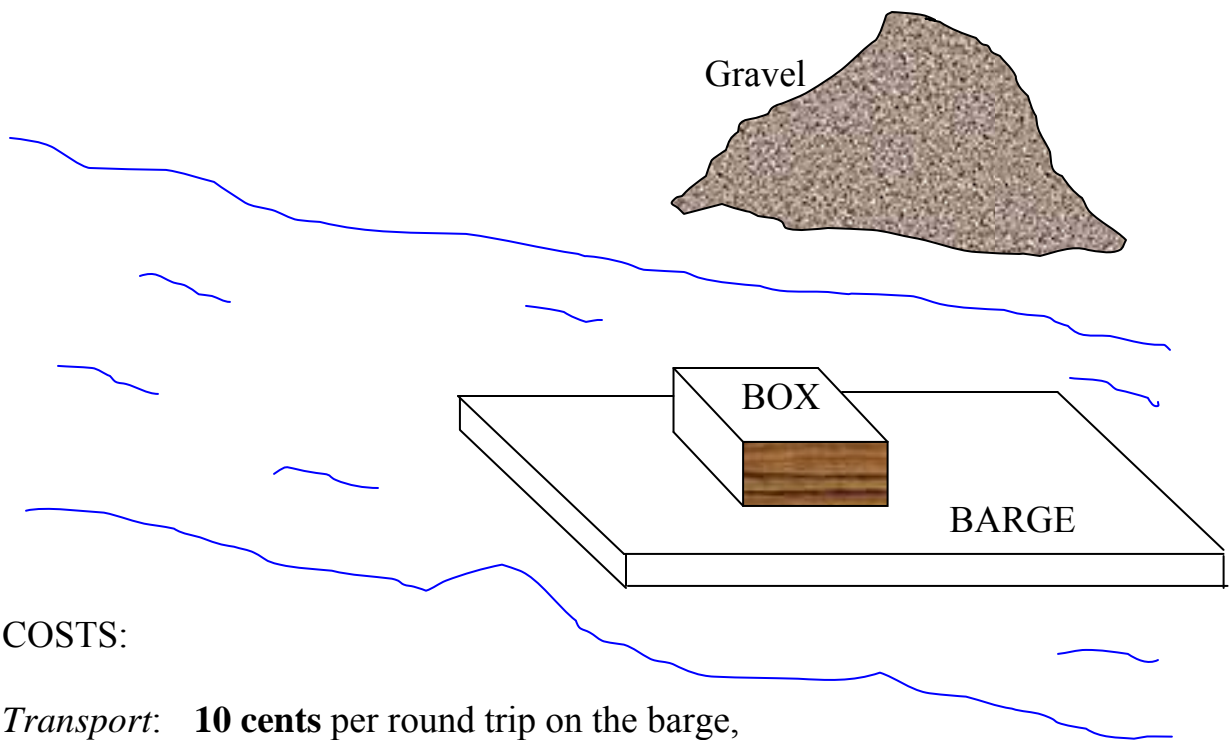


GEOMETRIC PROGRAMMING

Geometric Programming (**GP**) is a special form of NLP that has many interesting features, and finds a lot of use in engineering design. We motivate its study through an example....

GRAVEL BOX DESIGN

400 m^3 of gravel is to be ferried across a river on a barge. A box (with the top open) is to be designed for this purpose



COSTS:

Transport: **10 cents** per round trip on the barge,

Materials: Sides and bottom of box: **\$10/m²**

Ends of the box: **\$20/m²**

What are the optimal dimensions of the box?

Instead of attempting to directly solve this messy nonlinear system (e.g. by Newton's approach), GEOMETRIC PROGRAMMING uses a different approach...

POSYNOMIALS are polynomials with positive coefficients for all terms, e.g.,

$$1) 40x_1^{-1}x_2^{-1}x_3^{-1} + 40x_2x_3 + 20x_1x_2^{-1}$$

$$2) 0.2x_1x_2^{0.5}x_3 + 6x_1x_2^{1/3} + 3x_3$$

In general,

$$g(\mathbf{x}) = \sum_i c_i \prod_j x_j^{a_{ij}}$$

$$= c_1x_1^{a_{11}}x_2^{a_{12}} \dots x_m^{a_{1m}} + c_2x_1^{a_{21}}x_2^{a_{22}} \dots x_m^{a_{2m}} + \dots + c_nx_1^{a_{n1}}x_2^{a_{n2}} \dots x_m^{a_{nm}}$$

with all c_i being strictly positive.

Geometric Programming deals with such functions...

GEOMETRIC PROGRAMMING

All $c_i > 0$

(posynomials)

POSYNOMIAL GP

Some $c_i < 0$

(signomials)

SIGNOMIAL GP

To minimize a posynomial

$$f(\mathbf{x}) = \sum_{i=1}^n c_i \prod_{j=1}^m x_j^{a_{ij}} \quad (n \text{ terms and } m \text{ variables})$$

the necessary condition is

$$\nabla f(\mathbf{x}) = \mathbf{0}$$

$$\text{i. e., } \frac{\partial f}{\partial x_k}(\mathbf{x}) = \sum_{i=1}^n c_i \frac{\partial}{\partial x_k} \left(\prod_{j=1}^m x_j^{a_{ij}} \right) = 0 \quad \forall k$$

$$\Rightarrow \sum_{i=1}^n c_i a_{ik} x_k^{a_{ik}-1} \left(\prod_{\substack{j=1 \\ j \neq k}}^m x_j^{a_{ij}} \right) = 0 \quad \forall k$$

Assuming that $x_k > 0$, we may multiply both sides by x_k to obtain

$$\sum_{i=1}^n c_i a_{ik} x_k^{a_{ik}} \left(\prod_{\substack{j=1 \\ j \neq k}}^m x_j^{a_{ij}} \right) = 0 \quad \forall k$$

$$\text{i. e., } \boxed{\sum_{i=1}^n a_{ik} \left(c_i \prod_{j=1}^m x_j^{a_{ij}} \right) = 0 \quad \forall k}$$

TERM i

This nonlinear system may in general be very difficult to solve!

Let $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_m^*)$ be the optimal solution,

$f^* = f(\mathbf{x}^*)$ be the optimal objective value,

δ_i^* = fraction of the optimal cost f^* which is the contribution due to term i

$$= \frac{\text{value of term } i}{f^*} = \frac{c_i \prod_{j=1}^m (x_j^*)^{a_{ij}}}{f^*} \geq 0$$

Obviously,

$$\delta_1^* + \delta_2^* + \delta_3^* + \dots + \delta_n^* = 1$$

\approx NORMALITY

Also,

$$c_i \prod_{j=1}^m (x_j^*)^{a_{ij}} = \delta_i^* f^* \quad \forall i$$

Substituting into the necessary conditions for optimality namely

$$\sum_{i=1}^n a_{ik} \left(c_i \prod_{j=1}^m (x_j^*)^{a_{ij}} \right) = 0 \quad \text{for } k = 1, 2, \dots, m$$

we have $\sum_{i=1}^n a_{ij} \delta_i^* f^* = 0 \quad \text{for } j = 1, 2, \dots, m$

Since $f^* > 0$, we therefore have

$$\sum_{i=1}^n a_{ij} \delta_i^* = 0 \quad \text{for } j = 1, 2, \dots, m$$

\approx ORTHOGONALITY

Note that the nonlinear system (in \mathbf{x}) has been transformed into an equivalent LINEAR system (δ).

BACK TO THE GRAVEL BOX DESIGN PROBLEM...

$$\begin{aligned}
 f(\mathbf{x}) &= 40x_1^{-1}x_2^{-1}x_3^{-1} + 40x_2x_3 + 20x_1x_3 + 10x_1x_2 \\
 &= 40x_1^{-1}x_2^{-1}x_3^{-1} + 40x_1^0x_2x_3 + 20x_1x_2^0x_3 + 10x_1x_2x_3^0 \\
 &\quad \text{(transportation)} \qquad \text{(ends)} \qquad \text{(sides)} \qquad \text{(bottom)}
 \end{aligned}$$

Let

δ_1^* = fraction of optimum cost due to transportation

δ_2^* = fraction of optimum cost due to the ends

δ_3^* = fraction of optimum cost due to the sides

δ_4^* = fraction of optimum cost due to the bottom

Then

$$\text{NORMALITY} \Rightarrow \delta_1^* + \delta_2^* + \delta_3^* + \delta_4^* = 1$$

$$\text{ORTHOGONALITY} \Rightarrow \sum_{i=1}^4 a_{ij} \delta_i^* = 0 \quad \text{for } j = 1, 2, 3.$$

Thus

$$\begin{aligned}
 \delta_1^* + \delta_2^* + \delta_3^* + \delta_4^* &= 1 \\
 -\delta_1^* + \delta_3^* + \delta_4^* &= 0 && \text{(for } x_1) \\
 -\delta_1^* + \delta_2^* + \delta_4^* &= 0 && \text{(for } x_2) \\
 -\delta_1^* + \delta_2^* + \delta_3^* &= 0 && \text{(for } x_3)
 \end{aligned}$$

Gauss-Jordan elimination of this linear system yields

$$\delta_1^* = 2/5 (= 40\%), \quad \delta_2^* = 1/5 (= 20\%), \quad \delta_3^* = 1/5 (= 20\%), \quad \delta_4^* = 1/5 (= 20\%),$$

Note that we have not used the cost coefficients c_i anywhere as yet! Thus, IRRESPECTIVE of the cost coefficients, the box should be designed so that transportation accounts for 40% of the total cost, while the remaining 60% is split evenly between the sides, the ends and the bottom. These optimal "weights" are independent of the actual cost coefficients. For example, if transportation costs were to rise from 10 cents/trip to \$1/trip (or even \$100/trip!), the design should still be such that the total cost is divided in the same proportion.

Now consider the Orthogonality conditions for our example once again. It follows that

$$\begin{aligned}
 x_1^{*(-\delta_1^*+\delta_3^*+\delta_4^*)} x_2^{*(-\delta_1^*+\delta_2^*+\delta_4^*)} x_3^{*(-\delta_1^*+\delta_2^*+\delta_3^*)} &= 1 \\
 \Rightarrow (x_1^{*-1} x_2^{*-1} x_3^{*-1})^{\delta_1^*} (x_2^* x_3^*)^{\delta_2^*} (x_1^* x_3^*)^{\delta_3^*} (x_1^* x_2^*)^{\delta_4^*} &= 1 \\
 \Rightarrow \left(\frac{f^* \delta_1^*}{40}\right)^{\delta_1^*} \left(\frac{f^* \delta_2^*}{40}\right)^{\delta_2^*} \left(\frac{f^* \delta_3^*}{20}\right)^{\delta_3^*} \left(\frac{f^* \delta_4^*}{10}\right)^{\delta_4^*} &= 1 \\
 \Rightarrow f^{*(\delta_1^*+\delta_2^*+\delta_3^*+\delta_4^*)} \left(\frac{\delta_1^*}{40}\right)^{\delta_1^*} \left(\frac{\delta_2^*}{40}\right)^{\delta_2^*} \left(\frac{\delta_3^*}{20}\right)^{\delta_3^*} \left(\frac{\delta_4^*}{10}\right)^{\delta_4^*} \\
 &= f^* \left(\frac{\delta_1^*}{40}\right)^{\delta_1^*} \left(\frac{\delta_2^*}{40}\right)^{\delta_2^*} \left(\frac{\delta_3^*}{20}\right)^{\delta_3^*} \left(\frac{\delta_4^*}{10}\right)^{\delta_4^*} = 1 \\
 f^* &= \left(\frac{40}{\delta_1^*}\right)^{\delta_1^*} \left(\frac{40}{\delta_2^*}\right)^{\delta_2^*} \left(\frac{20}{\delta_3^*}\right)^{\delta_3^*} \left(\frac{10}{\delta_4^*}\right)^{\delta_4^*}
 \end{aligned}$$

$$\begin{aligned}
&= \left(\frac{40}{2/5}\right)^{2/5} \cdot \left(\frac{40}{1/5}\right)^{1/5} \cdot \left(\frac{20}{1/5}\right)^{1/5} \cdot \left(\frac{10}{1/5}\right)^{1/5} \\
&= (100)^{2/5} \cdot (200)^{1/5} \cdot (100)^{1/5} \cdot (50)^{1/5} \\
&= (10,000 \cdot 200 \cdot 100 \cdot 50)^{1/5} = (10^{10})^{1/5} = 100.
\end{aligned}$$

Thus the minimum cost for the box is **\$100**.

