

Geometric Programming Problem with Co-Efficients and Exponents Associated with Binary Numbers

A.K.Ojha¹ and A.K.Das²

¹ School of Basic Sciences, I.I.T Bhubaneswar
Orissa-751013, India

² Department of Mathematics, Bhadrak College
Bhadrak, Orissa-756100, India

Abstract

Geometric programming (GP) provides a power tool for solving a variety of optimization problems. In the real world, many applications of geometric programming (GP) are engineering design problems in which some of the problem parameters are estimating of actual values. This paper develops a solution procedure to solve nonlinear programming problems using GP technique by splitting the cost coefficients, constraint coefficients and exponents with the help of binary numbers. The equivalent mathematical programming problems are formulated to find their corresponding value of the objective function based on the duality theorem. The ability of calculating the cost coefficients, constraint coefficients and exponents developed in this paper might help lead to more realistic modeling efforts in engineering design areas. Standard nonlinear programming software has been used to solve the proposed optimization problem. Two numerical examples are presented to illustrate the method.

Keywords: *Geometric programming, Posynomial, Binary number, Duality theorem, Optimization.*

1. Introduction

Geometric programming (GP) a technique developed for solving algebraic non-linear programming problems subject to linear or nonlinear constraints is useful in the study of a variety of optimization problems. Duffin, Peterson and Zener [8] put a foundation stone to solve wide range of engineering problems by developing basic theories of geometric programming and its application in their text book. Geometric programming derives its name from its intimate connection with geometrical concepts because the method is based on geometric inequality and their properties that relate sums and products of positive numbers. The application of geometric inequality has also been very useful in the construction of the condensation technique for posynomial problems [21, 26]. The notion of a condensed posynomial problem is introduced by Duffin [8] and from then, it has played an important role in the method that they use to approach the resolution to the problem from the primal perspective. One of the remarkable properties of

geometric programming is that a problem with highly nonlinear constraints can be stated equivalently as one with only linear constraints. This is because there is a strong duality theorem for geometric programming problems. If the primal problem is in the posynomial form then a global minimizing solution to that problem can be obtained by solving the dual maximization. The dual constraints are linear and linearly constrained programs are generally easier to solve than ones with nonlinear constraints. GP problem has a dual impact in the area of integrated circuit design [4, 10, 20], manufacturing system design [3, 8], project management [28], maximization of long run and short run profit [18], generalized geometric programming problem with non-positive variables [34] and goal programming model [1]. Several algorithms due to Beightler and Phillips [2], Fang et al. [9], Kortanek et al. [13], Peterson [22], Rajgopal and Bricker [25] and Jhu and Kortanek [35] strengthen the solution of complicated GPP for the exact known value of the cost and constraint coefficients. Sensitive analysis of various optimal solutions due to Dembo [6], Dinkel and Tretter [7] and Kyparisis [15] using GP technique simplifies certain engineering design problem in which some of the problem parameters are estimating of actual values [2]. Entropy based problem due to Samanta and Mazumder [27] extend the application of GP to variety of entropy based transportation problem. In a recent paper Jana Mazumder [11] applied GP techniques to solve the entropy maximization problem and GP with Fuzzy parameters due to Liu [19] accelerate the work in this direction. As a few works on signomial geometric programming (SGP) have been made, various authors are concentrating to solve such problem. Some of the recent works due to Shen et al. [29] and Shen and Zharg [30] are accelerating to improve the global optimization algorithm for solving certain SGP problems. As the engineering design problems are complex in nature, the generalized GP techniques have been applied to solve such problems. Works on branch and pruning approach on GGP by Shen

and Jiao [31], linearization method of global optimization due to Shen [32] and Qu et al. [24] are some of the recent applications for solving GP problems. The monomial GP with fuzzy relation due to Shivanian and Khorram [33] and maximum likelihood estimation problem proposed by Lim et al. [16] are the most recent works on GPP.

In this paper we restrict attention to posynomial geometric programs and develop a solution procedure by splitting the cost coefficients, constraint coefficients and exponents using binary numbers. This result will provide the decision makers with more information for making better decisions.

The organization of this paper is as follows: following introduction, mathematical formulations and methodology for solving GPP with binary coefficients have been discussed to find the objective value of the problem in section-2. Transformation of the problem into binary model has been discussed in section-3. Dual form of GPP has been discussed in section-4. Some illustrative examples are given in section-5 for understanding the problems and finally in section-6 some conclusions are drawn from the discussion.

2. Mathematical Formulation

A typical constrained posynomial geometric programming problem is presented as follows:

$$Z = \min_x : f_0(x) = \sum_{t=1}^{T_0} C_{0t} \prod_{j=1}^n x_j^{a_{0tj}}$$

subject to

$$f_i(x) = \sum_{t=1}^{T_i} C_{it} \prod_{j=1}^n x_j^{a_{itj}} \leq 1, \\ i = 1, 2, \dots, m \quad (2.1)$$

$$x_j > 0, j = 1, 2, 3, \dots, n.$$

The posynomial $f_0(x)$ is an objective function containing T_0 number of terms where as the posynomial $f_i(x)$, $i = 1, 2, \dots, m$ contains T_i terms with m inequality constraints. By the definition of posynomial all the coefficients C_{it} , $i = 0, 1, 2, \dots, m$ and $t = 1, 2, \dots, T_i$ are positive and the exponents a_{0ij} and a_{ij} are arbitrary constants. Writing the right hand side of the geometric programming problem given by (2.1) in more general form, we have

$$Z = \min_x : f_0(x) = \sum_{t=1}^{T_0} C_{0t} \prod_{j=1}^n x_j^{a_{0tj}}$$

subject to

$$\sum_{t=1}^{T_i} \prod_{j=1}^n x_j^{a_{itj}} \leq 1$$

$$f_i(x) = \sum_{t=1}^{T_i} C_{it} \prod_{j=1}^n x_j^{a_{itj}} \leq b_i$$

$$i = 1, 2, \dots, m \quad (2.2)$$

$$x_j > 0, j = 1, 2, \dots, n.$$

where all b_i are positive real numbers. If $b_i = 1$ for all i then this modified geometric program becomes the original one given by (2.1).

