Journal for Educators, Teachers and Trainers
The PROFESIOLab electronic, peer-reviewed, open-access Magazine

ISSN 1989 –9572

DOI: 10.47750/jett.2023.14.06.011

Estimating Solution of Posynomial Geometric Programming Problems with Interval Coefficients

Harpreet Singh[1],
Amanpreet Singh[2],
Lal Singh[3]

Journal for Educators, Teachers and Trainers, Vol.14(6)

https://jett.labosfor.com/

Date of reception: 03 June 2023

Date of revision: 29 Oct 2023

Date of acceptance: 20 Nov 2023

Harpreet Singh, Amanpreet Singh, Lal Singh (2023). Estimating Solution of Posynomial Geometric Programming Problems with Interval Coefficients. *Journal for Educators, Teachers and Trainers*, Vol.14 (6). 536 –552.

Journal for Educators, Teachers and Trainers
The LabOSfor electronic, peer-reviewed, open-access Magazine

Journal for Educators, Teachers and Trainers, Vol.14(6)
ISSN1989– 9572

https://jett.labosfor.com/

Estimating Solution of Posynomial Geometric Programming Problems with Interval Coefficients

Harpreet Singh[1], Amanpreet Singh[2], Lal Singh[3]

[1]Research scholar, Desh Bhagat University, Mandi Gobindgarh&
Assistant Professor, Department of Mathematics, Guru Nanak College, Budhlada (Mansa)
Email: preethar_singh@yahoo.co.in

[2]Assistant Professor, Department of Mathematics, GSSDGS Khalsa College, Patiala
Email: drapsingh25@gmail.com

[3]Assistant Professor, Department of Computer Science, GSSDGS Khalsa College, Patiala
Email: lalsingh.kcp@gmail.com

Abstract
Posynomial Geometric Programming problems are considered to be complex in nature when the coefficients in the objective function, the constraints as well as on the right hand side of the constraints are in the form of an interval. Many approaches have been followedto find the approximate optimal solution for such type of geometric programming problems. In this paper, we have considered single objective posynomial geometric programming problem of above mentioned type. To solve such type of problems, we have converted the interval coefficient to a single number using interval valued function. Karush-Kuhn Tucker conditions are applied on this non-linear problem. To linearize the resulting non-linear problem we used first order Taylor's series expansionwhich is further solved for the optimal solution. The function codes are written in Python and executed using Google Colab due to free availability and ease of use as compared to other mathematical tools.

Keywords*: Posynomial Geometric programming, Interval valued function, Karush-Kuhn Tucker conditions, Taylor's series expansion*

1. Introduction

Geometric programming is a special type of optimization which is widely used to solve various types of real world applications in numerous fields such as engineering, management etc. The basic theory of geometric programming is to find an optimum solution for posynomial, signomial and multi-objective functions subject to certain constraints. It is different from the other programming problems in the sense that the terms involved in objective function and the constraints are not only non-linear in nature but are in more complex form. So far, massive literature has been developed and studied on GP techniques.

The theory was first introduced by Duffin, Peterson and Zener in 1967. Later, the study was continued by many researchers. Beightler et al. [1], Avriel et al. [2], Duffin et al. [3], Kortanek et al. [4, 5], Rajgopal [6] established various methods to solve posynomial and signomial geometric programming problems.

In 21st century, GP gained keen attention of the researchers. The theory was extended when the coefficients in the objective function, constraints, right hand side and exponents of the variables are multiple parameters and represented in form of an interval. Liu [7, 8], Ojha et al. [9], Mahapatra [10] gave the solution procedure to obtain the optimal solution of such type of complex problems. Ojha, Das [11], used Binary model to solve standard GPP by splitting the cost coefficients in the objective function, constraint coefficients and the exponents of the decision variable using binary numbers.

Ojha, Biswal [12] focused on the formulation of multi-objective geometric programming problems and developed a solution procedure using weighted mean method. The parametric approach of Mahapatra, Mandal [10] for single objective posynomial GP was implemented by Mousavi, Saraj [13] for multi-objective geometric programming problem with interval coefficients. Das & Roy [14], considered a Gravel box problem and the solution was compared by solving this multi-objective programming problem having two objectives using weighted-sum method, weighted-product method and weighted min-max method. Ojha, Biswal [15], developed a solution procedure to find the non-inferior solution for multi-objective programming problem using ∈-constraint method. The similar multi-objective programming problem was solved by ∈-constraint method using KKT conditions by Ojha, Ota [16]. Ojha, Ota [17], adopted a dynamic approach named hybrid method to solve MOGPP in which ∈-constraint method was integrated with that of weighted mean method. Oz et al. [18], introduce a new numerical technique in which the weighted objective function was minimized by KKT conditions followed by first order Taylor's series approximation. In Lexicogaphic multi-objective programming problem by Ojha, Biswal [19], objective functions can be prioritize and ranked by using row –column adoption method and eigen value method.

The idea of fuzziness was laid by Cao [20]. Biswal [21] proposed fuzzy programming technique to solve multi-objective GP.