More generally, from the Orthogonality conditions it follows that

$$\begin{aligned}
\prod_{j=1}^m (x_j^*)^{\sum_{i=1}^n a_{ij} \delta_i^*} &= 1 \\
\Rightarrow \prod_{j=1}^m \left(\prod_{i=1}^n (x_j^*)^{a_{ij} \delta_i^*} \right) &= \prod_{i=1}^n \left(\prod_{j=1}^m (x_j^*)^{a_{ij}} \right)^{\delta_i^*} = 1 \\
\Rightarrow \prod_{i=1}^n \left(\frac{f^* \delta_i^*}{c_i} \right)^{\delta_i^*} &= 1 \quad \Rightarrow (f^*)^{(\sum_{i=1}^n \delta_i^*)} \prod_{i=1}^n \left(\frac{\delta_i^*}{c_i} \right)^{\delta_i^*} = 1 \\
\Rightarrow f^* &= \prod_{i=1}^n \left(\frac{c_i}{\delta_i^*} \right)^{\delta_i^*}
\end{aligned}$$

BACK AGAIN TO THE GRAVEL BOX PROBLEM...

At this point, we have the optimal value of the objective and the fraction of this value that each term contributes.

Question: How do we find the optimal values of the primal variables (x_i^*) ?

For our example, of the optimal cost, 40% ($\delta_1^* = 2/5$) come from transport, 20% ($\delta_2^* = 1/5$) come from the sides, 20% ($\delta_3^* = 1/5$) come from the ends, and 20% ($\delta_4^* = 1/5$) come from the bottom.

In other words,

- $40x_1^{-1}x_2^{-1}x_3^{-1} = \frac{2}{5}(100) = 40 \Rightarrow x_1^{-1}x_2^{-1}x_3^{-1} = 1$
- $40x_2x_3 = \frac{1}{5}(100) = 20 \Rightarrow x_2x_3 = 0.5$
- $20x_1x_3 = \frac{1}{5}(100) = 20 \Rightarrow x_1x_3 = 1$
- $10x_1x_2 = \frac{1}{5}(100) = 20 \Rightarrow x_1x_2 = 1$

This nonlinear system is easily linearized by taking logarithms:

$$-\ln x_1^* - \ln x_2^* - \ln x_3^* = \ln 1 = 0$$

$$\ln x_2^* + \ln x_3^* = \ln 0.5$$

$$\ln x_1^* + \ln x_3^* = \ln 1 = 0$$

$$\ln x_1^* + \ln x_2^* = \ln 2$$

$$\text{Let } z_1 = \ln x_1^*, \quad z_2 = \ln x_2^*, \quad z_3 = \ln x_3^*$$

$$\Rightarrow -z_1 - z_2 - z_3 = 0$$

$$z_2 + z_3 = \ln 0.5 \qquad z_1 = \ln 2$$

$$z_1 + z_3 = 0 \qquad \Rightarrow z_2 = 0$$

$$z_1 + z_2 = \ln 2 \qquad z_3 = \ln 0.5$$

$$\text{Therefore, } x_1^* = \exp(z_1) = \exp(\ln 2) \quad \Rightarrow x_1^* = 2 \text{ m.}$$

$$x_2^* = \exp(z_2) = \exp(0) \quad \Rightarrow x_2^* = 1 \text{ m.}$$

$$x_3^* = \exp(z_3) = \exp(\ln 0.5) \quad \Rightarrow x_3^* = 0.5 \text{ m.}$$

The volume of the box is $2 \cdot 1 \cdot (0.5) = 1 \text{ m}^3$

The number of trips = 400, and the minimum total cost = **\$100**

More generally,

$$c_i \prod_{j=1}^m (x_j^*)^{a_{ij}} = f^* \delta_i^* \quad i = 1, 2, \dots, n$$

and taking logarithms on both sides we get a system of n linear equations in the variables $z_j = \log x_j^*$. We solve this system for z_j and obtain $x_j^* = \exp(z_j)$

Unfortunately, things are not always THIS straightforward!

In the gravel box design problem there were

- 3 design variables ($m=3$); each generated one orthogonality equation
- 1 normality constraint equation

Furthermore, we had

- 4 objective function terms ($n=4$); each generated a δ_i variable

Thus we had 4 variables (the δ_i 's) and $(3+1=4)$ equations in the dual system and

we were able to obtain a unique solution. In general, this need not be the case...

DEFINITION1: The quantity $D = n-(m+1)$ is referred to as the **Number of Degrees of Difficulty** for the problem. So $D = (\text{No. of } \delta\text{'s}) - (\text{No. of equations})$
 $= (\text{No. of terms in the Primal objective}) - (1+\text{No. of Primal variables})$

The gravel box design problem has zero degrees of difficulty; and we could thus solve for unique values of δ_i . In the general case D could be more than zero since it depends on the values of n (the no. of terms) and m (the no. of variables).

If $D > 0$ and the system of $(m+1)$ equations has a solution, then simple linear algebra tells us that the system has *infinitely many* solutions. Recall that the system of equations was derived from the necessary conditions; thus we have infinitely many points satisfying the necessary conditions.

HOWEVER, not all of these will satisfy the sufficient conditions. In other words, a feasible δ vector need NOT lead to an optimal design \mathbf{x}^* .

Question: So how do we find a solution δ (among the infinitely many solutions to normality and orthogonality) that will lead to an optimal design \mathbf{x}^* ?

Answer: We generalize what we did earlier by introducing the **GP DUAL** problem.

The GP DUAL is based on the **ARITHMETIC-GEOMETRIC MEAN INEQUALITY:**

$$\sum_{i=1}^n \delta_i v_i \geq \prod_{i=1}^n v_i^{\delta_i}, \quad \text{where } \sum_{i=1}^n \delta_i = 1 \text{ and } \delta_i \geq 0$$

(Arith. Mean) (Geom. Mean)

In particular, suppose $v_i = \frac{c_i \prod_{j=1}^m (x_j)^{a_{ij}}}{\delta_i}$

$$\Rightarrow \sum_{i=1}^n \delta_i v_i = \sum_{i=1}^n c_i \prod_{j=1}^m x_j^{a_{ij}} \approx \text{Primal objective function } f(\mathbf{x})$$

The AM-GM inequality thus yields

$$\begin{aligned} f(\mathbf{x}) &\geq \prod_{i=1}^n \left(\frac{c_i \prod_{j=1}^m (x_j)^{a_{ij}}}{\delta_i} \right)^{\delta_i} = \prod_{i=1}^n \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \left(\prod_{j=1}^m (x_j)^{a_{ij}} \right)^{\delta_i} \\ &\Rightarrow f(\mathbf{x}) \geq \prod_{i=1}^n \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \cdot \prod_{i=1}^n \left(\prod_{j=1}^m (x_j)^{a_{ij}} \right)^{\delta_i} \\ &\Rightarrow f(\mathbf{x}) \geq \prod_{i=1}^n \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \cdot \prod_{j=1}^m \left(\prod_{i=1}^n (x_j)^{a_{ij} \delta_i} \right) \\ &\Rightarrow f(\mathbf{x}) \geq \prod_{i=1}^n \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \cdot \left(\prod_{j=1}^m (x_j)^{\sum_{i=1}^n a_{ij} \delta_i} \right) \end{aligned} \quad (A)$$

Now, suppose that orthogonality is satisfied by the δ_i , i.e., $\sum_{i=1}^n a_{ij} \delta_i = 0$ for all $j=1,2,\dots,m$.

Then (A) reduces to

$$f(\mathbf{x}) \geq \prod_{i=1}^n \left(\frac{c_i}{\delta_i}\right)^{\delta_i}$$

Primal Objective
Dual Objective (define as...)

Thus for \mathbf{x} feasible in the Primal, and for δ_i values that satisfy orthogonality, normality and nonnegativity, PRIMAL O.F. \geq DUAL O.F. In particular, the optimal primal value is bounded below by the dual objective defined above at *any* vector δ satisfying orthogonality, normality and nonnegativity

Now consider the optimum solution \mathbf{x}^* for the Primal, and δ^* defined by

$$\delta_i^* = \frac{c_i \prod_{j=1}^m (x_j^*)^{a_{ij}}}{f^*}, \quad \text{where } f^* \equiv f(\mathbf{x}^*)$$

We already saw (page 222) that this nonnegative vector δ^* satisfies both Orthogonality and Normality.

Using this vector for the RHS of the result just obtained we have

$$\begin{aligned} \prod_{i=1}^n \left(\frac{c_i}{\delta_i^*}\right)^{\delta_i^*} &= \prod_{i=1}^n \left(\frac{f^*}{\prod_{j=1}^m (x_j^*)^{a_{ij}}}\right)^{\delta_i^*} = \left\{ \prod_{i=1}^n (f^*)^{\delta_i^*} \right\} \div \left\{ \prod_{i=1}^n \left(\prod_{j=1}^m (x_j^*)^{a_{ij}} \right)^{\delta_i^*} \right\} \\ &= ((f^*)^{\sum_{i=1}^n \delta_i^*}) \div \left(\prod_{j=1}^m (x_j^*)^{\sum_{i=1}^n a_{ij} \delta_i^*} \right) \\ &= f^* \quad \begin{array}{l} \downarrow \\ (=1) \\ \downarrow \end{array} \quad \begin{array}{l} \downarrow \\ (=0) \\ \downarrow \end{array} \end{aligned}$$

because δ^* satisfies both Normality and Orthogonality.

VALUE OF DUAL OBJECTIVE WITH δ^* DEFINED ABOVE**= VALUE OF OPTIMAL PRIMAL OBJECTIVE!**

When the problem has zero degrees of difficulty there can be only one nonnegative δ vector that satisfies Normality and Orthogonality and the value of the dual objective at this vector is equal to that of the primal objective, as we saw earlier with the gravel box example.

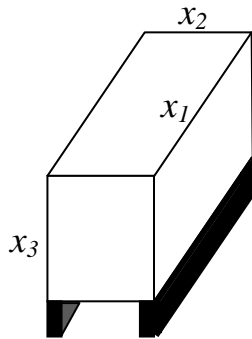
However, when the number of degrees of difficulty is positive there are typically infinitely many nonnegative δ vectors that satisfy Normality and Orthogonality! Since the dual objective at each of these is less than or equal to the optimal primal objective, it is natural to define the dual problem as:

MAXIMIZE the dual objective,
subject to ORTHOGONALITY, NORMALITY and NONNEGATIVITY.

The maximum value will be equal to f^* (the minimum value for the original primal problem).

BACK AGAIN TO THE GRAVEL BOX PROBLEM...