Since b_i in the model (2.2) may not be equal to constant 1 then dividing the constant coefficients C_{it} by $b_i \forall i$ then it is transformed to the standard form

$$Z = \min_x : f_0(x) = \sum_{t=1}^{T_0} C_{0t} \prod_{j=1}^n x_j^{a_{0tj}}$$

subject to

$$f_i(x) = \sum_{t=1}^{T_i} C_{it}(b_i)^{-1} \prod_{j=1}^n x_j^{a_{itj}} \leq 1 \\ i = 1, 2, \dots, m \quad (2.3)$$

$$x_j > 0, j = 1, 2, \dots, n.$$

In this paper our aim is to solve the above problem by splitting the coefficients and exponents of the variables using binary numbers. In order to solve this problem it is necessary to transform the problem to the standard mathematical programming problem using corresponding binary expressions. The following section explains the detail procedure for converting the problem using binary numbers in the different uses.

3. Transformation of the Problem into Binary Model

The proposed GP given by (2.1) and (2.2) can be reformulated by splitting any of C_{0t} , C_{it} , and a_{itj} to the 3 to 8 terms using binary numbers.

Case- 1: When C_{0t} has 3 terms.

$$Z = \min_x : \sum_{t=1}^{T_0} (a_1^t z_1(1-z_2) + a_2^t (1-z_1)z_2 \\ + a_3^t (1-z_1)(1-z_2)) \prod_{j=1}^n x_j^{a_{0tj}}$$

subject to

$$\sum_{t=1}^{T_i} (a_1^t z_1(1-z_2) + a_2^t (1-z_1)z_2 \\ + a_3^t (1-z_1)(1-z_2)) \prod_{j=1}^n x_j^{a_{itj}} \leq 1$$

$$i = 1, 2, \dots, m \quad (3.1)$$

$$z_1 + z_2 \leq 1 \quad (3.2)$$

Case -2: When C_{0t} has 4 terms

$$\min_x : \sum_{t=1}^{T_0}$$

$$Z = \sum_{t=1}^T (a_1^t z_1(1-z_2) + a_2^t (1-z_1)z_2) + a_3^t (1-z_1)(1-z_2) + a_4^t z_1z_2 \prod_{j=1}^n x_j^{a_{otj}}$$

subject to

$$\sum_{t=1}^T (a_1^t z_1(1-z_2) + a_2^t (1-z_1)z_2 + a_3^t (1-z_1)(1-z_2) + a_4^t z_1z_2) \prod_{j=1}^n x_j^{a_{ij}} \leq 1, i = 1, 2, \dots, m \quad (3.3)$$

$$z_1 + z_2 \leq 1 \quad (3.4)$$

Case -3 : When C_{ot} has 5 terms.

$$Z = \min_x : \sum_{t=1}^T (a_1^t z_1(1-z_2)(1-z_3) + a_2^t (1-z_1)z_2(1-z_3) + a_3^t (1-z_1)(1-z_2)z_3 + a_4^t (1-z_1)(1-z_2)(1-z_3) + a_5^t z_1z_2z_3) \prod_{j=1}^n x_j^{a_{otj}}$$

subject to

$$\sum_{t=1}^T (a_1^t z_1(1-z_2)(1-z_3) + a_2^t (1-z_1)z_2(1-z_3) + a_3^t (1-z_1)(1-z_2)z_3 + a_4^t (1-z_1)(1-z_2)(1-z_3) + a_5^t z_1z_2z_3) \prod_{j=1}^n x_j^{a_{ij}} \leq 1, \quad (3.5)$$

$$i = 1, 2, \dots, m \quad (3.6)$$

$$z_1z_2(1-z_3) = 0 \quad (3.7)$$

$$z_2z_3(1-z_1) = 0 \quad (3.8)$$

Case -4 : When C_{ot} has 6 terms.

$$Z = \min_x : \sum_{t=1}^T (a_1^t z_1(1-z_2)(1-z_3) + a_2^t (1-z_1)z_2(1-z_3) + a_3^t (1-z_1)(1-z_2)z_3 + a_4^t z_1z_2(1-z_3) + a_5^t z_1z_3(1-z_2) + a_6^t z_2z_3(1-z_1)) \prod_{j=1}^n x_j^{a_{otj}}$$

subject to

$$\sum_{t=1}^T (a_1^t z_1(1-z_2)(1-z_3) + a_2^t (1-z_1)z_2(1-z_3) + a_3^t (1-z_1)(1-z_2)z_3 + a_4^t z_1z_2(1-z_3) + a_5^t z_1z_3(1-z_2) + a_6^t z_2z_3(1-z_1)) \prod_{j=1}^n x_j^{a_{ij}} \leq 1, \quad (3.9)$$

$$+ a_5^t z_1z_3(1-z_2) + a_6^t z_2z_3(1-z_1) \prod_{j=1}^n x_j^{a_{ij}} \leq 1, i = 1, 2, \dots, m \quad (3.9)$$