In this paper, we considered single objective programming problem in which the cost coefficient in the objective function is represented in the form of an interval. To find the optimal solution for such type of problems we have firstly converted the interval coefficient to a single value by using interval valued function. Afterwards, we applied alternative approach proposed by Erzoy et al. [18] to obtain the pareto optimal solution. Using current approach, it will not be required to convert the problem to its dual. Moreover, it will be applicable to the GP with any degree of difficulty. Programming is done in Python to calculate the values of the objective function and constraints at initial feasible point and also to find the optimal solution of the transformed linear programming problem and the main GP problem. A web-based Python IDE platform called Google Colab is used to execute the programs.

2. Preliminaries

2.1 Posynomial Geometric programming problem

The Standard posynomial geometric programming problem is written as:

$$\text{Minimize} \quad f_0(x) = \sum_{i=1}^{n} p_i \prod_{j=i}^{m} x_j^{q_{ij}}$$

$$\text{Subject to} \quad f_k(x) = \sum_{i=1}^{l_k} r_{ik} \prod_{j=i}^{m} x_j^{s_{ij}} \leq 1, \qquad k = 1,2,\ldots..,t$$

$$x_j > 0, \qquad j = 1,2,\ldots....,m$$

the exponents q_{ij}, s_{ij} can assume arbitrary real values where as the coefficients p_i, r_{ik} are assumed to be positive.

2.2 Posynomial Geometric programming problem with interval coefficients

The general form of posynomial geometric programming problem with interval coefficients can be written as:

Minimize $$f_0(x) = \sum_{i=1}^{n} [p_i, q_i] \prod_{j=i}^{m} x_j^{q_{ij}}$$

Subject to $$f_k(x) = \sum_{i=1}^{l_k} [p_{ik}, q_{ik}] \prod_{j=i}^{m} x_j^{s_{ij}} \le [b_{ik}, c_{ik}], \quad k = 1,2,\ldots..,t$$

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

the exponents q_{ij}, s_{ij} can assume arbitrary real values where as the coefficients $p_i, q_i, p_{ik}, q_{ik}, b_{ik}, c_{ik}$ are assumed to be positive.

2.3 Interval-valued function

Consider an interval $[p, q]$ with $p > 0, q > 0$. Then the interval valued function is defined as

$$f(t) = p^{1-t} q^t \quad for \quad t \in [0,1]$$

2.4 KKT conditions

For general mathematical programming problem:

$$Min\ f_0(x)$$

Subject to $$f_k(x) \le 0, \quad k = 1,2,\ldots\ldots\ldots t$$

$$x_i > 0, \quad i = 1,2,\ldots\ldots\ldots,t$$

Then KKT conditions are stated as follows:

$$\frac{\partial f_0(x)}{\partial x_i} + \sum_{k=1}^{t} \lambda_k \frac{\partial f_k(x)}{\partial x_i} = 0, \quad i = 1,2\ldots\ldots,t$$

$$f_k(x) \le 0, \quad k = 1,2,\ldots\ldots\ldots t$$

$$\lambda_k f_k(x) = 0, \quad k = 1,2,\ldots\ldots\ldots t$$

$$x_i > 0, \quad i = 1,2,\ldots\ldots,t$$

2.5 Taylor's series expansion for multivariable

Consider a function $f(\mathbf{x})$. Let $\mathbf{x} = (x_1, x_2, \ldots\ldots\ldots\ldots, x_n) \in R^n$ and $\mathbf{x}^0 = (x_1^0, x_2^0, \ldots\ldots\ldots x_n^0) \in R^n$. Let f be differentiable in $N(\mathbf{x}^0)$. Then the first order Taylor's series expansion of f about $\mathbf{x}^0$ can be written as:

$$f(x) = f(\mathbf{x}^0) + \sum_{i=1}^{n} f_{x_i}(\mathbf{x}^0)(x_i - x_i^0), \quad f_{x_i} = \frac{\partial f}{\partial x_i}$$

or

$$f(x) = [f(x)]_{\mathbf{x}^0} + \left[\frac{\partial f}{\partial x_1}\right]_{\mathbf{x}^0} (x_1 - x_1^0) + \left[\frac{\partial f}{\partial x_2}\right]_{\mathbf{x}^0} (x_2 - x_2^0) + \ldots\ldots\ldots + \left[\frac{\partial f}{\partial x_n}\right]_{\mathbf{x}^0} (x_n - x_n^0)$$

3. A Overview of Software used for Computation

3.1 About Python

Python is widely used in Mathematics as it is simple and having a rich set of libraries such as NumPy, SciPy, and SymPy. These libraries offer robust tools for numerical and symbolic computation. Python also provide powerful visualization options and can integrate with other languages for different tasks which require high performance. Its active community, extensive resources, and the interactive environment of Jupyter Notebooks

further support its use in Mathematical research and education. In additionto this, Python's versatility makes it applicable in fields like data science and engineering, and it is open source, making it free to use.

3.2 Significance of Google Colab

Google Colaboratory named as Google colab provided by Google is a cloud based platform which allow the users to write and execute Python codes in their browsers. Some of important aspects of Google colab are as follows:

- It is free to use and easily accessible through any web browser without installing any software in the computer.
- The only requirement is to create a Google account and no setup is required.
- Same notebook can be accessible to multiple users to work on the same project.
- Can save work as a notebook which can be downloaded and shared to anyone anywhere worldwide just like a document.

4. Algorithm to solve the problem

The steps to find the optimal solution are as follows:

Step I Choose the problem having interval coefficients and apply interval valued function to convert the interval coefficients to a single number.