In order to make it is easy to slide the box on to the barge, "runners" should be placed on the bottom along the length of the box:



Cost (per runner) per
meter length = \$2.50

The total cost function now becomes

$$f(\mathbf{x}) = \underbrace{40x_1^{-1}x_2^{-1}x_3^{-1} + 40x_2x_3 + 20x_1x_3 + 10x_1x_2}_{\text{(old cost)}} + \underbrace{5x_1}_{\text{(runners)}}$$

Here, $n = \text{no. of terms} = 5$,

$$m = \text{no. of variables} = 3 \quad \Rightarrow \quad D = 5 - (1 + 3) = 1$$

The dual objective function to be maximized is

$$\left(\frac{40}{\delta_1}\right)^{\delta_1} \cdot \left(\frac{40}{\delta_2}\right)^{\delta_2} \cdot \left(\frac{20}{\delta_3}\right)^{\delta_3} \cdot \left(\frac{10}{\delta_4}\right)^{\delta_4} \cdot \left(\frac{5}{\delta_5}\right)^{\delta_5}$$

and the system of equations is

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 = 1 \quad \text{(NORMALITY)}$$

$$-\delta_1 + \delta_3 + \delta_4 + \delta_5 = 0$$

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

$$-\delta_1 + \delta_2 + \delta_3 = 0 \quad \text{(ORTHOGONALITY)}$$

We have 4 equations in 5 unknowns, and the above system has infinitely many solutions. We need to find the one solution that also maximizes the dual objective function, and we could solve this by GRG or CSM for example.

A better approach is to use the LOG-DUAL objective. Since the logarithmic function is monotonically (strictly) increasing, we could maximize the logarithm of the dual objective (rather than the dual objective itself):

$$\text{Maximize } \ln \left\{ \left(\frac{40}{\delta_1} \right)^{\delta_1} \cdot \left(\frac{40}{\delta_2} \right)^{\delta_2} \cdot \left(\frac{20}{\delta_3} \right)^{\delta_3} \cdot \left(\frac{10}{\delta_4} \right)^{\delta_4} \cdot \left(\frac{5}{\delta_5} \right)^{\delta_5} \right\}$$

i.e.,

$$\begin{aligned} \text{Maximize } \quad & \{ \delta_1 \ln 40 - \delta_1 \ln \delta_1 + \delta_2 \ln 40 - \delta_2 \ln \delta_2 + \delta_3 \ln 20 - \delta_3 \ln \delta_3 \\ & + \delta_4 \ln 10 - \delta_4 \ln \delta_4 + \delta_5 \ln 5 - \delta_5 \ln \delta_5 \} \end{aligned}$$

subject to

Normality, Orthogonality and Nonnegativity

ADVANTAGES

1. The objective function is now SEPARABLE,
2. The O.F. is also concave in the δ_i , which is very desirable since we are maximizing.

OBTAINING BOUNDS ON THE OPTIMAL COST

Lower Bound: In the earlier version (without the runners) we had $\delta_l^* = 2/5$, and $\delta_2^* = \delta_3^* = \delta_4^* = 1/5$. Suppose we include $\delta_5^* = 0$. This new solution is feasible in the dual constraints; therefore the maximum value of the dual objective is at least as high as its value at this solution, i.e., the optimal dual objective

$$\geq \left(\frac{40}{2/5}\right)^{2/5} \cdot \left(\frac{40}{1/5}\right)^{1/5} \cdot \left(\frac{20}{1/5}\right)^{1/5} \cdot \left(\frac{10}{1/5}\right)^{1/5} \cdot \left(\frac{5}{0}\right)^0 = 100$$

But the optimal dual objective = optimal primal objective f^* . Therefore

$$\boxed{f^* \geq 100}$$

[NOTE: $(c/\delta)^\delta$ and $(\delta \ln \delta)$ are both undefined for $\delta=0$. However,

$$\lim_{\delta \rightarrow 0^+} (c/\delta)^\delta = 0 \text{ and } \lim_{\delta \rightarrow 0^+} \delta \ln \delta = 0,$$

\therefore We define $(c/\delta)^\delta = 1$ and $(\delta \ln \delta) = 0$ for $\delta=0$.]

Upper Bound: The dimensions for the box that were found without accounting for the runners are still feasible (since the problem is unconstrained). Therefore the optimal (primal) cost is at least as low as the cost (including runner cost) of the old design of $x_1=2$, $x_2=1$, $x_3=1/2$, i.e.,

$$f^* \leq 40x_1^{-1}x_2^{-1}x_3^{-1} + 40x_2x_3 + 20x_1x_3 + 10x_1x_2 + 5x_1$$

$$\Rightarrow f^* \leq 100 + 5(2) = 110, \text{ i.e.,}$$

Thus, without even solving the problem we can say that $100 \leq f^* \leq 110$ (or

perhaps, $f^* = 105 \pm 5$).

GEOMETRIC PROGRAMMING WITH CONSTRAINTS

GP constraints must be of the form

$$g_k(\mathbf{x}) = \sum_{i \in [k]} c_i \prod_{j=1}^m x_j^{a_{ij}} \leq 1; \quad \text{where } c_i, x_j > 0$$

i.e., of the form (posynomial) ≤ 1 .

Here, $[k]$ is the (sub)set of indices for the terms in the k^{th} constraint.

Often, many constraints involving generalized polynomials can be readily rearranged to meet this format, even if they don't initially appear to be in the correct format. For example:

- $x_1 x_2 \geq 5 \Rightarrow 1/x_1 x_2 \leq 1/5 \Rightarrow 5x_1^{-1} x_2^{-1} \leq 1$
- $x_3 + x_2 \leq 5x_1 \Rightarrow (1/5x_1)(x_3 + x_2) \leq 1 \Rightarrow 0.2x_1^{-1} x_3 + 0.2x_1^{-1} x_2 \leq 1$
- $x_1 - x_2 \geq 10 \Rightarrow (10 + x_2) \leq x_1 \Rightarrow 10x_1^{-1} + x_1^{-1} x_2 \leq 1$

etc., etc., etc.

We now introduce the Primal-Dual pair in constrained Geometric Programming.

PRIMAL GP (P)Minimize $g_0(\mathbf{x})$

$$\text{st } \begin{aligned} g_k(\mathbf{x}) &\leq 1, & k=1,2,\dots,p, \\ x_j &> 0, & j=1,2,\dots,m, \end{aligned}$$

where

$$g_k(\mathbf{x}) = \sum_{i \in [k]} c_i \prod_{j=1}^m x_j^{a_{ij}} \text{ is a posynomial } \forall k,$$

$$[0] \cup [1] \cup [2] \cup \dots \cup [p] = \{1,2,\dots,n\} = I \text{ and } [k] \cap [l] = \emptyset \text{ for } k \neq l.$$

NOTE: Total no. of terms = n , I is an index set of terms and $[k]$ is an index subset of terms in posynomial k .

DUAL GP (D)

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

$$\text{st } \sum_{i \in [k]} \delta_i = \lambda_k, \quad k = 0, 1, \dots, p,$$

$$\sum_{i=1}^n a_{ij} \delta_i = 0, \quad j = 1, 2, \dots, m, \quad (\text{Orthogonality})$$

$$\lambda_0 = 1, \quad (\text{Normality})$$

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

To clarify the dual further, note that we maximize

$$v(\lambda, \delta) = \left\{ \prod_{i \in [0]} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \cdot \prod_{i \in [1]} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \cdot \dots \cdot \prod_{i \in [p]} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \right\} \cdot \lambda_1^{\lambda_1} \lambda_2^{\lambda_2} \dots \lambda_p^{\lambda_p}$$

subject to

$$\delta_1 + \delta_2 + \dots + \delta_{q_0} = 1, \quad \text{where } [0] = \{1, 2, \dots, q_0\} \quad (\text{Normality})$$

$$\left\{ \begin{array}{l} \delta_{q_0+1} + \delta_{q_0+2} + \dots + \delta_{q_1} = \lambda_1, \quad \text{where } [1] = \{q_0 + 1, q_0 + 2, \dots, q_1\} \\ \delta_{q_1+1} + \delta_{q_1+2} + \dots + \delta_{q_2} = \lambda_2, \quad \text{where } [2] = \{q_1 + 1, q_1 + 2, \dots, q_2\} \\ \dots\dots\dots \\ \delta_{q_{p-1}+1} + \delta_{q_{p-1}+2} + \dots + \delta_n = \lambda_p, \quad \text{where } [p] = \{q_{p-1} + 1, q_{p-1} + 2, \dots, n\} \end{array} \right.$$

$$\sum_{k=0}^p \sum_{i \in [k]} a_{ij} \delta_i = 0, \quad j = 1, 2, \dots, m, \quad (\text{Orthogonality})$$

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

Note that λ_0 has been eliminated altogether, and that the other λ_k can also be eliminated by substituting their values in terms of the δ_i into the objective function (in fact, λ_k is actually a Lagrange multiplier for the k^{th} primal constraint).

An example of the Primal-Dual pair:

$$(P) \quad \text{Minimize } g_0(\mathbf{x}) = 10x_1^2 + 10x_1x_2$$

$$\text{st} \quad g_1(\mathbf{x}) = 1000x_1^{-2}x_2^{-1} \leq 1$$

$$x_1, x_2 > 0.$$

Here $n = \text{total no. of terms} = 3$

$$m = \text{total no. of variables} = 2$$

$$p = \text{total no. of constraints} = 1$$

$$[0] = \{1,2\}; [1] = \{3\}, I = \{1,2,3\}$$

(D)

$$\text{Maximize } v(\lambda, \delta) = \left\{ \lambda_0^{\lambda_0} \left(\frac{10}{\delta_1} \right)^{\delta_1} \left(\frac{10}{\delta_2} \right)^{\delta_2} \right\} \left\{ \lambda_1^{\lambda_1} \left(\frac{1000}{\delta_3} \right)^{\delta_3} \right\}$$

$$\text{st} \quad \delta_1 + \delta_2 = \lambda_0$$

$$\delta_3 = \lambda_1$$

$$\left. \begin{array}{ll} (2)\delta_1 + (1)\delta_2 + (-2)\delta_3 = 0 & (\approx j=1) \\ (1)\delta_2 + (-1)\delta_3 = 0 & (\approx j=2) \end{array} \right\} \text{Orthogonality}$$

$$\lambda_0 = 1 \quad \text{Normality}$$

NOTE: In (D), we EFFECTIVELY have 3 variables ($\delta_1, \delta_2, \delta_3$) and 3 constraints

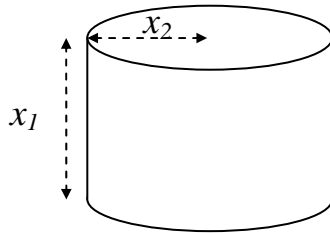
($2\delta_1 + \delta_2 - 2\delta_3 = 0$; $\delta_2 - \delta_3 = 0$; and $\delta_1 + \delta_2 = 1$). Hence, it is easily solved.

Note that this is because $D = n - (m+1) = 3 - (2+1) = 0 \dots$

AN EXAMPLE:

A closed storage tank is to be designed to store oil

- cost of material = $\$1/\text{m}^2$
- required volume = $1000\pi \text{ m}^3$.



Design Variables:

Height = x_1 meters

Radius = x_2 meters

Capacity requirements: $x_1(\pi x_2^2) = (1000\pi)$.

Obviously, we could replace the equality by \geq , i.e.,

$$x_1(\pi x_2^2) \geq (1000\pi) \Rightarrow 1/(x_1 x_2^2) \leq (1/1000) \Rightarrow 1000x_1^{-1}x_2^{-2} \leq 1$$

Thus the primal (P) is

$$\text{Minimize } 2\pi x_2^2 \quad + \quad 2\pi x_1 x_2$$

(top + bottom) (sides)

$$\text{st} \quad 1000x_1^{-1}x_2^{-2} \leq 1; \quad x_1, x_2 > 0.$$

The dual (D) is

$$\text{Maximize } v(\lambda, \delta) = \left(\frac{2\pi}{\delta_1}\right)^{\delta_1} \left(\frac{2\pi}{\delta_2}\right)^{\delta_2} \left(\frac{1000}{\delta_3}\right)^{\delta_3} \lambda_1^{\lambda_1}$$

$$\text{st} \quad \delta_1 + \delta_2 = 1 \quad (\text{Normality})$$

$$\delta_3 = \lambda_1$$

$$\delta_2 - \delta_3 = 0 \quad (\text{Orthogonality } \approx x_1)$$

$$2\delta_1 + \delta_2 - 2\delta_3 = 0 \quad (\text{Orthogonality } \approx x_2)$$

$$\delta_1, \delta_2, \delta_3, \lambda_1 \geq 0$$

Here we have 4 equations in 4 unknowns -- easily solved!

$$\delta_1^* = 1/3; \delta_2^* = 2/3; \delta_3^* = 2/3 (= \lambda_1)$$

Thus in an optimal design the top and bottom should account for 1/3 of the total cost, while the sides should account for the remaining 2/3.

