Geometric programming

From optimization

Authors: Helen Wu (ChE345 Spring 2015)
Steward: Dajun Yue and Fengqi You

Contents

Introduction

Geometric programming was introduced in 1967 by Duffin, Peterson and Zener. It is very useful in the applications of a variety of optimization problems, and falls under the general class of signomial problems[1]. It can be used to solve large scale, practical problems by quantifying them into a mathematical optimization model. Geometric programs (GP) are useful in the context of geometric design and models well approximated by power laws. Applications of GP include electrical circuit design and other topics such as finance and statistics[2].

Model Formulation

Standard Form

A geometric program is composed of an objective function that is subjected to constraints. All of the components must be in the nature of monomials and posynomials. A monomial is a single term and takes the form of

$$f(x) = Cx_1^{a_1} x_2^{a_2} ... x_n^{a_n},$$

where the coefficient C>0 and the exponents, $a_i \in \mathbb{R}$. Note that use of "monomial" is different from the meaning in algebra; here, monomials can have negative exponents. A posynomial takes the form of the sum of one or more monomials:

$$f(x) = \sum_{k=1}^{K} C_k x_1^{a_{1k}} x_2^{a_{2k}} ... x_n^{a_{nk}},$$

A geometric program in standard form looks like this:

$$\begin{array}{ll} Minimize & f_o(x) \\ Subject\ to & f_i(x) \leq 1, i = 1, ..., m, \\ & g_i(x) = 1, i = 1, ..., p, \end{array}$$

where f_i are posynomials, g_i are monomials, and x_i are optimization variables. The objective must be to minimize a posynomial. Often times the geometric program must be reformulated into standard form. If presented with a maximizing problem, the inverse can be taken to convert it into a minimizing problem[2].

Example

Consider the following example problem:

$$\begin{array}{ll} Maximize & \frac{x^2}{yz} \\ Subject\ to & 1 \leq x \leq 5, \\ & 2 \leq y \leq 4, \\ & y^2 + 2xy + 5\frac{z^2}{x} + x \leq \sqrt{y} \\ & \frac{x}{z} = y^2 \end{array}$$

with variables $x, y, z \in \mathbb{R}; x, y, z > 0$.

The equivalent standard form GP is as follows

$$\begin{array}{ll} Minimize & x^{-2}yz \\ Subject\ to & x^{-1} \leq 1 \\ & \frac{1}{5}x \leq 1 \\ & 2y^{-1} \leq 1 \\ & \frac{1}{4}y \leq 1 \\ & y^{\frac{3}{2}} + 2x^{\frac{1}{2}} + 5z^2x^{-1}y^{-\frac{1}{2}} + xy^{-\frac{1}{2}} \leq 1 \\ & xz^{-1}y^{-2} \leq 1 \end{array}$$

Solution Approaches

In order to solve a GP, there are many factors to consider. The GP must be in a specific form in order to solve, and we must determine the feasibility of the problem.

Convex Form

In order to solve a geometric program, it must be reformulated into a nonlinear, convex optimization problem via a change in variables. By applying a logarithmic transformation, GP can be seen as an extension of linear programming. Setting $y_i = \log x_i$ results in the following GP:

$$\begin{aligned} &Minimize && log\, f_o(e^y) \\ &Subject\ to && log\, f_i(e^y) \leq 0, i = 1, ..., m, \\ & && log\, g_i(e^y) = 0, i = 1, ..., p. \end{aligned}$$

By transforming the GP into this form, it can be solved more efficiently[2].

Feasibility

In order to solve the GP, the problem must be feasible. If it is not feasible, then no optimal solution will be found. In this case, at least one constraint must be relaxed. This can be done by adding a new scalar variable, *s*, to find a value $\hat{x}$ that is "close to feasible." The GP now looks like this:

$$\begin{aligned} &Minimize && s \\ &Subject\ to && f_i(x) \leq s, i = 1, ..., m, \\ & && g_i(x) = s, i = 1, ..., p, \\ & && s \geq 1 \end{aligned}$$

This problem can be solved to find the optimal values of $\bar{x}$ and $\bar{s}$. $\bar{S}$ is indicative of how feasible the original GP is. For example. If $\bar{s}$=1, then $\bar{x}$ is feasible for the original problem. If $\bar{s}$ is greater than 1, then we set $\hat{x}$ equal to $\bar{x}$.