Step II Form a new function with the help of Lagrange multiplier.

Step III Apply KKT conditions to obtain the constraints for the new function.

Step IV Use Taylor's theorem for multivariable to convert the non-linear model (obtained in step II and step III) to linear system.

Step V Use Python Programming to calculate the values of the objective function and constraints at initial feasible point andalso to find the optimal solution of the GPP. UseGoogle Colab to execute the programs.

5. Numerical example

In this section, we have taken an example to illustrate the proposed structure of the current algorithm. We have considered the same problem that was also evaluated by Liu [10] and Mahapatra, Mandal [11] to check the applicability and the accuracy of the algorithm. In this example, the interval coefficient is considered only in the objective function.

Problem: Min $f(x) = (20,70)\, x_1^{-1} x_2^{-1/2} x_3^{-1} + 20\, x_1 x_3 + 20\, x_1 x_2 x_3$

Subject to
$$\frac{1}{3} x_1^{-2} x_2^{-2} + \frac{4}{3} x_2^{-1/2} x_3^{-1} \le 1,$$
$$x_1, x_2, x_3 > 0$$

5.1 Solution by Liu approach

In this approach, the upper and the lower bound of the objective value of the given posynomial geometric programming problem is obtained. The lower bound of the solution is obtained by setting the lower end of the interval as the coefficient and the upper bound is obtained accordingly. Consider Z^L and Z^U be the objective values corresponding to lower and upper objective value. Thus the pair of geometric programs is as follows:

$$Z^L = \underset{w}{Max} \left(\frac{20}{w_{01}}\right)^{w_{01}} \left(\frac{20}{w_{02}}\right)^{w_{02}} \left(\frac{20}{w_{03}}\right)^{w_{03}} \left(\frac{w_{11}+w_{12}}{3w_{11}}\right)^{w_{11}} \left(\frac{4(w_{11}+w_{12})}{3w_{12}}\right)^{w_{12}}$$

Subject to
$$w_{01} + w_{02} + w_{03} = 1,$$
$$-w_{01} + w_{02} + w_{03} - 2w_{11} = 0,$$
$$-\frac{1}{2} w_{01} + w_{03} - 2w_{11} + \frac{1}{2} w_{12} = 0,$$
$$-w_{01} + w_{02} + w_{03} - w_{12} = 0,$$
$$w_{01}, w_{02}, w_{03}, w_{11}, w_{12} \ge 0, \quad ;$$

$$Z^U = \underset{w}{Max} \left(\frac{70}{w_{01}}\right)^{w_{01}} \left(\frac{20}{w_{02}}\right)^{w_{02}} \left(\frac{20}{w_{03}}\right)^{w_{03}} \left(\frac{w_{11}+w_{12}}{3w_{11}}\right)^{w_{11}} \left(\frac{4(w_{11}+w_{12})}{3w_{12}}\right)^{w_{12}}$$

Subject to

$$w_{01}+w_{02}+w_{03}=1,$$
$$-w_{01}+w_{02}+w_{03}-2w_{11}=0,$$
$$-\frac{1}{2}w_{01}+w_{03}-2w_{11}+\frac{1}{2}w_{12}=0,$$
$$-w_{01}+w_{02}+w_{03}-w_{12}=0,$$
$$w_{01}, w_{02}, w_{03}, w_{11}, w_{12} \geq 0,$$

After solving, the lower bound of the objective value comes out to be $Z^L = 90$ corresponding to $x_1=1, x_2=1, x_3=2$ and the upper bound of the objective value is $Z^U = 115$ corresponding to $x_1=1, x_2=1, x_3=2$.

5.2 Solution by Mahapatra and Mandal approach

Using parametric form of an interval, the given problem is transformed to the following form:

$$\text{Min} \quad f(x,q) = 20^{1-q}\, 70^q\, x_1^{-1} x_2^{-\frac{1}{2}} x_3^{-1} + 20\, x_1 x_3 + 20\, x_1 x_2 x_3$$

Subject to

$$\frac{1}{3} x_1^{-2} x_2^{-2} + \frac{4}{3} x_2^{-\frac{1}{2}} x_3^{-1} \leq 1,$$
$$x_1, x_2, x_3 > 0 \text{ where } \quad 0 \leq q \leq 1$$

To obtain the solution, the problem thus converted to its corresponding dual problem:

$$\text{Max } d(w,q) = \left(\frac{20^{1-q}\, 70^q}{w_{01}}\right)^{w_{01}} \left(\frac{20}{w_{02}}\right)^{w_{02}} \left(\frac{20}{w_{03}}\right)^{w_{03}} \left(\frac{w_{11}+w_{12}}{3w_{11}}\right)^{w_{11}} \left(\frac{4(w_{11}+w_{12})}{3w_{12}}\right)^{w_{12}}$$

Subject to

$$w_{01}+w_{02}+w_{03}=1,$$
$$-w_{01}+w_{02}+w_{03}-2w_{11}=0,$$
$$-\frac{1}{2}w_{01}+w_{03}-2w_{11}+\frac{1}{2}w_{12}=0,$$
$$-w_{01}+w_{02}+w_{03}-w_{12}=0,$$
$$w_{01}, w_{02}, w_{03}, w_{11}, w_{12} \geq 0, \qquad q \in [0,1]$$