OPTIMAL COST: Optimal dual objective value = 1187.45. Therefore the optimal primal objective = $g_0(\mathbf{x}^*)$ = optimal cost = \$1187.45.

Computing the dimensions:

$$2\pi x_2^2 = (1/3)*g_0(\mathbf{x}^*) = 395.82 \quad \Rightarrow \quad x_2^* = 7.937 \text{ meters}$$

$$2\pi x_1 x_2 = (2/3)*g_0(\mathbf{x}^*) = 791.63 \quad \Rightarrow \quad x_1^* = 15.874 \text{ meters}$$

Thus, optimum height = 15.874m; optimum radius = 7.937m.

RELATIONSHIP BETWEEN \mathbf{x}^* AND δ^*

Again, the "weights" of the objective function terms and the terms are related as:

$$c_i \prod_{j=1}^m (x_j^*)^{a_{ij}} = \delta_i^* g_0(\mathbf{x}^*), \quad \text{for all } i \in [0]$$

term i in the O.F

For $k > 0$, i.e. for the constraints, the relationship is

$$c_i \prod_{j=1}^m (x_j^*)^{a_{ij}} \lambda_k^* = \delta_i^*, \quad \text{for all } i \in [k]$$

Recall that λ_k is actually a Lagrange multiplier for constraint k . Thus by

complementary slackness if constraint k is slack at the optimum, then $\lambda_k^* = 0$, and

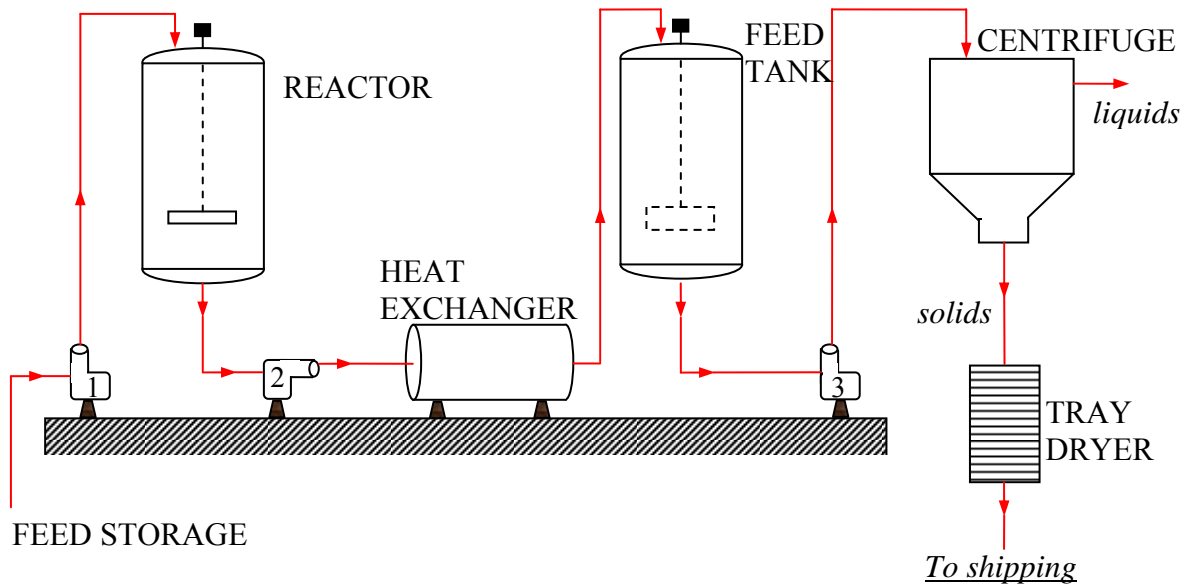
therefore $\delta_i^* = 0$ for all $i \in [k]$.

However, if constraint k is active at the optimum (i.e., $g_k(\mathbf{x}^*) = 1$), then λ_k^* could be greater than zero, **and if $\lambda_k^* > 0$** , then we may rewrite the relationship as:

$$\left\{ c_i \prod_{j=1}^m (x_j^*)^{a_{ij}} \right\} = \frac{\delta_i^*}{\lambda_k^*}, \quad \text{for all } i \in [k]$$

This allows us to recover \mathbf{x}^* from (δ^*, λ^*) .

DESIGN OF A BATCH PROCESS



Required average throughput = 50 ft /hr.

Define

V = product volume per batch (ft^3),

t_i = time required to pump batch through pump i , ($i=1,2,3$)

Combined residence time for reactor, feed tank and dryer is 10 hours.

COST COMPONENTS

REACTOR: $592V^{0.65}$

PUMP 1: $370(V/t_1)^{0.22}$

DRYER: $1200V^{0.52}$

PUMP 2: $250(V/t_2)^{0.40}$

FEED TANK: $582V^{0.39}$

PUMP 3: $250(V/t_3)^{0.40}$

HEAT EXCHANGER: $210(V/t_2)^{0.62}$

CENTRIFUGE: $200(V/t_3)^{0.85}$

MINIMIZE

$$\begin{aligned}
 & 592V^{0.65} + 582V^{0.39} + 1200V^{0.52} + 370V^{0.22}t_1^{-0.22} + \\
 & \text{(REACTOR) (FEED TANK) (DRYER) (PUMP 1)} \\
 & 250V^{0.4}t_2^{-0.4} + 210V^{0.62}t_3^{-0.4} + 250V^{0.4}t_3^{-0.4} + 200V^{0.85}t_3^{-0.85} \\
 & \text{(PUMP 2) (HEAT EXCHANGER) (PUMP 3) (CENTRIFUGE)}
 \end{aligned}$$

subject to,

$$\frac{V}{10 + t_1 + t_2 + t_3} \geq 50$$

i.e., $500V^{-1} + 50V^{-1}t_1 + 50V^{-1}t_2 + 50V^{-1}t_3 \leq 1$

$$V, t_1, t_2, t_3 > 0.$$

Degrees of difficulty = $D = n - (m+1) = 12 - (4+1) = 7$

OPTIMAL DUAL SOLUTION: (Note that $\sum_i \delta_i^* = 1$ for $i=1,8$)

Reactor: $\delta_1^* = 0.3464$ Feed Tank $\delta_2^* = 0.0610,$

Dryer: $\delta_3^* = 0.2970$ Pump 1 $\delta_4^* = 0.0204$

Pump 2: $\delta_5^* = 0.0240$ Heat Exchanger $\delta_6^* = 0.0796,$

Pump 3: $\delta_7^* = 0.0170$ Centrifuge $\delta_8^* = 0.1545,$

OPTIMAL PRIMAL SOLUTION

$$t_1^* = 0.12, \quad t_2^* = 1.46, \quad t_3^* = 3.42, \quad V^* = 740 \text{ ft}^3$$

$$g_0(\mathbf{x}^*) = v(\delta^*, \lambda^*) = \$126,303.17$$

Some Sensitivity Analysis...

Suppose that

- reactor cost coefficient increases by 2% to 604
- dryer cost coefficient increases by 1.67% to 1220
- centrifuge cost coefficient increases by 7% to 214

BY HOW MUCH WILL THE OPTIMUM COST GO UP?

Since only the cost coefficients have been altered, the current optimal solution would still be feasible (but very likely no longer optimal). Let

c_i = original cost coefficient,

C_i = new cost coefficient,

g_0^* = optimal cost before inflation (126,303.17)

G_0^* = optimal cost after inflation

δ^* = optimal dual solution before inflation

$$g_0^* = \prod_{i=1}^{12} \left(\frac{c_i}{\delta_i^*} \right)^{\delta_i^*} \lambda_1^* \lambda_1^* = \left[\prod_{i \in [0]} \left(\frac{c_i}{\delta_i^*} \right)^{\delta_i^*} \right] \left[\prod_{i \in [1]} \left(\frac{c_i}{\delta_i^*} \right)^{\delta_i^*} \lambda_1^* \lambda_1^* \right]$$

(1-8) (9-12)

$$= \left[\prod_{i \in [0]} \left(\frac{C_i}{\delta_i^*} \frac{c_i}{C_i} \right)^{\delta_i^*} \right] \left[\prod_{i \in [1]} \left(\frac{c_i}{\delta_i^*} \right)^{\delta_i^*} \lambda_1^* \lambda_1^* \right]$$

$$= \left[\prod_{i \in [0]} \left(\frac{c_i}{C_i} \right)^{\delta_i^*} \right] \underbrace{\left[\prod_{i \in [0]} \left(\frac{C_i}{\delta_i^*} \right)^{\delta_i^*} \prod_{i \in [1]} \left(\frac{c_i}{\delta_i^*} \right)^{\delta_i^*} \lambda_1^* \lambda_1^* \right]}$$

New dual obj. (with inflated costs), but evaluated at ORIGINAL SOLUTION --MUST BE $\leq G_0^*$ (the new maximum value of the dual objective).

Thus:

$$g_0^* \leq \left[\prod_{i \in [0]} \left(\frac{c_i}{C_i} \right)^{\delta_i^*} \right] \cdot G_0^* \quad \Rightarrow \quad G_0^* \geq \left[\prod_{i \in [0]} \left(\frac{C_i}{c_i} \right)^{\delta_i^*} \right] \cdot g_0^*$$

\Rightarrow New Optimal Cost is greater than or equal to

$$(126,303.17) \left(\frac{604}{592} \right)^{0.3464} \left(\frac{1220}{1200} \right)^{0.297} \left(\frac{214}{200} \right)^{0.1545} = 129,153.15$$

Thus, A LOWER BOUND on the new optimal cost is 129,153.15,

To obtain an UPPER BOUND: The current design is feasible and if retained would yield an upper bound. (We could not do any worse than the cost at the current design, which of course may be nonoptimal for the new problem with inflated costs). Thus

$$\text{New optimal cost} \leq G_0 \text{ evaluated at } \mathbf{x}^* = \underline{129,183.28} \text{ (CHECK...)}$$

Hence we can conclude that

$$\mathbf{\$129,153.169 \leq G_0^* \leq \$129,183.28}$$

An Interesting Robustness Result

Suppose one or more terms in the objective have their coefficients c_i **erroneously** estimated as ε_i . That is, we (mistakenly) solve a GP using the vector $\boldsymbol{\varepsilon}$ as opposed to the vector \boldsymbol{c} and obtain a (presumably) **suboptimal** solution - say $\boldsymbol{x}^*(\boldsymbol{\varepsilon})$ - as opposed to the true optimum solution - say $\boldsymbol{x}^*(\boldsymbol{c})$.

The question of interest:

By what amount does the value of the true objective (i.e., using \boldsymbol{c}) but computed at $\boldsymbol{x}^*(\boldsymbol{\varepsilon})$ exceed...

...the **best** value that could have been obtained if we had instead computed it at the true optimum of $\boldsymbol{x}^*(\boldsymbol{c})$?

More formally...

We are interested in the ratio $R = \frac{g_0[\boldsymbol{x}^*(\boldsymbol{\varepsilon}); \boldsymbol{c}]}{g_0[\boldsymbol{x}^*(\boldsymbol{c}); \boldsymbol{c}]}$

Clearly $R \geq 1$ and the deviation from 1 depends on the magnitude of ε_i w.r.t c_i , i.e., on the measurement error -

say $y_i = c_i / \varepsilon_i$ (*the relative error*).