Solvers also may use a trade-off analysis of the GP, where the constraints are varied to see how they may affect the optimal solution. This results in a "perturbed" GP, and can be modeled as:

$$\begin{aligned} &Minimize && f(x) \\ &Subject\ to && f_i(x) \leq u_i, i = 1, ..., m, \\ & && g_i(x) = v_i, i = 1, ..., p. \end{aligned}$$

Instead of having the constraints less than or equal to 1 or equal to 1, it is instead replaced with parameters *u* and *v* which are positive constants. If *u* is greater than one, then the inequality constraint is loosened; if *u* is less than 1, then the inequality constraint is tightened. Solving this perturbed model for different values of *u* and *v* allows analysis on how these values relate to the optimal solution. An optimal trade-off curve can be formed by plotting *p(u,v)* versus u_i, with all other u_i and v_j equal to one. This will display the "optimal trade-off" of the *i* th inequality constraint and objective.

Similarly, a sensitivity analysis allows the examination of how small changes in the constraints affects the optimal solution[2].

Generalized GP

In the case that the polynomials are taken to a fractional power, they can be handled by introducing a new variable and a bounding constraint. If, for example, f_1 and f_2 are posynomials taken to a fractional power, then we can introduce new variables t_1 and t_2 which represent the upper bounds of the posynomials. We can set

$$f_1(x) \leq t_1$$

$$f_2(x) \leq t_2.$$

Adding these new variables will now make the problem compatible with GP[2].

Methods

There have been many different methods that have been proposed to solve different types of GPs, and all have their own advantages and disadvantages. Some examples include:

- To solve a engineering optimization problem, Coello and Cortés created a method using a genetic algorithm with an artificial immune system. A limitation of this method includes that you can only obtain local optima solutions.
- To solve a Lipschitzian problem, Horst and Tuy created an analytical approach. A limitation of this method is that it is only feasible for problems that contain variables that can be reduced by analytical techniques.
- Sherali and Tuncbilek proposed a reformulation-linearization technique. This technique linearizes the problem by adding new variables, and creates new constraints. A limitation of this method is that it requires a long trial-and-error process and therefore can be harder to use.
- Li and Chang suggested using the approach of a using a logarithmic transformation of the problem, followed by a piecewise-linearization. This method is easy to implement, and can be used to calculate global minima. However, a limitation is that the addition of extra binary variables may cause it to become very complex.

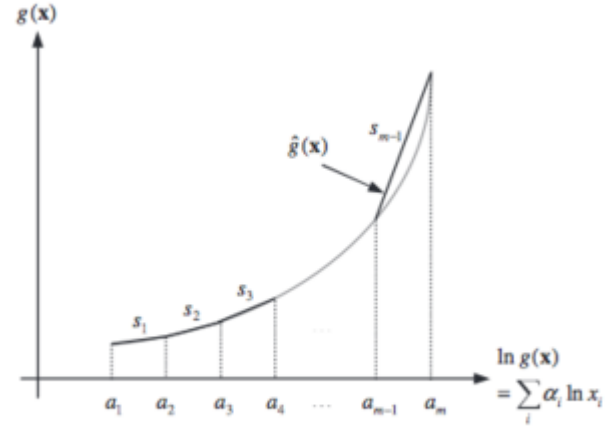

Fig. 1. A piecewise linear function of $g(\mathbf{x}) = \prod_{i=1}^{n} x_i^{\alpha_i}$

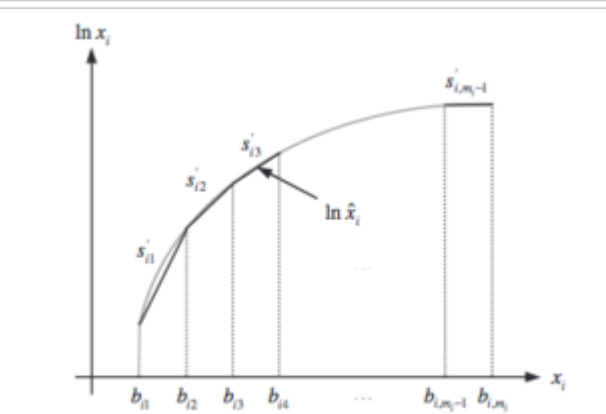

Fig. 2. A piecewise linear function of $\ln x_i$ in $g(\mathbf{x}) = \prod_{i=1}^{n} x_i^{\alpha_i}$