The given problem is having degree of difficulty 1, thus from the above relations the value of all the variables is written in terms of any variable say w_{12} as:

$$w_{01} = \frac{1-w_{12}}{2}, \qquad w_{02} = \frac{3w_{12}-1}{4}, \qquad w_{03} = \frac{3-w_{12}}{4}, \qquad w_{11} = \frac{1}{2} w_{12}$$

Therefore the dual of the problem is of the form:

$$d(w,q) = \left(2\,\frac{20^{1-q}\, 70^q}{1-w_{12}}\right)^{\frac{1-w_{12}}{2}} \left(\frac{80}{3w_{12}-1}\right)^{\frac{3w_{12}-1}{4}} \left(\frac{80}{3-w_{12}}\right)^{\frac{3-w_{12}}{4}} 2^{w_{12}}$$

Thus solving the above dual problem using GP technique, the optimal solutions of the problem for different values of the parameter q are represented in the following table:

Table 1. Optimal solutions of PGP

q	x_1	x_2	x_3	$f(x)$	Objective function

0.0	1.431749	1.039924	0.6984466	39.60562	60.408423
0.1	1.432887	1.042531	0.6978919	39.60774	63.052731
0.2	1.434099	1.045312	0.6973020	39.61182	66.036641
0.3	1.435390	1.048276	0.6966747	39.61512	69.411651
0.4	1.436766	1.051436	0.6960077	39.62690	73.221986
0.5	1.438231	1.054807	0.6952986	39.63848	77.527569
0.6	1.439792	1.058401	0.6945447	39.65317	82.391603
0.7	1.441456	1.062235	0.6937432	39.67134	87.885881
0.8	1.443228	1.066325	0.6928914	39.69338	94.090929
0.9	1.445116	1.070688	0.6919861	39.71974	101.097752
1.0	1.447128	1.075343	0.6910240	39.75087	109.011978

5.3 Solution by alternative approach(proposed method)

Step I: Using interval valued function (def. 2.3); the given problem can be transformed to the following form:

$$\text{Min } [f(x)]_q = 20^{1-q}\,70^q\,x_1^{-1}\,x_2^{-1/2}\,x_3^{-1} + 20\,x_1\,x_3 + 20\,x_1\,x_2\,x_3$$

Subject to $\quad \frac{1}{3}x_1^{-2}\,x_2^{-2} + \frac{4}{3}x_2^{-1/2}\,x_3^{-1} \le 1,$

$$x_1, x_2, x_3 > 0 \quad \text{where} \quad 0 \le q \le 1$$

By introducing the slack variables, the above problem takes the form:

$$\text{Min } [f(x)]_q = 20^{1-q}\,70^q\,x_1^{-1}\,x_2^{-1/2}\,x_3^{-1} + 20\,x_1\,x_3 + 20\,x_1\,x_2\,x_3$$

Subject to $\quad \frac{1}{3}x_1^{-2}\,x_2^{-2} + \frac{4}{3}x_2^{-1/2}\,x_3^{-1} + \gamma_1^2 - 1 = 0,$

$$x_1, x_2, x_3 > 0 \quad \text{where} \quad 0 \le q \le 1$$

Step II: Analogous to Lagrangian theorem and with the help of multiplier (y_1, say), the aboveproblem can be defined as:

$$\text{Min } [f(x)]_{q,y_1} = 20^{1-q}\,70^q\,x_1^{-1}\,x_2^{-1/2}\,x_3^{-1} + 20\,x_1\,x_3 + 20\,x_1\,x_2\,x_3 - y_1\left(1 - \frac{1}{3}x_1^{-2}\,x_2^{-2} - \frac{4}{3}x_2^{-1/2}\,x_3^{-1} - \gamma_1^2\right)$$

Step III: For its local minima we apply KKT conditions, the problem takes the form:

$$\text{Min } [f(x)]_{q,y_1} = 20^{1-q}\,70^q\,x_1^{-1}\,x_2^{-1/2}\,x_3^{-1} + 20\,x_1\,x_3 + 20\,x_1\,x_2\,x_3$$

Subject to $\quad -20^{1-q}\,70^q\,x_1^{-2}\,x_2^{-1/2}\,x_3^{-1} + 20x_3 + 20x_2\,x_3 - \frac{2}{3}y_1\,x_1^{-3}\,x_2^{-2} = 0$

$$-\frac{20^{1-q}\,70^q}{2}x_1^{-1}\,x_2^{-3/2}\,x_3^{-1} + 20x_1\,x_3 - \frac{2}{3}y_1\,x_1^{-2}\,x_2^{-3} - \frac{2}{3}y_1\,x_2^{-3/2}\,x_3^{-1} = 0$$

$$-20^{1-q}\,70^q\,x_1^{-1}\,x_2^{-1/2}\,x_3^{-2} + 20x_1 + 20x_1\,x_2 - \frac{4}{3}y_1\,x_2^{-1/2}\,x_3^{-2} = 0$$

$$y_1\left(1 - \frac{1}{3}x_1^{-2}\,x_2^{-2} - \frac{4}{3}x_2^{-1/2}\,x_3^{-1}\right) \ge 0$$