Presumably R increases as the relative error increases.

Question: Is there an upper bound on R ?

Let us define

- $\delta^*(\boldsymbol{\varepsilon})$ as the dual vector corresponding to the optimum of the problem that is defined by using the (wrong) vector $\boldsymbol{\varepsilon}$.
 - $\delta^*(\boldsymbol{c})$ as the true (optimum) dual vector corresponding to the optimum of the problem that is defined by using the *correct* vector \boldsymbol{c} .
1. Note that $g_0[\mathbf{x}^*(\boldsymbol{c});\boldsymbol{c}] = v[\delta^*(\boldsymbol{c});\boldsymbol{c}]$ by the usual strong duality result for GP.
 2. Furthermore $v[\delta^*(\boldsymbol{c});\boldsymbol{c}] \geq v[\delta^*(\boldsymbol{\varepsilon});\boldsymbol{c}]$ since $\delta^*(\boldsymbol{\varepsilon})$ is feasible (but not optimal) in the original dual problem while $\delta^*(\boldsymbol{c})$ is on the other hand, optimal for this maximization problem.
 3. Therefore it follows that

$$1 \leq R = \frac{g_0[\mathbf{x}^*(\boldsymbol{\varepsilon});\boldsymbol{c}]}{g_0[\mathbf{x}^*(\boldsymbol{c});\boldsymbol{c}]} = \frac{g_0[\mathbf{x}^*(\boldsymbol{\varepsilon});\boldsymbol{c}]}{v[\delta^*(\boldsymbol{c});\boldsymbol{c}]} \leq \frac{g_0[\mathbf{x}^*(\boldsymbol{\varepsilon});\boldsymbol{c}]}{v[\delta^*(\boldsymbol{\varepsilon});\boldsymbol{c}]} \quad (1)$$

Now consider

$$g_0[\mathbf{x}^*(\boldsymbol{\varepsilon});\boldsymbol{c}] = \sum_{i \in [0]} c_i \prod_{j=1}^m (x_j^*(\boldsymbol{\varepsilon}))^{a_{ij}} = \sum_{i \in [0]} \frac{c_i}{\boldsymbol{\varepsilon}_i} \left[\boldsymbol{\varepsilon}_i \prod_{j=1}^m (x_j^*(\boldsymbol{\varepsilon}))^{a_{ij}} \right] \quad (2)$$

Note that the quantity $\left[\boldsymbol{\varepsilon}_i \prod_{j=1}^m (x_j^*(\boldsymbol{\varepsilon}))^{a_{ij}} \right]$ represents the optimum value of the i^{th} term of the objective for the program with the *wrong* coefficients ($\boldsymbol{\varepsilon}$). Writing the optimal primal dual relationships for this “wrong” formulation, we have:

$$\delta_i^*(\boldsymbol{\varepsilon}) = \frac{\boldsymbol{\varepsilon}_i \prod_{j=1}^m (x_j^*(\boldsymbol{\varepsilon}))^{a_{ij}}}{v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \boldsymbol{\varepsilon}]} \quad \text{for each } i \in [0],$$

$$\text{i.e., } \boldsymbol{\varepsilon}_i \prod_{j=1}^m (x_j^*(\boldsymbol{\varepsilon}))^{a_{ij}} = v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \boldsymbol{\varepsilon}] \times \delta_i^*(\boldsymbol{\varepsilon})$$

Substituting this value for $\left[\boldsymbol{\varepsilon}_i \prod_{j=1}^m (x_j^*(\boldsymbol{\varepsilon}))^{a_{ij}} \right]$ into (2) we get

$$g_0[\mathbf{x}^*(\boldsymbol{\varepsilon}); \mathbf{c}] = \sum_{i \in [0]} \frac{c_i}{\boldsymbol{\varepsilon}_i} \left[v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \boldsymbol{\varepsilon}] \times \delta_i^*(\boldsymbol{\varepsilon}) \right] = v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \boldsymbol{\varepsilon}] \sum_{i \in [0]} \frac{c_i}{\boldsymbol{\varepsilon}_i} (\delta_i^*(\boldsymbol{\varepsilon})) \quad (3)$$

Now, consider the true dual objective

$$\begin{aligned} v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \mathbf{c}] &= \left\{ \prod_{i=1}^n \left(\frac{c_i}{\delta_i^*(\boldsymbol{\varepsilon})} \right)^{\delta_i^*(\boldsymbol{\varepsilon})} \right\} \times \left\{ \prod_{k=1}^p \left(\sum_{i \in [k]} \delta_i^*(\boldsymbol{\varepsilon}) \right)^{\sum_{i \in [k]} \delta_i^*(\boldsymbol{\varepsilon})} \right\} \\ &= \left\{ \prod_{i=1}^n \left(\frac{c_i \boldsymbol{\varepsilon}_i}{\boldsymbol{\varepsilon}_i \delta_i^*(\boldsymbol{\varepsilon})} \right)^{\delta_i^*(\boldsymbol{\varepsilon})} \right\} \times \left\{ \prod_{k=1}^p \left(\sum_{i \in [k]} \delta_i^*(\boldsymbol{\varepsilon}) \right)^{\sum_{i \in [k]} \delta_i^*(\boldsymbol{\varepsilon})} \right\} \\ &= \left\{ \prod_{i=1}^n \left(\frac{c_i}{\boldsymbol{\varepsilon}_i} \right)^{\delta_i^*(\boldsymbol{\varepsilon})} \right\} \times \left\{ \prod_{i=1}^m \left(\frac{\boldsymbol{\varepsilon}_i}{\delta_i^*(\boldsymbol{\varepsilon})} \right)^{\delta_i^*(\boldsymbol{\varepsilon})} \right\} \times \left\{ \prod_{k=1}^p \left(\sum_{i \in [k]} \delta_i^*(\boldsymbol{\varepsilon}) \right)^{\sum_{i \in [k]} \delta_i^*(\boldsymbol{\varepsilon})} \right\} \\ &= \left\{ \prod_{i=1}^n \left(\frac{c_i}{\boldsymbol{\varepsilon}_i} \right)^{\delta_i^*(\boldsymbol{\varepsilon})} \right\} \times v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \boldsymbol{\varepsilon}] \end{aligned} \quad (4)$$

From (4) and (3)

$$R \leq \frac{g_0[\mathbf{x}^*(\boldsymbol{\varepsilon}); \mathbf{c}]}{v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \mathbf{c}]} = \frac{\sum_{i \in [0]} \frac{c_i}{\varepsilon_i} (\delta_i^*(\boldsymbol{\varepsilon}))}{\prod_{i=1}^n \left(\frac{c_i}{\varepsilon_i} \right)^{\delta_i^*(\boldsymbol{\varepsilon})}} \quad (5)$$

Since we restrict ourselves to changes in coefficients of objective terms only, $c_i = \varepsilon_i$ for all $i \notin [0]$, so that (5) reduces to

$$R \leq \frac{g_0[\mathbf{x}^*(\boldsymbol{\varepsilon}); \mathbf{c}]}{v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \mathbf{c}]} = \frac{\sum_{i \in [0]} \frac{c_i}{\varepsilon_i} (\delta_i^*(\boldsymbol{\varepsilon}))}{\prod_{i \in [0]} \left(\frac{c_i}{\varepsilon_i} \right)^{\delta_i^*(\boldsymbol{\varepsilon})}} \quad (6)$$

Now, recalling that we defined $y_i = c_i / \varepsilon_i$ it follows from (1) and (6) that

$$1 \leq R = \frac{g_0[\mathbf{x}^*(\boldsymbol{\varepsilon}); \mathbf{c}]}{g_0[\mathbf{x}^*(\mathbf{c}); \mathbf{c}]} \leq \frac{g_0[\mathbf{x}^*(\boldsymbol{\varepsilon}); \mathbf{c}]}{v[\boldsymbol{\delta}^*(\boldsymbol{\varepsilon}); \mathbf{c}]} = \frac{\sum_{i \in [0]} y_i (\delta_i^*(\boldsymbol{\varepsilon}))}{\prod_{i \in [0]} (y_i)^{\delta_i^*(\boldsymbol{\varepsilon})}} \quad \text{i.e.,}$$

$$\boxed{R \leq \frac{\sum_{i \in [0]} y_i (\delta_i^*(\boldsymbol{\varepsilon}))}{\prod_{i \in [0]} (y_i)^{\delta_i^*(\boldsymbol{\varepsilon})}}} \quad (7)$$

Note that the RHS that bounds R from above is also a posynomial in $y_i = c_i / \varepsilon_i$ given by

$$\sum_{i \in [0]} \delta_i^*(\varepsilon) \left(\prod_{\substack{j \in [0] \\ j \neq i}} (y_j)^{-\delta_j^*(\varepsilon)} \right) (y_i)^{(1-\delta_i^*(\varepsilon))}$$

Therefore, if we wish to compute the maximum value for this bound on R over a range of values for $y_i = (c_i/\varepsilon_i)$, we solve

PROGRAM X

$$\text{Maximize } f(\mathbf{y}) = \sum_{i \in [0]} \delta_i^*(\varepsilon) \left(\prod_{\substack{j \in [0] \\ j \neq i}} (y_j)^{-\delta_j^*(\varepsilon)} \right) (y_i)^{(1-\delta_j^*(\varepsilon))} \quad (8)$$

$$\text{st } \frac{1}{(1+\rho)} \leq y_i \leq (1+\rho); \quad i \in [0] \text{ for some } \rho > 0 \quad (9)$$

This problem is relatively easy to solve for any **given** GP problem - the solution is at an extreme point of the convex (polyhedral) feasible region determined by (9); thus each y_i^* is either $(1+\rho)$ or $1/(1+\rho)$.

However, without even solving explicitly a quick approximate bound on the Maximum value of $f(\mathbf{y})$ may be obtained elegantly:

Partition $[0]$ into the sets $[0]_1$ and $[0]_2$, where we define

$$[0]_1 = \{i \in [0] \mid y_i^* = (1+\rho)\} \quad \text{and} \quad [0]_2 = \{i \in [0] \mid y_i^* = 1/(1+\rho)\}$$

Obviously, $[0]_1 \cup [0]_2 = [0]$ and $[0]_1 \cap [0]_2 = \emptyset$

Going back to the bound on R given in (7) we have

$$R \leq \frac{\sum_{i \in [0]} y_i (\delta_i^*(\boldsymbol{\varepsilon}))}{\prod_{i \in [0]} (y_i)^{\delta_i^*(\boldsymbol{\varepsilon})}} = \frac{\sum_{i \in [0]_1} (1+\rho)(\delta_i^*(\boldsymbol{\varepsilon})) + \sum_{i \in [0]_2} \frac{1}{(1+\rho)}(\delta_i^*(\boldsymbol{\varepsilon}))}{\prod_{i \in [0]_1} (1+\rho)^{\delta_i^*(\boldsymbol{\varepsilon})} \prod_{i \in [0]_2} \left(\frac{1}{(1+\rho)}\right)^{\delta_i^*(\boldsymbol{\varepsilon})}}$$

Let us now define

$$\boldsymbol{\delta}^1 = \sum_{i \in [0]_1} \delta_i^*(\boldsymbol{\varepsilon}), \quad \text{and} \quad \boldsymbol{\delta}^2 = 1 - \boldsymbol{\delta}^1 = \sum_{i \in [0]_2} \delta_i^*(\boldsymbol{\varepsilon})$$

Then it follows that $R \leq \frac{(1+\rho)\boldsymbol{\delta}^1 + \frac{1}{(1+\rho)}(1-\boldsymbol{\delta}^1)}{(1+\rho)^{\boldsymbol{\delta}^1} \left(\frac{1}{(1+\rho)}\right)^{1-\boldsymbol{\delta}^1}}$, i.e.,

$$R \leq \frac{(1+\rho)\boldsymbol{\delta}^1 + \frac{1}{(1+\rho)}(1-\boldsymbol{\delta}^1)}{(1+\rho)^{2\boldsymbol{\delta}^1-1}}, \text{ i.e.,}$$

$$\boxed{R \leq \boldsymbol{\delta}^1 \left(\frac{1}{(1+\rho)^{(2\boldsymbol{\delta}^1-2)}} \right) + (1-\boldsymbol{\delta}^1) \left(\frac{1}{(1+\rho)^{2\boldsymbol{\delta}^1}} \right) = M(\rho)} \quad (10)$$

Note that the value of $M(\rho)$ depends on the optimal values of the dual variables

$\delta_i^*(\boldsymbol{\varepsilon})$ obtained by solving the wrongly formulated GP.