$$z_1z_2z_3 = 0 \quad (3.10)$$

$$(1-z_1)(1-z_2)(1-z_3) = 0 \quad (3.11)$$

Case -5 : When C_{ot} has 7 terms

$$Z = \min_x : \sum_{t=1}^T (a_1^t z_1(1-z_2)(1-z_3) + a_2^t (1-z_1)z_2(1-z_3) + a_3^t (1-z_1)(1-z_2)z_3 + a_4^t z_1z_2(1-z_3) + a_5^t z_1z_3(1-z_2) + a_6^t z_2z_3(1-z_1) + a_7^t (1-z_1)(1-z_2)(1-z_3)) \prod_{j=1}^n x_j^{a_{otj}}$$

subject to

$$\sum_{t=1}^T (a_1^t z_1(1-z_2)(1-z_3) + a_2^t (1-z_1)z_2(1-z_3) + a_3^t (1-z_1)(1-z_2)z_3 + a_4^t z_1z_2(1-z_3) + a_5^t z_1z_3(1-z_2) + a_6^t z_2z_3(1-z_1) + a_7^t (1-z_1)(1-z_2)(1-z_3)) \prod_{j=1}^n x_j^{a_{ij}} \leq 1, \quad (3.12)$$

$$i = 1, 2, \dots, m \quad (3.13)$$

$$z_1z_2z_3 = 0 \quad (3.13)$$

Case -6: When C_{ot} has 8 terms.

$$Z = \min_x : \sum_{t=1}^T (a_1^t z_1(1-z_2)(1-z_3) + a_2^t (1-z_1)z_2(1-z_3) + a_3^t (1-z_1)(1-z_2)z_3 + a_4^t z_1z_2(1-z_3) + a_5^t z_1z_3(1-z_2) + a_6^t z_2z_3(1-z_1) + a_7^t (1-z_1)(1-z_2)(1-z_3) + a_8^t z_1z_2z_3) \prod_{j=1}^n x_j^{a_{otj}}$$

$$+ a_8^t z_1z_2z_3) \prod_{j=1}^n x_j^{a_{otj}}$$

subject to :

$$\sum_{t=1}^T (a_1^t z_1(1-z_2)(1-z_3) + a_2^t (1-z_1)z_2(1-z_3) + a_3^t (1-z_1)(1-z_2)z_3 + a_4^t z_1z_2(1-z_3) + a_5^t z_1z_3(1-z_2) + a_6^t z_2z_3(1-z_1) + a_7^t (1-z_1)(1-z_2)(1-z_3) + a_8^t z_1z_2z_3) \prod_{j=1}^n x_j^{a_{ij}} \leq 1, \quad (3.14)$$

$$\begin{aligned}
 &+ a_3^t (1-z_1) (1-z_2) z_3 + a_4^t z_1 z_2 (1-z_3) \\
 &+ a_5^t z_1 z_3 (1-z_2) + a_6^t z_2 z_3 (1-z_1) \\
 &+ a_7^t (1-z_1) (1-z_2) (1-z_3) \\
 &+ a_8^t z_1 z_2 z_3 \prod_{j=1}^n x_j^{a_{otj}} \leq 1, \\
 &i = 1, 2, \dots, m \quad (3.15) \\
 &z_1 z_2 z_3 = 0 \quad (3.16)
 \end{aligned}$$

4. Dual form of GPP

Since model (2.2) is the conventional geometric programming problem then it can be solved directly by using primal based algorithm or dual based algorithm [22]. Methods due to Rajgopal and Bricker [25], Beightler and Phillips [2] and Duffin [8] projected in their analysis that the dual problem has the desirable features of being linearly constrained and having an objective function with attractive structural properties with more suitable solution. According to Duffin [8], Beightler and Phillips [2] one can transform the program of (2.2) into the corresponding dual geometric problem as follows:

$$Z = \max_w \prod_{t=1}^{T_0} \left(\frac{C_{0t}}{w_{0t}} \right)^{w_{0t}} \prod_{i=1}^m \prod_{t=1}^{T_i} \left(\frac{C_{it} b_i^{-1} w_{it}}{w_{it}} \right)^{w_{it}} \prod_{t=1}^{T_i} \lambda(w_{it})^{\lambda(w_{it})} \prod_{t=1}^{T_0} w_{0t} = 1 \quad (4.1)$$

$$\sum_{i=1}^m \sum_{t=1}^{T_i} a_{itj} w_{it} = 0, \quad j = 1, 2, \dots, n. \quad (4.2)$$

$$w_{it} \geq 0 \quad \forall t, i. \quad (4.3)$$

The model (4.1) is the useful dual problem and it can be solved using the method relating to the dual theorem.

5. Illustrative Examples

To illustrate the methodology proposed in this paper for solving a GPP when splitting the cost coefficients, constraint coefficients and the exponents of the decision variables using binary numbers, a few numerical examples are considered.

Example: 1

Let us consider the geometric programming problem which has the following mathematical form:

$$z = \min_x : c x_1^p + \frac{3}{x_2^3} + x_1 x_2 \quad (5.1)$$

$$\begin{aligned}
 \text{subject to} \quad & a x_1 + x_2 \leq 1 \\
 & x_1, x_2 > 0
 \end{aligned}$$