Huang and Kao expand on the Li and Chang's proposed method of a logarithmic piecewise-linearization, and attempt to reduce the number of binary variables. Considering the posynomial function This process includes the following steps:

1. Considering posynomial function $g(x) = \prod_{i=1}^{N} x_i^{\alpha_i}$, take the logarithm to get $ln\ g(x) = \prod_{i=1}^{N} \alpha_i ln x_i$.
2. The expression can be represented in a different way to obtain "break points" which are used to linearize the problem. Here, $g(x) = \prod_{i=1}^{N} x_i^{\alpha_i}$ can also be expressed as

$$\hat{g}(x) = e^{a_1} + s_1(\sum_{i=1}^{n} \alpha_i\ lnx_i - a_1) + \sum_{j=2}^{m-1} \frac{s_j - s_{j-1}}{2}(|\sum_{i=1}^{n} \alpha_i\ lnx_i - a_j| + \sum_{i=1}^{n} \alpha_i\ lnx_i - a_j)$$

where $a_i, a_2, ..., a_m$ are the break points, and

$$ln\hat{x}_i = lnb_{i1} + s\prime_{i1}(x_i - b_{i1}) + \sum_{j=2}^{m_i-1} \frac{s\prime_{ij} - s\prime_{ij-1}}{2}(|x_i - b_{ij}| + x_i - b_{ij})$$

where $b_{i1}, b_{i2}, ...b_{i,m_i}$ are the break points for lnx_i. Refer to Figures 1 and 2.

3. Let D represent a set of binary variables $D = \{u_1, u_2, ...u_h\}$ where $h = \lceil log_2(m-1) \rceil$. Then, the following will be true:

$$a_\theta - Mq_\theta \leq lng \leq a_{\theta+1} + Mq_\theta,$$

$$\begin{cases} e^{a_\theta} + s_\theta(\sum_{i=1}^{n} \alpha_i lnx_i - a_\theta) - M\prime q_\theta \leq \hat{g} \\ \hat{g} \leq e^{a_\theta} \sum_{i=1}^{n} \alpha_i lnx_i - a_\theta) + M\prime q_\theta, \end{cases}$$

$$q_\theta = ||G(\theta)|| - \sum_{u_j \in D, j \in G(\theta)} u_j + \sum_{u_j \in D, j \notin G(\theta)} u_j.$$

4. Rewriting the expressions in the way allows us to calculate the slopes of the piecewise-linearization function. The slope between points $(a_\theta, g(a_\theta))$ and $(a_{\theta+1}, g(a_{\theta+1}))$ can be calculated as:

$$s_\theta = \frac{g(a_{\theta+1}) - g(a_\theta)}{a_{\theta+1} - a_\theta}$$

where $\theta = 1, 2, ...m-1$.

5. A similar process is used when linearizing lnx_i. Let D_i represent a set of binary variables $D_i = \{u_{i1}, u_{i2}, ...u_{i,h_i}\}$ where $h_i = \lceil log_2(m_i - 1) \rceil$. Then, the following will be true:

$$b_{i,\theta_i} - Mq_{i,\theta_i} \leq x_i \leq b_{i,\theta_i+1} + Mq_{i,\theta_i},$$

$$\begin{cases} lnb_{i,\theta_i} + s\prime_{i,\theta_i}(x_i - b_{i,\theta_i}) - M\prime q_{i,\theta_i} \leq ln\hat{x}_i \\ ln\hat{x}_i \leq lnb_{i,\theta_i} + s\prime_{i,\theta_i}(x_i - b_{i,\theta_i}) + M\prime q_{i,\theta_i}, \end{cases}$$

$$q_{i,\theta_i} = ||G(i,\theta_i)|| - \sum_{u_{i,j} \in D_{i,j} \in G(i,\theta_i)} u_{i,j} + \sum_{u_{i,j} \in D_{i,j}, j \notin G(i,\theta_i)} u_{i,j}$$

Therefore the slope between points $(b_{i,\theta_i}$ and $lnb_{i,\theta_i})$ can be calculated as:

$$s\prime_{\theta_i} = \frac{lnb_{i,\theta_i+1} - lnb_{i,\theta_i}}{b_{i,\theta_i+1} - b_{i,\theta_i}},$$

where $\theta_i = 1, 2, ...m_i - 1$.