Since $\delta^l \in [0,1]$ an upper bound for $M(\rho)$ for a given ρ can be obtained by maximizing $M(\rho)$ over $\delta^l \in [0,1]$. It is easy to show that $M(\rho)$ is concave in δ^l with values of 1 at $\delta^l=1$ and $\delta^l=0$, and that its maximum value occurs when δ^l is equal to

$$\delta_{\max}^1 = \frac{1}{2\ln(1+\rho)} - \frac{1}{\rho^2 + 2\rho}$$

Substituting the above into the expression for $M(\rho)$ yields a bound for R

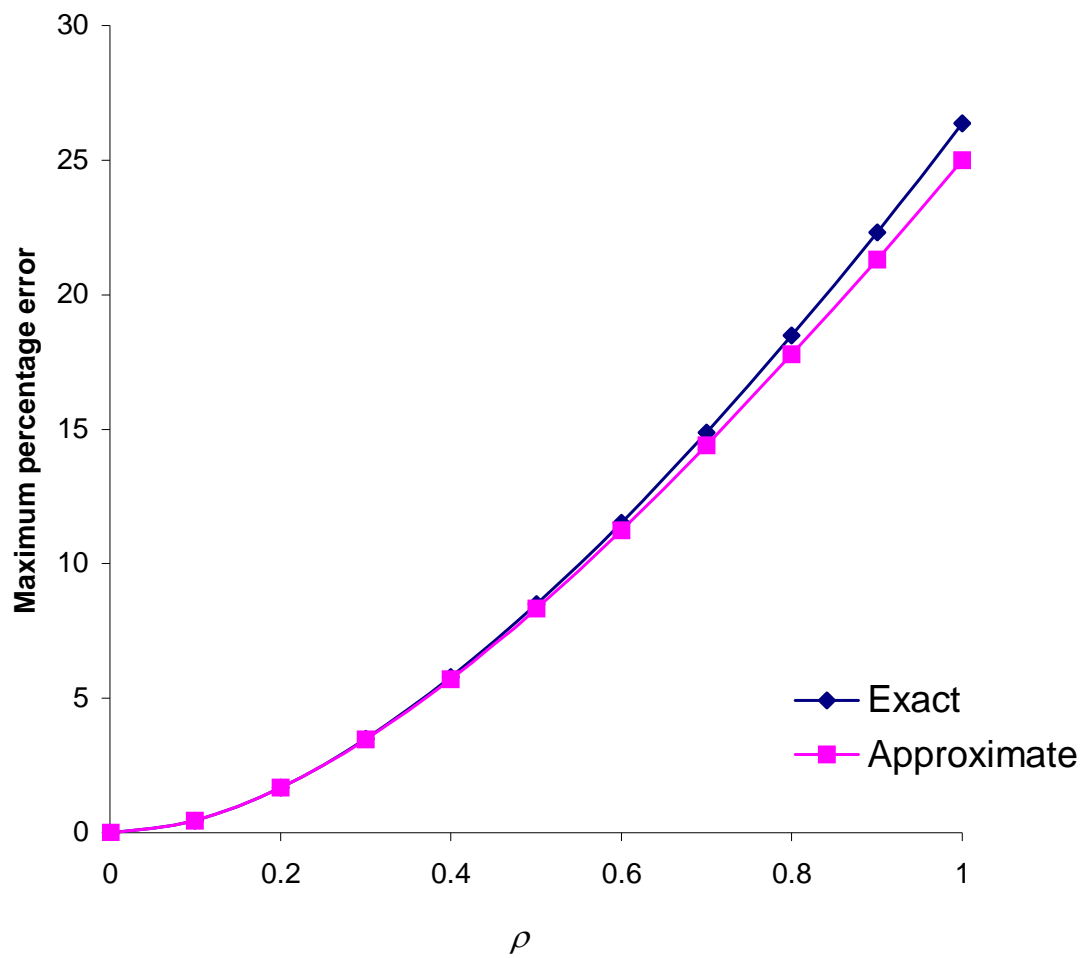
An even more quick and dirty alternative is to not bother computing $M(\rho)$ exactly at the above value of δ_{\max}^1 but by simply approximating it as $\delta_{\max}^1 \cong 0.5$ (since

$\lim_{\rho \rightarrow 0} (\delta_{\max}^1) = 0.5$ when ρ is small). In this case,

$$M(\rho) \cong 0.5 \left(\frac{1}{(1+\rho)^{-1}} \right) + 0.5 \left(\frac{1}{(1+\rho)} \right) = 0.5 \left(\frac{1}{(1+\rho)} + (1+\rho) \right) = \left(1 + \frac{\rho^2}{2(1+\rho)} \right)$$

ERROR TABLE

ρ	$(1+\rho)$	δ_{\max}^I	M (exact)	M (approx.)
0.001	1.001	0.499833417	1.00000050	1.00000050
0.01	1.01	0.498341622	1.000049505	1.000049505
0.1	1.1	0.484124582	1.004550045	1.004545455
0.2	1.2	0.469680201	1.016728349	1.016666667
0.5	1.5	0.433151731	1.084870906	1.083333333
1.0	2.0	0.388014187	1.263740721	1.25



THE BOTTOM LINE...

Proposition 1: Consider a feasible, canonical posynomial geometric program. Suppose that the exact value of the coefficient for each term i in the objective function is not known, but can be bounded to lie in the range $[c_i/(1+\rho), c_i(1+\rho)]$ where $\rho > 0$. Then the optimum value that is obtained by solving the program using ε_i as estimates for these coefficients is guaranteed to be in error by no more than

$$100 * \left(\frac{(1+\rho)^2 \delta - (1+\rho)^{2\delta} + (1-\delta)}{(1+\rho)^{2\delta}} \right) \%,$$

where

$$\delta = \frac{1}{2 \ln(1+\rho)} - \frac{1}{\rho^2 + 2\rho}.$$

Proposition 2: An approximation for the error bound in Proposition 1 is

$$100 * \left(\frac{\rho^2}{2(1+\rho)} \right) \%.$$

A COLUMN GENERATION ALGORITHM FOR THE GP DUAL

Consider the GP dual objective:

$$\text{Maximize } v(\boldsymbol{\lambda}, \boldsymbol{\delta}) = \left(\prod_{i=1}^n \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \right) \left(\prod_{k=0}^p \lambda_k^{\lambda_k} \right)$$

Since the log function is monotone increasing we can use the log-dual objective and write the dual problem as

$$\text{Maximize } \ln [v(\boldsymbol{\lambda}, \boldsymbol{\delta})] = \sum_{i=1}^n (\delta_i \ln c_i - \delta_i \ln \delta_i) + \sum_{k=0}^p (\lambda_k \ln \lambda_k)$$

st

$$\sum_{i \in [k]} \delta_i = \lambda_k, \quad k=0,1,\dots,p, \quad (1)$$

$$\lambda_0 = 1, \quad (\textit{normality}) \quad (2)$$

$$\sum_{i=1}^n \delta_i a_{ij} = 0, \quad j=0,1,\dots,m, \quad (\textit{orthogonality}) \quad (3)$$

$$\delta_i, \lambda_k \geq 0 \text{ for all defined } i \text{ and } k.$$

Advantages of this (in addition to the linear/polyhedral constraint structure):

1. The above objective is separable in the δ_i and the λ_k – can exploit this fact in developing an algorithm.
2. On the other hand, if we eliminate the λ_k by substituting from (1) and (2) into the objective we lose the separability, but we now get a concave function of the δ_i to be maximized: again a very desirable property for an algorithm.

Unfortunately...

There are also certain disadvantages with dual-based methods. These arise primarily when one or more primal constraints are slack at the optimum. Recall that λ_k is equivalent to the Lagrange multiplier for constraint k , so by complementary slackness if primal constraint k is inactive at the optimum, then $\lambda_k = 0$ at the dual optimum. Then (1) implies that each $\delta_i, i \in [k]$ also = 0. This causes difficulties.

- First, terms such as $(c_i/\delta_i)^{\delta_i}$ and $\lambda_k^{\lambda_k}$ (or $\delta_i \ln \delta_i$ and $\lambda_k \ln \lambda_k$ if the log of the dual objective is being used) are no longer defined. In the limit these quantities approach 1 (or 0 with the log-dual objective) and may therefore be explicitly defined as being equal to 1 (or 0) when δ_i and λ_k are zero.

However, in a computer code special provisions need to be made by **fixing** these at a value of 1 (or 0) when the values of δ_i and λ_k get "close" to 0.

- Second, the dual objective function is also nondifferentiable now, thus causing problems for any algorithm that requires computation of gradients.
- Third, there may be problems recovering an optimum primal vector from the dual optimum; recall that at the optimum

$$c_i \prod_{j=1}^m x_j^{a_{ij}} = \delta_i v(\boldsymbol{\lambda}, \boldsymbol{\delta}), \quad i \in [0], \quad (4)$$

$$c_i \prod_{j=1}^m x_j^{a_{ij}} = \delta_i / \lambda_k, \quad i \in [k] \quad \text{for } k \geq 1 \ni \lambda_k > 0. \quad (5)$$

In the system given by (4) and (5), there are

- m unknowns (the vector \mathbf{x}), and
- one equation corresponding to each term in the objective, as well as to each term that lies in a *constraint posynomial for which the corresponding value of λ_k is positive at the dual optimum.*

We could solve this by taking logarithms to obtain a linear system in $w_j = \ln x_j$

$$\sum_{j=1}^m a_{ij} w_j = \begin{cases} \ln[\delta_i v(\boldsymbol{\lambda}, \boldsymbol{\delta}) / c_i], & i \in [0] \\ \ln[\delta_i / (\lambda_k c_i)], & i \in [k] \ni k \geq 1, \lambda_k > 0 \end{cases} \quad (6)$$

However, if $\lambda_k = 0$ at the dual optimum, for terms in constraint k , equations (6) are not defined and it may happen that the number of equations in (6) is insufficient to provide a system with full rank, so that directly solving the above system for the w_j becomes impossible.

When the situation described above occurs, one must resort to solving what are referred to as “subsidiary problems” in order to recover the primal vector \mathbf{x}^* . In general, it may actually be necessary to solve a *sequence* of several subsidiary problems in order to obtain a primal solution. Each subsidiary problem is a separate GP problem and has the role of activating one constraint, so that dual variables corresponding to terms in that constraint are no longer zero and thus additional information becomes available in the form of (5) above for recovering

the values of the primal variables. It was shown by Duffin et al. in their original work that as constraints are activated one by one and additional information becomes available, eventually one will have sufficient information to recover a complete primal solution.

Now, suppose we define $\rho_i \geq 0$ such that

$$1. \quad \rho_i \lambda_k = \delta_i, \quad i \in [k]$$

$$2. \quad \sum_{i \in [k]} \rho_i = 1$$

(Note: **IF** $\lambda_k > 0$, then $\rho_i = \frac{\delta_i}{\lambda_k}$. But, if $\lambda_k = 0$, then any $\rho_i \geq 0 \ni \sum_{i \in [k]} \rho_i = 1$ is OK!)