Step IV: Using first order Taylor's series expansion for multivariable about any initial feasible point X^0, the above non-linear problem can be converted to linear programming problem as below:

$$Min\,[f(x)] \approx \left[20^{1-q}\,70^{q}\,x_1^{-1}x_2^{-1/2}x_3^{-1}+20x_1x_3+20x_1x_2x_3\right]_{X^0}+\left[-20^{1-q}\,70^{q}\,x_1^{-2}x_2^{-1/2}x_3^{-1}+20x_3+20x_2x_3\right]_{X^0}(x_1-x_1^0)+$$
$$\left[-\frac{20^{1-q}\,70^{q}}{2}x_1^{-1}x_2^{-3/2}x_3^{-1}+20x_1x_3\right]_{X^0}(x_2-x_2^0)+\left[-20^{1-q}\,70^{q}\,x_1^{-1}x_2^{-1/2}x_3^{-2}+20x_1+20x_1x_2\right]_{X^0}(x_3-x_3^0)$$

… (1)

Subject to

$$\left[-20^{1-q}\,70^{q}\,x_1^{-2}x_2^{-1/2}x_3^{-1}+20x_2x_3-\frac{2}{3}y_1x_1^{-3}x_2^{-2}\right]_{X^0}+\left[2\times20^{1-q}\,70^{q}\,x_1^{-3}x_2^{-1/2}x_3^{-1}+2y_1x_1^{-4}x_2^{-2}\right]_{X^0}(x_1-x_1^0)+$$
$$\left[\frac{20^{1-q}\,70^{q}}{2}x_1^{-2}x_2^{-3/2}x_3^{-1}+\frac{4}{3}y_1x_1^{-3}x_2^{-3}\right]_{X^0}(x_2-x_2^0)+\left[20^{1-q}\,70^{q}\,x_1^{-2}x_2^{-1/2}x_3^{-2}+20+20x_2\right]_{X^0}(x_3-x_3^0)+\left[-\frac{2}{3}x_1^{-3}x_2^{-2}\right]_{X^0}(y_1-y_1^0)=0$$

… (2)

$$\left[-\frac{20^{1-q}\,70^{q}}{2}x_1^{-1}x_2^{-3/2}x_3^{-1}+20x_1x_3-\frac{2}{3}y_1x_1^{-2}x_2^{-3}-\frac{2}{3}y_1x_2^{-3/2}x_3^{-1}\right]_{X^0}+\left[\frac{20^{1-q}\,70^{q}}{2}x_1^{-2}x_2^{-3/2}x_3^{-1}+20x_3+\frac{4}{3}y_1x_1^{-3}x_2^{-3}\right]_{X^0}(x_1-x_1^0)+$$
$$\left[\frac{3\times20^{1-q}\,70^{q}}{4}x_1^{-1}x_2^{-5/2}x_3^{-1}+2y_1x_1^{-2}x_2^{-4}+y_1x_2^{-5/2}x_3^{-1}\right]_{X^0}(x_2-x_2^0)+\left[\frac{20^{1-q}\,70^{q}}{2}x_1^{-1}x_2^{-3/2}x_3^{-2}+20x_1+\frac{2}{3}y_1x_2^{-3/2}x_3^{-2}\right]_{X^0}(x_3-x_3^0)+$$
$$\left[-\frac{2}{3}x_1^{-2}x_2^{-3}-\frac{2}{3}x_2^{-3/2}x_3^{-1}\right]_{X^0}(y_1-y_1^0)=0$$

… (3)

$$\left[-20^{1-q}\,70^{q}\,x_1^{-1}x_2^{-1/2}x_3^{-2}+20x_1+20x_1x_2-\frac{4}{3}y_1x_1^{-1/2}x_2^{-2}\right]_{X^0}+\left[20^{1-q}\,70^{q}\,x_1^{-2}x_2^{-1/2}x_3^{-2}+20+20x_2\right]_{X^0}(x_1-x_1^0)+$$
$$\left[\frac{20^{1-q}\,70^{q}}{2}x_1^{-1}x_2^{-3/2}x_3^{-2}+20x_1+\frac{2}{3}y_1x_2^{-3/2}x_3^{-2}\right]_{X^0}(x_2-x_2^0)+\left[2\times20^{1-q}\,70^{q}\,x_1^{-1}x_2^{-1/2}x_3^{-3}+\frac{8}{3}y_1x_2^{-1/2}x_3^{-3}\right]_{X^0}(x_3-x_3^0)+\left[-\frac{4}{3}x_2^{-1/2}x_3^{-2}\right]_{X^0}(y_1-y_1^0)=0$$

… (4)

$$\left[y_1\left(1-\frac{1}{3}x_1^{-2}x_2^{-2}-\frac{4}{3}x_2^{-1/2}x_3^{-1}\right)\right]_{X^0}+\left[y_1\left(\frac{2}{3}x_1^{-3}x_2^{-2}\right)\right]_{X^0}(x_1-x_1^0)+\left[y_1\left(\frac{2}{3}x_1^{-2}x_2^{-3}+\frac{2}{3}x_2^{-3/2}x_3^{-1}\right)\right]_{X^0}(x_2-x_2^0)+\left[y_1\left(\frac{4}{3}x_2^{-1/2}x_3^{-2}\right)\right]_{X^0}(x_3-x_3^0)+$$
$$\left[\left(1-\frac{1}{3}x_1^{-2}x_2^{-2}-\frac{4}{3}x_2^{-1/2}x_3^{-1}\right)\right]_{X^0}(y_1-y_1^0)\geq0$$

… (5)

Step V: Since q lies between 0 and 1, thus taking $q=0$ and assuming the initial feasible point to be:

$$X^0=(x_1^0=2,x_2^0=2,x_3^0=1,y_1^0=2),$$

the linear programming problem (1) to (5) takes the form as below. It is noted that all the calculations are done on Google Colab. The codes are written in Python individually for every equation. For the sake of calculations, the variables x1, x2, x3 and y1 are replaced by a1, a2, a3 and b1 respectively whereas the initial point $x_1^0, x_2^0, x_3^0, y_1^0$ is taken as x1, x2, x3 and y1.