6. By rewriting all these expressions, the number of binary variables is reduced, and the GP may be simpler to solve[1].

Illustrative Example

Here is an example taken from a paper written by Huang and Kao regarding their method[1].

$$\begin{aligned} Minimize \quad & x_1 x_2^{0.5} x_3^{1.2} + 2x_1 \\ Subject\ to \quad & 2x_1 + x_2 + x_3 \geq 8 \\ & x_2 + 2x_3 \geq 10.5 \\ & x_1 + 2x_3 \leq 10 \\ & 1 \leq x_1, x_2, x_3 \leq 5 \end{aligned}$$

Given this problem, we can set:

$g(x) = x_1 x_2^{0.5} x_3^{1.2}$ and $lng(x) = lnx_1 + 0.5lnx_2 + 1.2lnx_3$.

Assuming that we would like to calculate three break points for $g(x)$ within the upper and lower bounds of $lng(x)$ which are [0, 4.3455]. Following the method previously explained, the break points, a_2, a_3, a_4 can be calculated in the following way:

$$a_3 = ln\frac{e^4.3455 - e^0}{4.3455 - 0} = 2.8632,$$
$$a_2 = ln\frac{e^2.8632 - e^0}{2.8632 - 0} = 1.7525,$$
$$a_4 = ln\frac{e^4.3455 - e^2.8632}{4.3455 - 2.8632} = 3.6943.$$

The slopes can then be calculated:

$$s_1 = \frac{e^1.7525 - e^0}{1.7525 - 0} = 2.7213,$$
$$s_2 = \frac{e^2.8632 - e^1.7525}{2.8632 - 1.7525} = 10.5784,$$
$$s_3 = \frac{e^3.6943 - e^2.8632}{3.6943 - 2.8632} = 27.3143,$$
$$s_4 = \frac{e^4.3455 - e^3.6943}{4.3455 - 3.6943} = 56.6845,$$

We can also rewrite the expressions of lng as:

$$\begin{cases} 0.0 - Mq_1 \leq lng \leq 1.7525 + Mq_1 \\ 1.7525 - Mq_2 \leq lng \leq 2.8632 + Mq_2 \\ 2.8632 - Mq_3 \leq lng \leq 3.6943 + Mq_3 \\ 3.6943 - Mq_4 \leq lng \leq 4.3455 + Mq_4 \end{cases}$$

and we can express $\hat{g}$ as:

$$\begin{cases} e^{0.0} + s_1(H - 0.0) - M\prime q_1 \leq \hat{g} \\ \hat{g} \leq e^{0.0} + s_1(H - 0.0) - M\prime q_1 \\ e^{1.7525} + s_2(H - 1.7595) - M\prime q_2 \leq \hat{g} \\ \hat{g} \leq e^{1.7525} + s_2(H - 1.7525) - M\prime q_2 \\ e^{2.8632} + s_3(H - 2.8632) - M\prime q_3 \leq \hat{g} \\ \hat{g} \leq e^{2.8632} + s_3(H - 2.8632) - M\prime q_3 \\ e^{3.6943} + s_4(H - 3.6943) - M\prime q_4 \leq \hat{g} \\ \hat{g} \leq e^{3.6943} + s_4(H - 3.6943) - M\prime q_4 \end{cases}$$

where $H = lng(x) = lnx_1 + 0.5lnx_2 + 1.2lnx_3$ and

$$\begin{cases} q_1 = 0 + u_1 + u_2 \\ q_2 = 1 - u_1 + u_2 \\ q_3 = 1 + u_1 - u_2 \\ q_4 = 2 - u_1 - u_2 \end{cases}$$

Now we can use the same method to calculate the break points of lng within the range of [1,5].

$$b_{i3} = \frac{5-1}{ln5 - ln1} = 2.4853$$
$$b_{i2} = \frac{2.4853-1}{ln2.4853 - ln1} = 1.6315$$
$$b_{i4} = \frac{5-2.4853}{ln5 - ln2.4853} = 3.5974$$