Assume for now that $\lambda_k > 0$. Upon substituting $\delta_i = \rho_i \lambda_k$ into the dual program the constraints reduce to:

$$\sum_{i \in [k]} \rho_i \lambda_k = \lambda_k, \quad \text{i.e.,}$$

$$\sum_{i \in [k]} \rho_i = 1 \quad k=0,1,\dots,p,$$

$$\lambda_0 = 1, \quad (\text{normality})$$

$$\sum_{i=1}^n \rho_i \lambda_k a_{ij} = 0, \quad \text{i.e.,}$$

$$\sum_{k=0}^p \lambda_k \sum_{i \in [k]} \rho_i a_{ij} = 0, \quad j=0,1,\dots,m, \quad (\text{orthogonality})$$

$$\rho_i, \lambda_k \geq 0 \text{ for all defined } i \text{ and } k.$$

and the objective becomes:

$$\begin{aligned}
\text{Maximize } \ln [v(\boldsymbol{\lambda}, \boldsymbol{\rho})] &= \sum_{i=1}^n (\rho_i \lambda_k \ln c_i - \rho_i \lambda_k \ln \rho_i \lambda_k) + \sum_{k=0}^p (\lambda_k \ln \lambda_k) \\
&= \sum_{k=0}^p \left(\lambda_k \sum_{i \in [k]} (\rho_i \ln c_i - \rho_i \ln \rho_i - \rho_i \ln \lambda_k) \right) + \sum_{k=0}^p (\lambda_k \ln \lambda_k) \\
&= \sum_{k=0}^p \lambda_k \left(\ln \lambda_k + \sum_{i \in [k]} (\rho_i \ln c_i - \rho_i \ln \rho_i - \rho_i \ln \lambda_k) \right) \\
&= \sum_{k=0}^p \lambda_k \left(\ln \lambda_k - \ln \lambda_k \sum_{i \in [k]} \rho_i + \sum_{i \in [k]} (\rho_i \ln c_i - \rho_i \ln \rho_i) \right) \\
&= \sum_{k=0}^p \lambda_k \left(\sum_{i \in [k]} \left(\rho_i \ln \frac{c_i}{\rho_i} \right) \right)
\end{aligned}$$

We may now state the GP Dual problem as a **Generalized Linear Program (GLP)** in the λ_k variables. A GLP is a linear program where the column of coefficients for each variable is drawn from some (convex) set. To see this we define first

$$\mathcal{Q}_k = \{ \boldsymbol{\rho} \in \mathbb{R}^n \mid \rho_i = 0 \forall i \notin [k], \sum_{i=1}^n \rho_i = \sum_{i \in [k]} \rho_i = 1, \rho_i \geq 0 \}, \quad (7)$$

Next the convex functions A_{kj} and G_k are defined:

$$A_{kj}(\boldsymbol{\rho}) = \sum_{i=1}^n a_{ij} \rho_i, \boldsymbol{\rho} \in \mathcal{Q}_k = \sum_{i \in [k]} a_{ij} \rho_i, \boldsymbol{\rho} \in \mathcal{Q}_k, \quad (8)$$

$$G_k(\boldsymbol{\rho}) = \sum_{i=0}^n \rho_i \ln(c_i / \rho_i), \boldsymbol{\rho} \in \mathcal{Q}_k = \sum_{i \in [k]} \rho_i \ln(c_i / \rho_i), \boldsymbol{\rho} \in \mathcal{Q}_k. \quad (9)$$

The GLP form of the dual is then stated as:

Program D Find a set of vectors $\rho^0, \rho^1, \dots, \rho^p$ from the sets Q_0, Q_1, \dots, Q_p

respectively, and the nonnegative vector $\lambda = [\lambda_0, \lambda_1, \dots, \lambda_p]^T$ so as to

$$\text{Maximize } V(\lambda, \rho) = \sum_{k=0}^p G_k(\rho^k) \lambda_k \quad (10)$$

$$\sum_{k=0}^p \lambda_k A_{kj}(\rho^k) = 0, \quad j = 1, 2, \dots, m \quad (11)$$

$$\lambda_0 = 1 \quad (12)$$

$$\lambda_k \geq 0, k = 0, 1, \dots, p$$

Note that this is actually a nonlinear program in the ρ^k and λ . It is a generalized linear program because for a fixed choice of values for each ρ^k the problem is a linear program in λ . The set of column vectors that can be generated from any Q_k is the set S_k given by

$$S_k = \{(\gamma, \alpha) \mid \alpha \in \mathcal{R}^m, \gamma \in \mathcal{R}, \alpha_j = A_{kj}(\rho^k), \gamma \leq G_k(\rho^k), \rho^k \in Q_k\}, \quad k = 0, 1, \dots, p \quad (13)$$

The above set is easily shown to be a convex one, and as a consequence of the maximization of $V(\lambda, \rho)$, one only needs to choose from among the boundary points of this convex set in order to generate the candidate columns for Program D, i.e., columns having $\gamma = G_k(\rho^k)$.

Note that this dual has infinitely many columns (each generated by one ρ vector). The primal corresponding to this dual thus has infinitely many constraints (also called a Semi-Infinite LP):

$$\text{Program P} \quad \text{Minimize } w_0 \quad (14)$$

$$\text{st} \quad \sum_{j=1}^m w_j A_{0j}(\rho^0) + w_0 \geq G_0(\rho^0) \quad \forall \rho^0 \in Q_0 \quad (15)$$

$$\sum_{j=1}^m w_j A_{kj}(\rho^k) \geq G_k(\rho^k) \quad \forall \rho^k \in Q_k, \quad k = 1, 2, \dots, p \quad (16)$$

Theorem 1: Suppose the original GP primal is consistent and attains its optimum at \mathbf{x}^* . Then the vector \mathbf{w}^* with $w_0^* = \ln g_o(\mathbf{x}^*)$, and $w_j^* = -\ln x_j^*$ for $j=1, 2, \dots, m$ is optimal for Program P. Conversely if the vector \mathbf{w}^* is optimal for Program P, then the vector \mathbf{x}^* with $x_j^* = \exp(-w_j^*)$ for $j=1, 2, \dots, m$ is optimal for the GP primal and the value of the optimal primal objective is given by $g_o(\mathbf{x}^*) = \exp(w_0^*)$.

Next, as a matter of convenience we make the following definition

Definition 1: Define $\underline{\rho}$ as the vector in \mathcal{R}^n whose i^{th} element is obtained via $\underline{\rho}_i = \{\rho_i^k, i \in [k]\}$, i.e., for each $k=0,1,\dots,p$ the elements of ρ^k whose indices correspond to the index subset $[k]$ are extracted (all other elements being zero by definition), and then combined to form a single vector in \mathcal{R}^n .

OBSERVATION: If a primal constraint k is slack at the optimum, then λ_k is zero so that $G_k(\rho^k)$ can take on any value as far as the objective is concerned; this implies that any vector ρ^k in Q_k would be “optimal,” so that the GLP dual has infinitely many optimal solutions! However, the need is for a specific $\underline{\rho}$ that will allow us to recover a primal solution; we refer to such a $\underline{\rho}$ as a **superoptimal** vector.

Theorem 2: Suppose Program P attains its optimum at w^* . Then Program D is also consistent and attains its optimum at a point $(\lambda^*, \underline{\rho}^*)$ with $V(\lambda^*, \underline{\rho}^*) = w_0^*$.

Moreover, the strictly positive vector $\underline{\rho}^*$ with

$$\underline{\rho}_i^* = \frac{c_i \exp\left(\sum_{j=1}^m -w_j^* a_{ij}\right)}{\sum_{s \in [k]} c_{sj} \exp\left(\sum_{j=1}^m -w_j^* a_{sj}\right)}, i \in [k] \quad (17)$$

is said to be superoptimal.

Note that if vector \mathbf{x}^* with $x_j^* = \exp(-w_j^*)$ for all j is optimal in Program A, one may use Theorem 1 to rewrite (17) and relate the optimal primal vector \mathbf{x}^* to the superoptimal GLP dual vector $\underline{\rho}^*$ as

$$\underline{\rho}_i^* = \frac{c_i \prod_{j=1}^m x_j^{*a_{ij}}}{\sum_{q \in [k]} c_{sj} \prod_{j=1}^m x_j^{*a_{sj}}} = \frac{c_i \prod_{j=1}^m x_j^{*a_{ij}}}{g_k(\mathbf{x}^*)}, \quad i \in [k] \quad (18)$$

COMPUTATIONAL IMPLICATIONS

- At the optimum, the variables in the conventional GP dual (represented by δ) had a similar set of relationships with the primal solution (Equations 4 and 5). However, (18), which relates the primal solution to the superoptimal dual solution $\underline{\rho}^*$ is different in one very important respect: the vector $\underline{\rho}^*$ is **strictly positive** and is defined for all terms in the problem, regardless of the posynomial from which the term comes and regardless of whether the constraint defined by the posynomial is active or inactive.
- Also note that if $g_k(\mathbf{x}^*)=1$ this is just the usual primal-dual relationship!
- Nondifferentiability is no longer an issue, since the elements of the vector $\underline{\rho}$ are strictly positive.

- Since (18) is defined for each of the n terms in the problem there is always a system of n equations for recovering an optimal primal solution. So subsidiary problems don't arise.
- In practice, (18) need never be explicitly used since the simplex multipliers corresponding to successive LP approximations to Program D that are solved at each iteration of a column-generation algorithm may be shown to converge to the vector $-w$. Thus the elements of this vector are merely exponentiated at the final approximation to obtain the optimum primal solution to the original problem (via Theorem 1).

SOLUTION APPROACH

The problem may be viewed as a two-stage one.

1. Given $\rho^k \in Q_k$, $k=0,1,2,\dots,p$, a total of $(p+1)$ columns may be generated based upon (13) and the resulting LP solved for λ ; this may be viewed as the first or "inner" level.
2. In the second or "outer" level, one needs to find, among all possible $\rho^k \in Q_k$, $k=0,1,2,\dots,p$, the specific vectors for which the optimal value of the corresponding LP is *maximized*.

Now, suppose that it is possible to generate a column for each of the (infinitely many) $\rho^k \in Q_k$, $k=0,1,\dots,p$. The procedure could then be combined into a single step by solving the resulting (giant!) LP with infinitely many columns and allowing the simplex method to enter into the basis only that column for each k which is optimal.

This is of course impossible as a practical matter; but the fact that only one column is required corresponding to each k immediately suggests a column generation procedure:

- Start with an initial set of columns,
- Generate additional columns at each iteration to improve the value of the LP objective.

Note that any LP with some subset of the infinitely many columns is an approximate representation of the original (complete) problem D.

A COLUMN GENERATION ALGORITHM

STEP 0 (Initialization): Define a suitable tolerance $\varepsilon > 0$. Construct an initial LP approximation to the GLP dual (Program D) by adding an initial set of columns. Solve this LP approximation and obtain the simplex multiplier vector $\mathbf{w} = (w_1, w_2, \dots, w_m, w_0)$.

STEP 1 (Check for Convergence): Obtain vector \mathbf{x} with $x_j = \exp(-w_j)$.

Define $L = \{k \mid g_k(\mathbf{x}) \geq (1 + \varepsilon) \text{ for } k=1, 2, \dots, p \text{ and } g_k(\mathbf{x}) \geq \exp(w_k) + \varepsilon, \text{ for } k=0\}$. If $L = \emptyset$ then the algorithm has converged and the optimum solution is given by \mathbf{x} ; STOP.