Primal Solution

Case -1: Splitting c, p, a into 3 terms we have

$$\min = c * x_1 \wedge p + 3 / (x_2 \wedge 3) + x_1 * x_2, \quad (5.2)$$

$$a * x_1 + x_2 \leq 1;$$

$$\begin{aligned}
 c &= 5 * (1-z_1) * (1-z_2) + (1-z_1) * z_2 \\
 &+ 3 * z_1 * (1-z_2) \quad (5.3)
 \end{aligned}$$

$$p = -(1-z_3) * (1-z_4) - 3 * (1-z_3) * z_4 - 2 * z_3 * (1-z_4) \quad (5.4)$$

$$a = 3 * (1-z_5) * (1-z_6) + (1-z_5) * z_6 + 2 * z_5 * (1-z_6) \quad (5.5)$$

$$z_1 + z_2 = 1$$

$$z_3 + z_4 = 1$$

$$z_5 + z_6 = 1$$

Case-2: Splitting c, p, a into 4 terms we have

$$\min = c * x_1 \wedge p + 3 / (x_2 \wedge 3) + x_1 * x_2 \quad (5.6)$$

$$a * x_1 + x_2 = 1$$

$$c = 5 * (1-z_1) * (1-z_2) + (1-z_1) * z_2 + 3 * z_1 * (1-z_2) + 4 * z_1 * z_2 \quad (5.7)$$

$$p = -(1-z_3) * (1-z_4) - 3 * (1-z_3) * z_4 - 2 * z_3 * (1-z_4) - 4 * z_3 * z_4 \quad (5.8)$$

$$a = 3 * (1-z_5) * (1-z_6) + (1-z_5) * z_6 + 2 * z_5 * (1-z_6) + 4 * z_5 * z_6 \quad (5.9)$$

$$z_1 + z_2 \leq 1$$

$$z_3 + z_4 \leq 1$$

$$z_5 + z_6 \leq 14$$

Case-3 : Splitting c, p, a into 5 terms we have

$$\min = c * x_1 \wedge p + 3 / (x_2 \wedge 3) + x_1 * x_2 \quad (5.10)$$

$$a * x_1 + x_2 \leq 1$$

$$\begin{aligned}
 c &= 5 * z_1 * (1-z_2) * (1-z_3) + z_2 * (1-z_1) * (1-z_3) + 3 * z_3 * (1-z_1) * (1-z_2) \\
 &+ 4 * (1-z_1) * (1-z_2) * (1-z_3) + 6 * z_1 * z_2 z_3 \quad (5.11)
 \end{aligned}$$

$$\begin{aligned}
 p &= -z_4 * (1-z_5) * (1-z_6) - 3 * z_5 * (1-z_4) * (1-z_6) - 2 * z_6 * (1-z_4) * (1-z_5) \\
 &- (1-z_4) * (1-z_5) * (1-z_6) - 4 * z_4 * z_5 * z_6 \quad (5.12)
 \end{aligned}$$

$$\begin{aligned}
 a &= 3 * z_7 * (1-z_8) * (1-z_9) + z_8 * (1-z_7) * (1-z_9) + 2 * z_9 * (1-z_7) * (1-z_8) \\
 &+ 5 * (1-z_7) * (1-z_8) * (1-z_9) + 6 * z_7 * z_8 * z_9 \quad (5.13)
 \end{aligned}$$

$$z_1 * z_2 * (1-z_3) = 0$$

$$z_2 * z_3 * (1-z_4) = 0$$

$$z_1 * z_3 * (1-z_2) = 0$$

$$z_4 * z_5 * (1-z_6) = 0$$

$$z_5 * z_6 * (1-z_4) = 0;$$

$$z_4 * z_6 * (1-z_5) = 0;$$

$$z_7 * z_8 * (1-z_9) = 0;$$

$$z_8 * z_9 * (1-z_7) = 0;$$

$$z_7 * z_9 * (1-z_8) = 0;$$

Case -4: Splitting c, p, a into 6 terms we have

$$\min = c * x_1 \wedge p + 3 / (x_2 \wedge 3) + x_1 * x_2; \quad (5.14)$$

$$a * x_1 + x_2 \leq 1;$$

$$c=5*z_1*(1-z_2)*(1-z_3)+z_2*(1-z_1)*(1-z_3)+3*z_3*(1-z_1)*(1-z_2)+4*z_1*z_2*(1-z_3)+6*z_1*z_3*(1-z_2)+2*z_2*z_3*(1-z_1); \quad (5.15)$$

$$p=-z_4*(1-z_5)*(1-z_6)-3*z_5*(1-z_4)*(1-z_6)-2*z_6*(1-z_4)*(1-z_5)-z_4*z_5*(1-z_6)-4*z_4*z_6*(1-z_5)-5*z_5*z_6*(1-z_4); \quad (5.16)$$

$$a=3*z_7*(1-z_8)*(1-z_9)+z_8*(1-z_7)*(1-z_9)+2*z_9*(1-z_7)*(1-z_8)+5*z_7*z_8*(1-z_9)+6*z_7*z_9*(1-z_8)+4*z_8*z_9*(1-z_7); \quad (5.17)$$

$$z_1 * z_2 * z_3 = 0;$$

$$(1-z_1) * (1-z_2) * (1-z_3) = 0;$$

$$z_4 * z_5 * z_6 = 0;$$

$$(1-z_4) * (1-z_5) * (1-z_6) = 0;$$

$$z_7 * z_8 * z_9 = 0;$$

$$(1-z_7) * (1-z_8) * (1-z_9) = 0;$$

Case-5: Splitting c, p, a into 7 terms we have

$$\min=c*x_1\wedge p+3/(x_2\wedge 3)+x_1*x_2; \quad (5.18)$$

$$a*x_1+x_2 \leq 1;$$

$$c=5*z_1*(1-z_2)*(1-z_3)+z_2*(1-z_1)*(1-z_3)+3*z_3*(1-z_1)*(1-z_2)+4*z_1*z_2*(1-z_3)+6*z_1*z_3*(1-z_2)+2*z_2*z_3*(1-z_1)+(1-z_1)*(1-z_2)*(1-z_3); \quad (5.19)$$