5.3.1 Python code of objective function eq. (1) for the linearized programming problem:

```
minz=(((pow(20,1-q)*pow(70,q)*pow(x1,-1)*pow(x2,-1/2)*pow(x3,-1))+(20*x1*x3)+(20*x1*x2*x3))+((-
pow(20,1-q)*pow(70,q)*pow(x1,-2)*pow(x2,-1/2)*pow(x3,-1)+20*x3+20*x2*x3)*(a1-x1))+(((-pow(20,1-
q)*pow(70,q)/2)*pow(x1,-1)*pow(x2,-3/2)*pow(x3,-1)+20*x1*x3)*(a2-x2))+((-pow(20,1-
q)*pow(70,q)*pow(x1,-1)*pow(x2,-1/2)*pow(x3,-2)+20*x1+20*x1*x2)*(a3-x3)))
smpl = simplify(minz)
smpl
```

Substituting initial values, Objective function takes the form:

$56.4644660940673a1+38.2322330470336a2+112.928932188135a3-175.251262658471$

5.3.2 Python code for first constraint eq. (2):

```
C1=(((-pow(20,1-q)*pow(70,q)*pow(x1,-2)*pow(x2,-1/2)*pow(x3,-1))+(20*x3)+(20*x2*x3)-((2*y1*pow(x1,-3)*pow(x2,-2))/3))+((2*pow(20,1-q)*pow(70,q)*pow(x1,-3)*pow(x2,-1/2)*pow(x3,-1)+2*y1*pow(x1,-4)*pow(x2,-2))*(a1-x1))+((((pow(20,1-q)*pow(70,q)*pow(x1,-2)*pow(x2,-3/2)*pow(x3,-1))/2)+20*x3+((4*y1*pow(x1,-3)*pow(x2,-3))/3))*(a2-x2))+((pow(20,1-q)*pow(70,q)*pow(x1,-2)*pow(x2,-1/2)*pow(x3,-2)+20+20*x2)*(a3-x3))-(((2*pow(x1,-3)*pow(x2,-2))/3)*(b1-y1)))
smpl = simplify(C1)
smpl
```

Similarly, first constraint after simplification becomes:
$3.59803390593274a1+20.9255501431499a2+63.5355339059327a3-0.0208333333333333b1-56.1182359100307=0$

5.3.3 Code for second constraint eq. (3):

```
C2=((((-pow(20,1-q)*pow(70,q)*pow(x1,-1)*pow(x2,-3/2)*pow(x3,-1))/2)+(20*x3*x1)-((2*y1*pow(x1,-2)*pow(x2,-3))/3)-((2*y1*pow(x2,-3/2)*pow(x3,-1))/3))+((((pow(20,1-q)*pow(70,q)*pow(x1,-2)*pow(x2,-3/2)*pow(x3,-1))/2)+20*x3+((4*y1*pow(x1,-3)*pow(x2,-3))/3))*(a1-x1))+((((3*pow(20,1-q)*pow(70,q)*pow(x1,-1)*pow(x2,-5/2)*pow(x3,-1))/4)+(2*y1*pow(x1,-2)*pow(x2,-4))+y1*pow(x2,-5/2)*pow(x3,-1))*(a2-x2))+((((pow(20,1-q)*pow(70,q)*pow(x1,-1)*pow(x2,-3/2)*pow(x3,-2))/2)+20*x1+((2*y1*pow(x2,-3/2)*pow(x3,-2))/3))*(a3-x3))+((((-2*pow(x1,-2)*pow(x2,-3))/3)-((2*pow(x2,-3/2)*pow(x3,-1))/3))*(b1-y1)))
smpl = simplify(C2)
smpl
```

Second constraint after simplification:
$20.9255501431499a1+1.74187860531805a2+42.2391714737574a3-0.256535593728849b1-49.3417959236596=0$

5.3.4 Code for third constraint eq. (4):

```
C3=(((-pow(20,1-q)*pow(70,q)*pow(x1,-1)*pow(x2,-1/2)*pow(x3,-2))+(20*x1)+(20*x1*x2)-((4*y1*pow(x2,-1/2)*pow(x3,-2))/3))+((pow(20,1-q)*pow(70,q)*pow(x1,-2)*pow(x2,-1/2)*pow(x3,-2)+20+20*x2)*(a1-x1))+((((pow(20,1-q)*pow(70,q)*pow(x1,-1)*pow(x2,-3/2)*pow(x3,-2))/2)+20*x1+((2*y1*pow(x2,-3/2)*pow(x3,-2))/3))*(a2-x2))+((2*pow(20,1-q)*pow(70,q)*pow(x1,-1)*pow(x2,-1/2)*pow(x3,-3)+((8*y1*pow(x2,-1/2)*pow(x3,-3))/3))*(a3-x3))-(((4*pow(x2,-1/2)*pow(x3,-2))/3)*(b1-y1)))
smpl = simplify(C3)
smpl
```