. The slopes are then calculated as:

$$sl_{i1} = \frac{ln1.6315 - ln1}{1.6315 - 1} = 0.7751$$
$$sl_{i2} = \frac{ln2.4853 - ln1.6315}{2.4853 - 1.6315} = 0.4929$$
$$sl_{i3} = \frac{ln3.5974 - ln2.4853}{3.5974 - 2.4853} = 0.0.3325$$
$$sl_{i4} = \frac{ln5 - ln3.5974}{5 - 3.5974} = 0.2347$$

We can also rewrite the expressions of x_i as:

$$\begin{cases} 1.0000 - Mq_{i1} \leq x_i \leq 1.6315 + Mq_{i1} \\ 1.6315 - Mq_{i2} \leq x_i \leq 2.4853 + Mq_{i2} \\ 2.4853 - Mq_{i3} \leq x_i \leq 3.5974 + Mq_{i3} \\ 3.5974 - Mq_{i4} \leq x_i \leq 5.0000 + Mq_{i4} \end{cases}$$

and we can express lnx_i as:

$$\begin{cases} ln1 + 0.7751(x_i - 1) - Mlq_{i1} \leq lnx_i \\ lnx_i \leq ln1 + 0.7751(x_i - 1) - Mlq_{i1} \\ ln1.6315 + 0.4929(x_i - 1.6315) - Mlq_{i2} \leq lnx_i \\ lnx_i \leq ln1.6315 + 0.4929(x_i - 1.6315) - Mlq_{i2} \\ ln2.4853 + 0.0.3325(x_i - 2.4853) - Mlq_{i3} \leq lnx_i \\ lnx_i \leq ln2.4853 + 0.0.3325(x_i - 2.4853) - Mlq_{i3} \\ ln3.5974 + 0.2347(x_i - 3.5974) - Mlq_{i4} \leq lnx_i \\ lnx_i \leq ln3.5974 + 0.2347(x_i - 3.5974) - Mlq_{i4} \end{cases}$$

where $i = 1, 2, 3$ and

$$\begin{cases} q_{i1} = 0 + u_{i1} + u_{i2} \\ q_{i2} = 1 - u_{i1} + u_{i2} \\ q_{i3} = 1 + u_{i1} - u_{i2} \\ q_{i4} = 2 - u_{i1} - u_{i2} \end{cases}$$

The problem then becomes

$$\begin{array}{ll} Minimize & g + 2x_1 \\ Subject\ to & (15) - (24) \end{array}$$

This new problem now has 8 binary variables, rather than the 16 binary variables originally using Li and Chang's approach described above. The answer to this problem is $(x_1, x_2, x_3) = (1.0, 1.5, 4, 5)$ with a minimal value of 9.446[1].

Applications

There are many different applications of GPs in different fields. Here are some examples:

1. Engineering
   - Membrane separation process design
   - Chemical equilibrium problems
   - Statistical mechanics
   - Minimum weight design
   - Entropy maximization
   - Optimizing nuclear systems
   - Structural design
2. Other
   - Regional planning of economic models
   - Inventory models in management science
   - Transportation planning
   - Maximizing reliability[3]

Conclusion

In conclusion, geometric programming is a very powerful type of application that can be used to solve a variety of different optimization problems. There are many different methods to solve GPs, and it depends on the different constraints and conditions for the specific GP. Although it may be difficult to quantify a problem into a GP, doing so can be very useful to get an approximate answer, if not an exact answer, which still can be valuable. This kind of programming has applications across a variety of fields from engineering to economics, and will continue to be useful in the future as more problems are formatted into GPs.

References

1. Huang, C. H.; Kao, H. Y. (2009). An effective linear approximation method for geometric programming problems. IEEE Conference Publications. 1743-1746.

2. Boyd, S.; Kim, S. J.; Vandenberghe, L.; Hassibi, A. (2007). A tutorial on geometric programming. Springer Science+Business Media, LLC, 1-11.

3. Ecker, J. G. (1980). Geometric programming: methods, computations and applications. Society for Industrial and Applied Mathematics, 22(3), 338-341, 351- 352.