$$p=-z_4*(1-z_5)*(1-z_6)-3*z_5*(1-z_4)*(1-z_6)-2*z_6*(1-z_4)*(1-z_5)-z_4*z_5*(1-z_6)-4*z_4*z_6*(1-z_5)-5*z_5*z_6*(1-z_4)-(1-z_4)*(1-z_5)*(1-z_6); \quad (5.20)$$

$$a=3*z_7*(1-z_8)*(1-z_9)+z_8*(1-z_7)*(1-z_9)+2*z_9*(1-z_7)*(1-z_8)+5*z_7*z_8*(1-z_9)+6*z_7*z_9*(1-z_8)+4*z_8*z_9*(1-z_7)+4*(1-z_7)*(1-z_8)*(1-z_9); \quad (5.21)$$

$$z_1 * z_2 * z_3 = 0;$$

$$z_4 * z_5 * z_6 = 0;$$

$$z_7 * z_8 * z_9 = 0;$$

Case-6: Splitting c, p, a into 8 terms we have

$$\min=c*x_1\wedge p+3/(x_2\wedge 3)+x_1*x_2; \quad (5.22)$$

$$a * x_1 + x_2 \leq 1;$$

$$c=5*z_1*(1-z_2)*(1-z_3)+z_2*(1-z_1)*(1-z_3)+3*z_3*(1-z_1)*(1-z_2)+4*z_1*z_2*(1-z_3)+6*z_1*z_3*(1-z_2)+2*z_2*z_3*(1-z_1)+(1-z_1)*(1-z_2)*(1-z_3)+2*z_1*z_2*z_3; \quad (5.23)$$

$$p=-z_4*(1-z_5)*(1-z_6)-3*z_5*(1-z_4)*(1-z_6)-2*z_6*(1-z_4)*(1-z_5)-z_4*z_5*(1-z_6)-4*z_4*z_6*(1-z_5)-5*z_5*z_6*(1-z_4)-(1-z_4)*(1-z_5)*(1-z_6)-2*z_4*z_5*z_6; \quad (5.24)$$

$$a=3*z_7*(1-z_8)*(1-z_9)+z_8*(1-z_7)*(1-z_9)+2*z_9*(1-z_7)*(1-z_8)+5*z_7*z_8*(1-z_9)+6*z_7*z_9*(1-z_8)+4*z_8*z_9*(1-z_7)+4*(1-z_7)*(1-z_8)*(1-z_9)+2*z_7*z_8*z_9; \quad (5.25)$$

$$z_1*z_2*z_3=0;$$

$$z_4*z_5*z_6=0;$$

$$z_7*z_8*z_9=0;$$

Using Lingo Software we have found that the solution in each case gives $z=11.01098, c=1, p=-1, a=1, x_1=0.2069792, x_2=0.7930208$.

Dual Solution

$$\max \quad =(1/w_01)\wedge w_01*(3/w_02)\wedge w_02*(1/w_03)\wedge w_03*(1/w_11)\wedge w_11*(1/w_12)\wedge w_12*(w_11+w_12)\wedge (w_11+w_12); \quad (5.26)$$

$$w_01 + w_02 + w_03 = 1;$$

$$-w_01 + w_03 + w_11 = 0;$$

$$-3 * w_02 + w_03 + w_12 = 0;$$

$$w_01, w_02, w_03, w_11, w_12 > 0;$$

In this case we shall find $z = 11.01098, w_01 = 0.4387805, w_02 = 0.5463127, w_03 = 0.1490681E - 01, w_11 = 0.4238737, w_12 = 1.624031$.

Example: 2

Let us consider the geometric programming problem which has the following mathematical form:

$$Z = \min : c x_1 + 10 x_2 + 4 x_3 + 2 x_4 \quad (5.27)$$

$$\begin{aligned} & X \\ \text{subject to} & \quad a x_1^p x_4^{-2} + x_2^2 x_4^{-2} = 1 \\ & \quad 100 x_1^{-1} x_2^{-1} x_3^{-1} = 1 \\ & \quad x_1, x_2, x_3, x_4 > 0 \end{aligned}$$