Third constraint after simplification:
$63.5355339059327a1+42.2391714737574a2+17.9133717900592a3-0.942809041582063b1-116.533850361305=0$

5.3.5 Code for fourth constraint eq. (5):

```
C4=((y1*(1-((pow(x1,-2)*pow(x2,-2))/3)-((4*pow(x2,-1/2)*pow(x3,-1))/3)))+(((2*y1*pow(x1,-3)*pow(x2,-2))/3)*(a1-x1))+((y1*(((2*pow(x1,-2)*pow(x2,-3))/3)+((2*pow(x2,-3/2)*pow(x3,-1))/3)))*(a2-x2))+(((y1*4*pow(x2,-1/2)*pow(x3,-2))/3)*(a3-x3))+((1-((pow(x1,-2)*pow(x2,-2))/3)-((4*pow(x2,-1/2)*pow(x3,-1))/3))*(b1-y1)))
smpl = simplify(C4)
smpl
```

Fourth constraint after simplification:
$0.0416666666666667a1+0.513071187457698a2+1.88561808316413a3+0.0363576250846032b1-2.99509379141286 \geq 0$

Thus the resulting LPP corresponding to assumed feasible solution X^0 and for $q = 0$, rounded off to four decimal places, takes the form:

$$\text{Min } Z = 56.4645\ x_1 + 38.2322\ x_2 + 112.9289\ x_3 - 175.2513$$

$$3.5980\ x_1 + 20.9256\ x_2 + 63.5356\ x_3 - 0.0208\ y_1 - 56.1182 = 0$$

$$20.9256\ x_1 + 1.7419\ x_2 + 42.2392\ x_3 - 0.2566\ y_1 - 49.3418 = 0$$

$$63.5356\ x_1 + 42.2392\ x_2 + 17.9134\ x_3 - 0.9428\ y_1 - 116.5339 = 0$$

$$0.0417\ x_1 + 0.5131\ x_2 + 1.8856\ x_3 + 0.0364\ y_1 - 2.9951 \geq 0$$

Thus our main non-linear problem is reduced into linear form which will be solved for the optimal solution. The calculations are again done using Python in Google Colab.

5.3.6 The corresponding codes for the solution of the above linear programming problemare given as:

```
pip install pulp
from pulp import *
importmatplotlib.pyplotasplt
importnumpyas np

# Create an object of a model
prob = LpProblem("Simple LP Problem", LpMinimize)

# Define the decision variables
x1 = LpVariable("x1", 0)
x2 = LpVariable("x2", 0)
x3 = LpVariable("x3", 0)
y1= LpVariable("y1", 0)

# Define the objective function
prob += (56.4645*x1) + (38.2322*x2) + (112.9289*x3) - (175.2513)

# Define the constraints
prob += (3.5980*x1) + (20.9256*x2) + (63.5355*x3) - (0.0208*y1) - (56.1182) == 0, "1st constraint"
prob += (20.9256*x1) + (1.7419*x2) + (42.2391*x3) - (0.2565*y1) - (49.3418) == 0, "2nd constraint"
prob += (63.5355*x1) + (42.2392*x2) + (17.9134*x3) - (0.9428*y1) - (116.5339) == 0, "3rd constraint"
prob += (0.0417*x1) + (0.5131*x2) + (1.8856*x3) + (0.0364*y1) - (2.9951) >= 0, "4th constraint"

Code cell <iOZ6S-uhUqjM>
# %% [code]
prob.solve()
      1

# Print the results
print ("Status: ", LpStatus[prob.status])
      Status:  Optimal

for v inprob.variables():
   print (v.name, "=", v.varValue)

      x1 = 1.7434617
      x2 = 0.84012202
      x3 = 0.52139342
      y1 = 41.433836
```

5.3.7 Codes for the Optimal solution corresponding to x1, x2, x3, y1 and q=0 (rounded off upto four decimal places):

```
fromsympyimport *

q=0
```

```
x1=1.7435
x2=0.8401
x3=0.5214
minz=((pow(20,1-q)*pow(70,q)*pow(x1,-1)*pow(x2,-1/2)*pow(x3,-1))+20*x1*x3+20*x1*x2*x3)
smpl = simplify(minz)
smpl

57.4583497077539
```

The values of x_1, x_2 , x_3, y_1 and the optimal solutions for the objective function corresponding to different values of q are represented in the following table:

Table 2. Optimal solution of PGP by alternative approach

q	x_1	x_2	x_3	y	Solutions of Objective function of linear system	Optimal solutions of Objective function for the main problem
0.0	1.7435	0.8401	0.5214	41.4338	14.1925	57.4586
0.1	1.7494	0.8664	0.5271	40.7617	17.9495	60.8321
0.2	1.7556	0.8963	0.5332	40.0151	22.1184	64.4955
0.3	1.7627	0.9292	0.5401	39.1865	26.7537	68.4686
0.4	1.7702	0.9664	0.5475	38.2703	31.8764	72.7636
0.5	1.7784	1.0076	0.5557	37.2587	37.5377	77.3986
0.6	1.7871	1.0535	0.5645	36.1454	43.7701	82.3903
0.7	1.7964	1.1046	0.5739	34.9240	50.6101	87.7594
0.8	1.8061	1.1615	0.5840	33.5880	58.0917	93.5288
0.9	1.8161	1.2249	0.5947	32.1312	66.2455	99.7252
1.0	1.8262	1.2954	0.6058	30.5474	75.1022	106.3814

The lower and upper objective values thus obtained are nearly the same as obtained in other approaches. Also the intermediate values can be obtained corresponding to values of q according to the requirement of the decision makers.