STEP 2 (Iterative Step): For each $k \in L$, evaluate the vector $\boldsymbol{\rho}^k$ whose i^{th} element is

$$\rho_i^k = c_i \exp\left(\sum_{j=1}^m -w_j a_{ij}\right) \bigg/ \sum_{s \in [k]} c_s \exp\left(\sum_{j=1}^m -w_j a_{sj}\right).$$

Then generate a column $[G_k(\boldsymbol{\rho}^k), A_{k1}(\boldsymbol{\rho}^k), \dots, A_{km}(\boldsymbol{\rho}^k), \sigma]^T$ with

$$G_k(\boldsymbol{\rho}^k) = \sum_{i \in [k]} \rho_i^k \ln(c_i / \rho_i^k) \quad \text{and} \quad A_{kj}(\boldsymbol{\rho}^k) = \sum_{i \in [k]} a_{ij} \rho_i^k \quad \text{for } j=1, 2, \dots, m; \text{ and } \sigma = 1 \text{ if}$$

$k=0$ and 0 otherwise. Solve the resulting LP approximation after adding

column(s) and obtain the new simplex multiplier vector $\mathbf{w} = (w_1, w_2, \dots, w_m, w_0)$.

Return to Step 1.

NOTE: Step 0 is easily accomplished so that one need not specify a starting point....

ITERATIVE STEP AND CONVERGENCE CHECK

Suppose the current LP approximation to Program D has been solved and the corresponding set of simplex multipliers are (w_1, \dots, w_m) and w_0 corresponding to (11) and (12) respectively. To price out a particular set k , consider the reduced cost of variable λ_k . From (10), (11) and (12) it may be seen that this is given by

$$\bar{G}_k = \begin{cases} \sum_{j=1}^m w_j A_{kj}(\rho^k) - G_k(\rho^k) & \text{for } k > 0 \\ w_0 + \sum_{j=1}^m w_j A_{kj}(\rho^k) - G_k(\rho^k) & \text{for } k = 0 \end{cases} \quad (19)$$

- If there exists $\rho^k \in Q_k$ such that \bar{G}_k is negative then the corresponding column may be introduced into the basis to improve the objective.
- Equivalently, there is no attractive column from Q_k if $\bar{G}_k \geq 0$ for all $\rho^k \in Q_k$.
- Thus for each k , a subproblem of minimizing \bar{G}_k over all $\rho^k \in Q_k$ is solved and the optimum value checked to see if it is negative. If so, the corresponding value of ρ^k is used to generate an additional column for the LP, otherwise there is no beneficial column for k .
- This subproblem has a closed form solution; the i^{th} element of the minimizing vector ρ^k is given by

$$\rho_i^k = c_i \exp\left(\sum_{j=1}^m -w_j a_{ij}\right) \Bigg/ \sum_{s \in [k]} c_s \exp\left(\sum_{j=1}^m -w_j a_{sj}\right) \quad (20)$$

Substituting the value of ρ^k from (20) into the expressions for $G_k(\rho^k)$ and $A_{kj}(\rho^k)$ given by (9) and (8), and the resulting values into the expression for the reduced cost \bar{G}_k given by (19) yields the maximum value of the latter.

Some (much!) algebraic simplification yields these maximum values for the reduced costs as:

$$-\ln\left(\sum_{i \in [k]} c_i \exp\left(\sum_{j=1}^m -w_j a_{ij}\right)\right) \text{ for } k \geq 1 \text{ and}$$

$$w_0 - \ln\left(\sum_{i \in [0]} c_i \exp\left(\sum_{j=1}^m -w_j a_{ij}\right)\right) \text{ for } k=0.$$

Thus, the current solution is optimal if

$$-\ln\left(\sum_{i \in [k]} c_i \exp\left(\sum_{j=1}^m -w_j a_{ij}\right)\right) \geq 0, \text{ i.e.,}$$

$$\left(\sum_{i \in [k]} c_i \exp\left(\sum_{j=1}^m -w_j a_{ij}\right)\right) \leq 1, \quad k \geq 1 \tag{21}$$

$$w_0 - \ln\left(\sum_{i \in [0]} c_i \exp\left(\sum_{j=1}^m -w_j a_{ij}\right)\right) \geq 0, \text{ i.e.,}$$

$$\left(\sum_{i \in [0]} c_i \exp\left(\sum_{j=1}^m -w_j a_{ij}\right)\right) \leq \exp(w_0). \tag{22}$$

In order to see an elegant relationship between the preceding statement and the original primal, let us define

$$x_j = \exp(-w_j), \quad j=1,2,\dots,m. \quad (23)$$

Substituting from (23) into (21) & (22) yields the following optimality condition:

$$\sum_{i \in [k]} c_i \prod_{j=1}^m x_j^{a_{ij}} = g_k(\mathbf{x}) \leq 1, \quad k=1,\dots,p \quad (24)$$

$$\sum_{i \in [0]} c_i \prod_{j=1}^m x_j^{a_{ij}} = g_0(\mathbf{x}) \leq \exp(w_0) \quad (25)$$

Thus at each iteration of the algorithm

- The simplex multipliers corresponding to (11) are negated and then exponentiated to obtain estimates of the primal solution vector via (23).
- Then (24) is used to check whether this vector is feasible in the original GP primal, and (25) is used to check for any duality gap.
- If (24) and (25) are satisfied then the vector \mathbf{x} must be optimal for the original problem and the algorithm stops.
- Otherwise, corresponding to each k for which (24) is violated (and for $k=0$ if (25) is violated), a vector ρ^k is computed via (20) and used to generate a column which is then added on to the current LP approximation. This is solved to obtain a new simplex multiplier vector and the process continues.

SIGNOMIAL GEOMETRIC PROGRAMMING

This is similar to Posynomial GP, except that the term coefficients (c_i) are *no longer restricted to be positive*.

PRIMAL

$$\text{Minimize } g_0(\mathbf{x}) = \sum_{i \in [0]} \sigma_i c_i \prod_{j=1}^m x_j^{a_{ij}}$$

$$\text{st } g_k(\mathbf{x}) = \sum_{i \in [k]} \sigma_i c_i \prod_{j=1}^m x_j^{a_{ij}} \leq \sigma_{k0}, \quad k = 1, 2, \dots, p$$

$$x_j > 0, \quad j = 1, 2, \dots, m,$$

$$c_i > 0, \sigma_i, \sigma_{k0} = +1 \text{ or } -1 \text{ (given)}$$

DUAL

$$\text{Maximize } v(\lambda, \delta) = \sigma_{00} \left(\prod_{i=1}^n \left(\frac{c_i}{\delta_i} \right)^{\delta_i \sigma_i} \prod_{k=1}^p \lambda_k^{\lambda_k \sigma_{k0}} \right)^{\sigma_{00}}$$

$$\text{st } \sigma_{k0} \sum_{i \in [k]} \sigma_i \delta_i = \lambda_k, \quad k = 1, \dots, p,$$

$$\sum_{i=1}^n a_{ij} \delta_i \sigma_i = 0, \quad j = 1, 2, \dots, m, \quad \text{(Generalized Orthogonality)}$$

$$\sigma_{00} \sum_{i \in [0]} \sigma_i \delta_i = 1, \quad \text{(Generalized Normality)}$$

$$\delta_i, \lambda_k \geq 0 \text{ for all } i, k.$$

(σ_{00} is the sign of the primal objective at the optimum)

In posynomial GP there was NO DUALITY GAP (primal optimum value = dual optimum value). This was so because the primal posynomial GP is a CONVEX PROGRAM (this is easy to show by rewriting it using the change in variables $t_j = \ln x_j$).

On the other hand the signomial GP primal is NOT a convex program and is likely to have local minima. In general there WILL be a duality gap, and most importantly, a feasible primal vector will not necessarily lead to a primal value that is greater than the dual value for a feasible dual vector. This unfortunately precludes the use of dual solutions for quick estimates or bounds.

For every \mathcal{D} that yields a local maximum for the dual, there is a corresponding \mathbf{x}' that is a stationary point for the primal, with the respective objective functions having the same value. Unfortunately, this \mathbf{x}' could be a local maximum or a local minimum for the primal. In practice, one must find all local maxima for the dual, then choose the smallest among these!

EXAMPLE: $\text{Max } 5x_1^2 - x_2^2 x_3^4$

 st $1.5x_2^{-1}x_3 - 2.5x_1^2x_2^{-2} \geq 1$

$x_1, x_2, x_3 > 0$

Converting to the "standard" form, we have,

 Min $x_2^2 x_3^4 - 5x_1^2$

 st $2.5x_1^2 x_2^{-2} - 1.5x_2^{-1}x_3 \leq -1$

Here $m=3$, $n=4$, $\sigma_1=+1$, $\sigma_2=-1$, $\sigma_3=+1$, $\sigma_4=-1$, $\sigma_{10}=-1$,

Assume that $\sigma_{00}=1$...

The DUAL constraints are

Normality: $\sigma_{00} \sum_{i \in [0]} \sigma_i \delta_i = 1$

\Rightarrow $\boxed{\delta_1 - \delta_2 = 1}$

Orthogonality: $\sum_{i=1}^n a_{ij} \delta_i \sigma_i = 0, \quad j = 1, 2, 3.$

\Rightarrow $\boxed{\begin{array}{rcl} -2\delta_2 + 2\delta_3 & = & 0 \\ 2\delta_1 & - & 2\delta_3 + \delta_4 = 0 \\ 4\delta_1 & & - \delta_4 = 0 \end{array}}$

Lagrange Multipliers: $\lambda_k = \sigma_{k0} \sum_{i \in [k]} \sigma_i \delta_i, \quad k = 1,$

\Rightarrow $\boxed{\lambda_1 = \delta_4 - \delta_3}$

Nonnegativity: $\delta_i \geq 0, i=1, 2, 3, 4.$

Solving the normality and orthogonality conditions yields:

$$\delta_1^* = -0.5; \delta_2^* = -1.5; \delta_3^* = -1.5; \delta_4^* = -2,$$

which obviously violates nonnegativity and is therefore infeasible in the dual.

THEREFORE, $7s_1 = 1$ IS A WRONG ASSUMPTION

Assume that $\sigma_{00} = -1$

The only constraint that is affected by this assumption is NORMALITY, which now becomes

$$\boxed{\delta_2 - \delta_1 = 1} \quad (\text{instead of } \delta_1 - \delta_2 = 1)$$

The solution to normality and duality now yields:

$$\delta_1^* = 0.5; \delta_2^* = 1.5; \delta_3^* = 1.5; \delta_4^* = 2, \lambda_1^* = 0.5$$

which is feasible.

The dual objective is

$$\begin{aligned} & -1 \left(\prod_{i=1}^4 \left(\frac{c_i}{\delta_i^*} \right)^{\delta_i^* \sigma_i} \lambda_1^* \lambda_1^* \sigma_{10} \right)^{-1} \\ & = - \left(\left(\frac{1}{0.5} \right)^{0.5} \left(\frac{5}{1.5} \right)^{-1.5} \left(\frac{2.5}{1.5} \right)^{1.5} \left(\frac{1.5}{2} \right)^{-2} (0.5)^{-0.5} \right)^{-1} \end{aligned}$$

$$= -0.795495 = g_0(\mathbf{x})$$

To recover the optimal primal variables we use the relationships:

$$c_i \prod_{j=1}^m x_j^{a_{ij}} = \sigma_{00} \delta_i g_0(\mathbf{x}), \quad i \in [0]$$

$$c_i \prod_{j=1}^m x_j^{a_{ij}} = \delta_i / \lambda_k, \quad i \in [k]$$

Thus

$$x_2^2 x_3^4 = (-0.5) \cdot (-0.795495)$$

$$5x_1^2 = (-1.5) \cdot (-0.795495) \quad \Rightarrow \quad x_1 = 0.4885$$

$$2.5x_1^2 x_2^{-2} = 1.5/0.5 = 3 \quad \Rightarrow \quad x_2 = 0.4885$$

$$1.5x_2^{-1} x_3 = 2.0/0.5 = 4 \quad \Rightarrow \quad x_3 = 1.189$$

Although this is a stationary point for the primal we cannot say anything further about it without more analysis ...