Primal Solution

Case-1: Splitting c, p, a into 3 terms we have

$$\min=c*x_1+10*x_2+4*x_3+2*x_4; \quad (5.28)$$

$$a*x_1\wedge p*x_4\wedge -2+x_2\wedge 2*x_4\wedge -2 \leq 1;$$

$$100*x_1\wedge -1*x_2\wedge -1+x_3\wedge -1 \leq 1;$$

$$c=5*(1-z_1)*(1-z_2)+(1-z_1)*z_2+3*(1-z_2)*z_1; \quad (5.29)$$

$$p=-(1-z_3)*(1-z_4)-3*(1-z_3)*z_4-2*(1-z_4)*z_3; \quad (5.30)$$

$$a=3*(1-z_5)*(1-z_6)+(1-z_5)*z_6+2*(1-z_6)*z_5; \quad (5.31)$$

$$z_1 + z_2 \leq 1;$$

$$z_3 + z_4 \leq 1;$$

$$z_5 + z_6 \leq 1;$$

Case-2: Splitting c, p, a into 4 terms we have

$$\min=c*x_1+10*x_2+4*x_3+2*x_4; \quad (5.32)$$

$$a*x_1\wedge p*x_4\wedge -2+x_2\wedge 2*x_4\wedge -2 \leq 1;$$

$$100*x_1\wedge -1*x_2\wedge -1*x_3\wedge -1 \leq 1;$$

$$c=5*(1-z_1)*(1-z_2)+(1-z_1)*z_2+3*(1-z_2)*z_1+4*z_1*z_2; \quad (5.33)$$

$$p=-(1-z_3)*(1-z_4)-3*(1-z_3)*z_4-2*(1-z_4)*z_3-4*z_3*z_4; \quad (5.34)$$

$$a=3*(1-z_5)*(1-z_6)+(1-z_5)*z_6+2*(1-z_6)*z_5+4*z_5*z_6; \quad (5.35)$$

$$z_1 + z_2 \leq 1;$$

$$z_3 + z_4 \leq 1;$$

$$z_5 + z_6 \leq 1;$$

Case-3: Splitting c, p, a into 5 terms we have

$$\min=c*x_1+10*x_2+4*x_3+2*x_4; \quad (5.36)$$

$$a*x_1\wedge p*x_4\wedge -2+x_2\wedge 2*x_4\wedge -2 \leq 1;$$

$$100*x_1\wedge -1*x_2\wedge -1*x_3\wedge -1 \leq 1;$$

$$c=5*z_1*(1-z_2)*(1-z_3)+z_2*(1-z_1)*(1-z_3)+3*z_3*(1-z_1)*(1-z_2)+3*(1-z_1)*(1-z_2)*(1-z_3)+5*z_1*z_2*z_3; \quad (5.37)$$

$$p=z_4*(1-z_5)*(1-z_6)-3*z_5*(1-z_4)*(1-z_6)-2*z_6*(1-z_4)*(1-z_5)-3*(1-z_4)*(1-z_5)*(1-z_6)-2*z_4*z_5*z_6; \quad (5.38)$$

$$a=z_7*(1-z_8)*(1-z_9)+z_8*(1-z_7)*(1-z_9)+2*z_9*(1-z_7)*(1-z_8)+4*(1-z_7)*(1-z_8)*(1-z_9)+3*z_7*z_8*z_9; \quad (5.39)$$

$$z_1 * z_2 * (1-z_3) = 0;$$

$$z_2 * z_3 * (1-z_4) = 0;$$

$$z_1 * z_3 * (1-z_2) = 0;$$

$$z_4 * z_5 * (1-z_6) = 0;$$

$$z_5 * z_6 * (1-z_4) = 0;$$

$$z_4 * z_6 * (1-z_5) = 0;$$

$$\begin{aligned} z_7 * z_8 * (1-z_9) &= 0; \\ z_8 * z_9 * (1-z_7) &= 0; \\ z_7 * z_9 * (1-z_8) &= 0; \end{aligned}$$

Case-4: Splitting c, p, a into 6 terms we have

$$\min = c * x_1 + 10 * x_2 + 4 * x_3 + 2 * x_4; \quad (5.40)$$

$$a * x_1 \wedge p * x_4 \wedge -2 + x_2 \wedge 2 * x_4 \wedge -2 \leq 1;$$

$$100 * x_1 \wedge -1 * x_2 \wedge -1 * x_3 \wedge -1 \leq 1;$$

$$c = 5 * z_1 * (1-z_2) * (1-z_3) + z_2 * (1-z_1) * (1-z_3) + 3 * z_3 * (1-z_1) * (1-z_2) + 3 * z_1 * z_2 * (1-z_3) + z_1 * z_3 * (1-z_2) + z_2 * z_3 * (1-z_1); \quad (5.41)$$

$$p = z_4 * (1-z_5) * (1-z_6) - 3 * z_5 * (1-z_4) * (1-z_6) - 2 * z_6 * (1-z_4) * (1-z_5) - 2 * z_4 * z_5 * (1-z_6) - 3 * z_4 * z_6 * (1-z_5) - z_5 * z_6 * (1-z_4); \quad (5.42)$$

$$a = z_7 * (1-z_8) * (1-z_9) + z_8 * (1-z_7) * (1-z_9) + 2 * z_9 * (1-z_7) * (1-z_8) + 4 * z_7 * z_8 * (1-z_9) + 3 * z_7 * z_9 * (1-z_8) + 4 * z_8 * z_9 * (1-z_7); \quad (5.43)$$

$$\begin{aligned} z_1 * z_2 * z_3 &= 0; \\ (1-z_1) * (1-z_2) * (1-z_3) &= 0; \\ z_4 * z_5 * z_6 &= 0; \\ (1-z_4) * (1-z_5) * (1-z_6) &= 0; \\ z_7 * z_8 * z_9 &= 0; \\ (1-z_7) * (1-z_8) * (1-z_9) &= 0; \end{aligned}$$

Case-5: Splitting c, p, a into 7 terms we have

$$\min = c * x_1 + 10 * x_2 + 4 * x_3 + 2 * x_4; \quad (5.44)$$

$$a * x_1 \wedge p * x_4 \wedge -2 + x_2 \wedge 2 * x_4 \wedge -2 \leq 1;$$

$$100 * x_1 \wedge -1 * x_2 \wedge -1 * x_3 \wedge -1 \leq 1;$$

$$c = 5 * z_1 * (1-z_2) * (1-z_3) + z_2 * (1-z_1) * (1-z_3) + 3 * z_3 * (1-z_1) * (1-z_2) + 3 * z_1 * z_2 * (1-z_3) + 5 * z_1 * z_3 * (1-z_2) + 2 * z_2 * z_3 * (1-z_1) + (1-z_1) * (1-z_1) * (1-z_2) * (1-z_3); \quad (5.45)$$

$$p = -z_4 * (1-z_5) * (1-z_6) - 3 * z_5 * (1-z_4) * (1-z_6) - 2 * z_6 * (1-z_4) * (1-z_5) - 3 * z_4 * z_5 * (1-z_6) - 2 * z_4 * z_6 * (1-z_5) - z_5 * z_6 * (1-z_4) - (1-z_4) * (1-z_5) * (1-z_6); \quad (5.46)$$

$$a = z_7 * (1-z_8) * (1-z_9) + z_8 * (1-z_7) * (1-z_9) + 2 * z_9 * (1-z_7) * (1-z_8) + 4 * z_7 * z_8 * (1-z_9) + 3 * z_7 * z_9 * (1-z_8) + 4 * z_8 * z_9 * (1-z_7) + 4 * (1-z_7) * (1-z_8) * (1-z_9); \quad (5.47)$$

$$\begin{aligned} z_1 * z_2 * z_3 &= 0; \\ z_4 * z_5 * z_6 &= 0; \\ z_7 * z_8 * z_9 &= 0; \end{aligned}$$