6. Conclusion

One problem that can be effectively addressed using interval valued is the scheduling of project tasks with uncertain durations. By representing the task durations as intervals rather than single point values, the scheduling algorithm can account for the inherent variability and uncertainty in the estimates. This allows for more robust and flexible scheduling plans that can adapt to changes and delays without requiring constant adjustments. Additionally, interval valued can help identify critical paths and potential bottlenecks in the project timeline, enabling better resource allocation and risk management. While converting interval coefficients to a single number can be useful for simplifying calculations, it may not necessarily provide a more accurate representation of critical paths and potential bottlenecks in the project timeline compared to other methods. Additionally, the complexity and time required for implementing multiple mathematical techniques and programming languages may outweigh the benefits gained from using them in this context. Therefore, it is important to carefully weigh the trade-offs between accuracy and efficiency when choosing the appropriate method for analyzing project constraints. It may be beneficial to consult with experts in operations research or project management to determine the most suitable approach for a specific project. By considering all factors,

including the level of detail required, the available resources, and the desired outcome, project managers can make informed decisions that will ultimately lead to successful project completion.

References

1. Beightler, C., & Phillips, D. (1976). *Applied Geometric Programming.* Wiley.
2. Avriel, M., Dembo, R., & Passy, U. (1975). Solution of Generalized Geometric Programs. *International Journal for Numerical Methods in Engineering* , Vol. 9.
3. Duffin, R., & Peterson, E. (1973). Geometric Programming with Signomials. *Journal of Optimization Theory and Applications, 11 No.1*.
4. Kortanek, K., & Hoon, N. (1992). A Second Order Affine Scaling Algorithm for the Geometric Progeamming Dual with Logarithmic Barrier. *Optimization: A Journal of Mathematical Programming and Operations Research, 23*, 303-322.
5. KortanekK.O, Xu, X., & Ye, Y. (1996). An Infeasible Interior-point Algorithm for Solving Primal and Dual Geometric Programs. *Mathematical Programming, 76*, 155-181.
6. Rajgopal, J. (1992). An alternative approach to the refined duality theory of geometric programming. *Journal of mathematical analysis and applications, 167*, 266-288.
7. Liu, S.-T. (2006). Posynomial Geometric Programming with parametric uncertainty. *European journal of operational research, 168*, 345-353.
8. Liu, S.-T. (2008). Posynomial Geometric Programming with interval exponents and coefficients. *European journal of operational research, 186*, 17-27.
9. Ojha, A., & Biswal, K. (January 2010). Posynomial geometric programming problems with multiple parameters. *Journal of computing, vol. 2* (issue 1).
10. Mahapatra, G., & Mandal, T. (2012). Posynomial parametric geometric programming with interval valued coefficient. *J Optim Theory Appl, 154*, 120-132.
11. Ojha, A., & Das, A. (January 2010). Geometric programming problem with coefficients and Exponents Associated with Binary numbers. *International journal of computer science, vol. 7* (issue 1, no. 2).
12. Ojha, A., & Biswal, K. (2010). Multi-objective Geometric Programming problem with weighted mean method. *International Journal of Computer science and Information security, vol. 7* (no. 2).
13. Mousavi, Z., & Saraj, M. (2019). Multi-objective Geometric Programming with interval coefficients: A Parametric approach. *Earthline journal of Mathematical Sciences, vol. 2* (no. 2), 395-407.
14. Das, P., & Roy, T. K. (2014). Solving a Multi-Objective Geometric Programming and its Application in Gravel box Problem. *Journal of Global Research in Computer Science, 5 no. 7*.
15. Ojha, A., & Ota, R. R. (2013). Multi-objective geometric programming problems with cost co-efficient as multiple parameters. *Advanced Modelling and Optimization, vol. 15* (no. 3).
16. Ojha, A., & Biswal, K. (2013). Multi-objective Geometric Programming Problem with Constraint Method. *Applied mathematical Modeling*.
17. Ojha, A., & Ota, R. R. (2015). A Hybrid Method for Solving Multi-objective Geometric Programming Problem. *Int. J. Mathematics in Operational Research, Vol. 7* (No. 2).
18. Oz, E., Guzel, N., & Alp, S. (2017). An Alternative Approach to the solution of Multi-Objective Geometric Programming Problem. *Open Journal of Optimization, 6*, 11-25.
19. Ojha, A., & Biswal, K. (2009). Lexicographic Multi-Objective Programming Problems. *International Journal of Computer Science, 6* (2), 20-24.
20. Bing-yuan, C. (1993). Fuzzy geometric Programming. *Fuzzy Sets and Systems, 53*, 135-153.
21. Biswal, M. (1992). Fuzzy Programming Technique to Solve Multi-Objective Geometric Programming Problems. *Fuzzy Sets and Systems, 51*, 67-71.