Case-6: Splitting c, p, a into 8 terms we have

$$\min = c * x_1 + 10 * x_2 + 4 * x_3 + 2 * x_4; \quad (5.48)$$

$$a * x_1 \wedge p * x_4 \wedge -2 + x_2 \wedge 2 * x_4 \wedge -2 \leq 1;$$

$$100 * x_1 \wedge -1 * x_2 \wedge -1 * x_3 \wedge -1 \leq 1;$$

$$c = 5 * z_1 * (1-z_2) * (1-z_3) + z_2 * (1-z_1) * (1-z_3) + 3 * z_3 * (1-z_1) * (1-z_2) + 3 * z_1 * z_2 * (1-z_3) + 5 * z_1 * z_3 * (1-z_2) + 2 * z_2 * z_3 * (1-z_1) + (1-z_1) * (1-z_2) * (1-z_3) + z_1 * z_2 * z_3; \quad (5.49)$$

$$p = -z_4 * (1-z_5) * (1-z_6) - 3 * z_5 * (1-z_4) * (1-z_6) - 2 * z_6 * (1-z_4) * (1-z_5) - 3 * z_4 * z_5 * (1-z_6) - 2 * z_4 * z_6 * (1-z_5) - z_5 * z_6 * (1-z_4) - (1-z_4) * (1-z_5) * (1-z_6) - 2 * z_4 * z_5 * z_6; \quad (5.50)$$

$$a = z_7 * (1-z_8) * (1-z_9) + z_8 * (1-z_7) * (1-z_9) + 2 * z_9 * (1-z_7) * (1-z_8) + 4 * z_7 * z_8 * (1-z_9) + 3 * z_7 * z_9 * (1-z_8) + 4 * z_8 * z_9 * (1-z_7) + 4 * (1-z_7) * (1-z_8) * (1-z_9) + 2 * z_7 * z_8 * z_9; \quad (5.51)$$

$$\begin{aligned} z_1 * z_2 * z_3 &= 0; \\ z_4 * z_5 * z_6 &= 0; \\ z_7 * z_8 * z_9 &= 0; \end{aligned}$$

Using Lingo Software we have found that the solution in each case gives $z = 50.60611$, $c = 1$, $a = 1$, $p = -3$, $x_1 = 16.86890$, $x_2 = 1.405717$, $x_3 = 4.217114$, $x_4 = 1.405791$.

Dual Solution:

$$\max = (1/w_{01}) \wedge w_{01} * (10/w_{02}) \wedge w_{02} * (4/w_{03}) \wedge w_{03} * (2/w_{04}) \wedge w_{04} * (1/w_{11}) \wedge w_{11} * (1/w_{12}) \wedge w_{12} * (w_{11} + w_{12}) \wedge (w_{11} + w_{12}) * 100 \wedge w_{21}; \quad (5.52)$$

$$w_{01} + w_{02} + w_{03} + w_{04} = 1;$$

$$w_{01} + (-3) * w_{11} - w_{21} = 0;$$

$$w_{02} + 2 * w_{12} - w_{21} = 0;$$

$$w_{03} - w_{21} = 0;$$

$$w_{04} - 2 * w_{11} - 2 * w_{12} = 0;$$

$$w_{01}, w_{02}, w_{03}, w_{04}, w_{11}, w_{12}, w_{21} > 0;$$

In this case we find $z = 50.60611$, $w_{01} = 0.3333372$, $w_{02} = 0.2777762$, $w_{03} = 0.3333285$, $w_{04} = 0.5555811E-01$, $w_{11} = 0.2903339E-05$, $w_{12} = 0.2777615E-01$, $w_{21} = 0.3333285$.

6. Conclusions

Since 1960 geometric programming problem has undergone several changes. In most of the engineering problems the parameters are considered as deterministic. In this paper we have discussed the problems by splitting the cost coefficients, constraint coefficients and exponents using binary numbers. Geometric programming has already shown its power in practice in the past. In many real world geometric programming problem the parameters may not be known precisely due to insufficient information and hence this paper will help the wider applications in the field of engineering problems.

References

- [1] Aouni .B and Kettani.O: Goal programming model: A Glorious history and a promising future. European Journal of Operation Research 133, 225-231, 2001.
- [2] Beightler C.S and Phillips D.T: Applied geometric Programming, John Wiley and Son, New York, 1976.
- [3] Choi.J.C and Bricker D.L: Effectiveness of a geometric Programming algorithm for optimization of machining Economics models. Computers and Operation Research 10, 1495-1522, 1977.
- [4] Chu C. and Wong D.F: VLSI circuit performance optimization by geometric programming. Annals of Operations Research 105, 37-60, 2001.
- [5] Clasen R: The linear algorithmic problem. Rand Corp. Memo RM 37-7,PR, 1963.
- [6] Dembo.R.S: The sensitivity of optimal engineering designs using geometric programming. Engineering optimization 5, 27-40, 1980.
- [7] Dinkel J. J. and Tretter M. J: An interval arithmetic approach to sensitivity analysis in geometric programming. Operations Research 35, 866-869.1987.
- [8] Duffin.R.J, Peterson E. L and Zener C. M: Programming theory and application .Weily, New York, 1967.
- [9] Fang.S.C, Peterson.E.L and Rajasekera. J.R: Controlled dual perturbations for posynomial programs. European Research 35, 111-17, 1988.

- [10] Hershenson.M.D, Boyd S.P and Lee T.H: Optimal design of a CMOS op-amp via geometric programming. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems* 20, 1-21, 2001.
- [11] Jana P. and Mazumder S. K: Geometric programming and dual entropy optimization problem. *Applied Mathematics and Computation* 193, 340-345,2007.
- [12] Kantorovich L. V., *Mathematical methods for organizing and planning production in Russia. English Translation in Management Science* 6, 366-422, 1960.
- [13] Kortanek K. O., NO, H. , A second order affine scaling algorithm for the geometric programming dual with logarithmic barrier. *Optimization* 23, 303-322,1992.
- [14] Kortanek K. O, Xu. X and Ye Y: An infeasible interior point algorithm for solving primal and dual geometric programs. *Mathematical programming* 76, 155-181, 1997.
- [15] Kyparisis J: Sensitivity analysis in geometric programming: Theory and Computation. *Annals of Operations Research*, 27, 39-64,1990.
- [16] Lim.J, Wang X. and Choi W: Maximum likelihood estimation of ordered multinomial probabilities by geometric programming. *Computational Statistics and Data Analysis* 53, 889-893,2009.
- [17] Liu S.T: A computational method for the maximization of long-run and short-run profit. *Applied Mathematics and Computation* 186, 1104-1112,2007.
- [18] Liu.S.T: Posynomial geometric programming with interval exponents and coefficient. *European Journal of Operations Research*,2007.
- [19] Liu.S.T: Geometric programming with fuzzy parameters in engineering optimization. *International Journal of Approximate Reasoning* 46, 484-498,2007.
- [20] Mandal P.P. and Visvanathan V: FCMOS Op-ampsizing using a geometric programming function, *IEEE Transaction on computer-Aided Design of Integrated Circuits and Systems* 20, 22-38,2001.
- [21] Passy. U: Considering generalized poly -nomials. *Journal of Optimization Theory and Applications*9 (4), 221-237,1972
- [22] Peterson.E.L: The fundamental relations between geometric programming duality, parametric programming duality. *Annals of Operations Research* 105, 109-153,2001.
- [23] Prasad A. V, S. R. K., Rao P.N. and Rao U.R.K: Optimal selection of process parameters for turning operations in CAPP system. *International Journal of Production Research* 35, 1495-1522,1997.
- [24] Qu.S,Zhang.K and Wang.F: A global optimization using linear relaxation for generalized geometric programming. *European journal of Operational Research*,190,345-356,2008.
- [25] Rajgopal J. and Bricker D.L: Solving posynomial geometric programming problems via generalized linear programming. *Computational Optimization and Applications* 21, 95-109,2002.
- [26] Rijckaert.M.J and Martens.X.M: A condensation method for generalized geometric programming, *Mathematical programming* 11, 89-93, 1976.
- [27] Samanta B. and Mazumder S. K: Entropy based transportation problem use geometric programming method. *Applied Mathematics and Computation* 179,431-439,2006.
- [28] Scott C. H. and Jefferson T.R: Allocation of resources in project management. *International Journal of Systems Science* 26, 413-420, 1995.
- [29] Shen P. P, Li X and Jiao H.W: Accelerating method of global optimization for signomial geometric programming.

Journal of Computational and Applied Mathematics , 214, 66-77, 2008.

[30] Shen P. P and Zharg K.C: Global optimization of signomial geometric programming using relaxation. *Applied Math. Comput.* 150, 99-114,2004.

[31] Shen P. and Jiao H: A new rectangle branch and pruning approach for generalized geometric programming. *Applied Mathematics and Computation* 183, 1027-1038, 2006.

[32] Shen P: Linearization method of global optimization for generalized geometric programming. *Applied Mathematics and Computation.*162, 353-370,2005.

[33] Shivanian E. and Khorram E: Monomial geometric programming with fuzzy relation inequality constraints with Max-product composition. *Computers and Industrial Engineering, Elsevier (Journal homepage)*, 2008.

[34] Tsai J. F, Lin M. H. and Hu Y.C: On generalized geometric programming problems with non-positive variables. *European Journal of Operations Research*, 2006.

[35] Zhu J, Kortanek K. O. and Huang S: Controlled dual perturbations for central path trajectories in geometric programming. *European Journal of Operations Research* 73, 524-531,1992.

Dr.A.K.Ojha: Dr A.K.Ojha received a Ph.D (Mathematics) from Utkal University in 1997. Currently he is an Asst. Prof. in Mathematics at I.I.T Bhubaneswar, India. He is performing research in Neural Network, Genetical Algorithm, Geometric Programming and Particle Swarm Optimization. He is served more than more than 27 years in different Govt. colleges in the state of Orissa. He is published 22 research paper in different journals and 7 books for degree students such as: Fortran 77 Programming, A text book of Modern Algebra, Fundamentals of Numerical Analysis etc.

A.K.Das: Mr.A.K.Das received a M.Sc (Mathematics) from Ravenshaw College. Currently he is an Lecturer in Mathematics at Bhadrak College, Bhadrak, Orissa, India. He is performing research in Geometric Programming and Optimization Theory.