American J. of Engineering and Applied Sciences 2 (2): 481-487, 2009
ISSN 1941-7020

An Original Geometric Programming Problem Algorithm to Solve Two Coefficients Sensitivity Analysis

[1]Abbas Y. Al-Bayati and [2]Huda E. Khalid
[1]Department of Computer Sciences and Mathematics, Mosul University, Mosul, Iraq
[2]Department of Computer Engineering, Technical College of Mosul, Mosul, Iraq

Abstract: Problem statement: It has been noticed by Dinkel and Kochenberger that they developed sensitivity procedure for Posynomial Geometric Programming Problems based on making a small changes in one coefficient. **Approach:** This study presented an original algorithm for finding the ranging analysis while studying the effect of perturbations in the original coefficients without resolving the problem, this proposed procedure had been trapped on two coefficients simultaneously. We also had developed one of the incremental strategies to make suitable comparisons. **Results:** Comparison results had been done between the gained result from the sensitivity analysis approach and the incremental analysis approach. **Conclusion:** In the standard Geometric Programming Problem, we obtained an original algorithm, for the first time, by changing two coefficients simultaneously in the objective function.

Key words: Geometric programming, sensitivity analysis, ranging analysis, incremental analysis

INTRODUCTION

This study deals with the sensitivity analysis in the case of less than type inequalities. Techniques designed to study the effects of small changes in the input variables on the optimal solution of an optimization problems, with out having to solve the entire problem again and again, are known in the literature as sensitivity analysis techniques[1]. Dinkel and Kochenberger studying the effect of changing coefficients separately on the optimal solution[2,4].

MATERIALS AND METHODS

The mathematical formulation of the sensitivity analysis for posynomials (polynomials with positive coefficients) are discussed in the research of Dinkel *et al*.[3] as follow:

Theorem 1: Suppose that the primal geometric program has d>0 and rank $(a_{ij}) = m$ If the solution to the dual geometric program has $\delta^*>0$ and if the Jacobian matrix $J(\delta)$with components is:

$$J_{ij}(\delta) = \sum\nolimits_{q=1}^{n} \frac{b_q^{(t)} b_q^{(j)}}{\delta_q} - \sum\nolimits_{k=1}^{p} \frac{\lambda_k^{(t)} \lambda_k^{(J)}}{\lambda_k(\delta)} i, j = 1,...,d \qquad (1)$$

nonsingular at δ^*, then the functions which give the optimized parameters δ^* and $\nu(\delta^*)$ in terms of the variable coefficient vector c are differentiable on an open neighborhood of c. These differentials are:

$$\frac{d\nu}{\nu^*} = \sum_{i=1}^{n} \delta_i^* \frac{dc_i}{c_i} \qquad (2)$$

$$d\delta_i = \sum\nolimits_{j=1}^{d} \left\{ b_i(j) \sum\nolimits_{k=1}^{d} \left[J_{jk}^{-1}(\delta^*) \sum\nolimits_{i=1}^{n} b_i(k) \frac{dc_i}{c_i} \right] \right\} \qquad (3)$$

$i = 1,...., n$

where, $J_{jk}^{-1}(\delta^*)$ represents the components of the inverse of $J(\delta)$ and:

$$d\lambda_k = \sum\nolimits_{i=mk}^{nk} d\delta_i , k = 1,..., p \qquad (4)$$

The major restriction of this result, from an applications point of view, is that are no inactive primal constraints at the optimal solution $(\delta_i^* > 0 \text{ for all } i)$ .Thus assuming the problem has been reformulated, if necessary , to meet this restriction.

For differential changes dc_i that maintain the positivity conditions on all dual variables, the new dual solution is estimated as:

$$\delta_i^{*'} = \delta_i^* + d\delta_i , i = 1,..., n \qquad (5)$$

$$\nu^{*'} = \nu^* + \nu^* \sum_{i=1}^{n} \delta_i^* \frac{dc_i}{c_i} \qquad (6)$$

Corresponding Author: Abbas Y. Al-Bayati, Department of Computer Sciences and Mathematics, Mosul University, Mosul, Iraq

where, $d\delta_i$ is given by (3). Once the dual solution is known the estimate of the new primal solution is computed as:

$$c_i x_1^{a_{i1}} x_2^{a_{i2}} \dots x_m^{a_{im}} = \begin{cases} \delta_i^* v(\delta^*) & i \in [0] \\ \delta_i^* / \lambda_k(\delta^*) & i \in [k] \end{cases} \quad (7)$$

and

$$(\log x_j^{*'}) = [(a_{ij})^T (a_{ij})]^{-1} (a_{ij})^T (K_i) \quad (8)$$

Where:

$$k_i = \begin{cases} \log \delta_j^{*'} + \log v(\delta^{*'}) - \log(c_i + dc_i) \\ i = 1, \dots, n_o \\ \log \delta_j^{*'} - \log \lambda_k^{*'} - \log(c_i + dc_i) \\ i = n_o + 1, \dots, n \end{cases} \quad (9)$$

and n_o is the number of terms in the primal objective function.

If the sub-matrix $b_i(j) = 1, \dots, n_o$, $j = I, \dots, d$, has rank d then $J(\delta)$ is nonsingular for each $\delta > 0$.

Theorem 2: Suppose the primal GPP has $d>0$ and let $b(j)$, $j = 1, \dots, m$ are linearly independent. If the sub-matrix with components $b_i(j)$, $I = 1, \dots, n_o$ and $j = 1, \dots d$ has rank d then $j(\delta)$, given by (1), is nonsingular for each $\delta > 0$. Since we are interested in other than differential changes we will interpret $\frac{dv}{v}$ and $\frac{dc_i}{c_i}$ as rates of change[3]. That is:

$$\frac{dv}{v} = \frac{v^{*'} - v^*}{v^*} \quad (10)$$

$$\frac{dc_i}{c_i} = \frac{c_i' - c_i}{c_i} \quad (11)$$

where, v' and c_i' denote the new values of the objective function and coefficients respectively.

An original GPP algorithm: Before we make some observations of the new original procedure, let us consider the outlines of this algorithm:

Step 1: Put $\delta_i + d\delta_i = 0$ as an equations of the two variables Δ_1 and Δ_2 where $i = 1, 2 \dots, n$ is the number of dual variables.

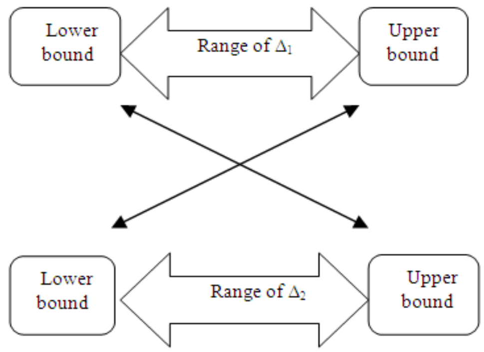

Fig. 1: Cross-shape figure

Step 2: Calculate the cofactors of Δ_1 and Δ_2 in those equations obtained in the step 1, we note that the sign of Δ_1 is the opposite to the sign of Δ_2 for each $i = 1, 2 \dots, n$.

Step 3: Categorized those equations in two groups:

- The first group is containing the +ive cofactors of Δ_1 and the –ive cofactors of Δ_2
- The second group is containing the –ive cofactors of Δ_1 and the +ive cofactors of Δ_2

Step 4: From the first group, calculate the lower bound of Δ_1 and the upper bound of Δ_2, while the upper of Δ_1 and the lower bound of Δ_2 will be calculated from the 2nd group.

Step 5: Since our searching is concerned about the range of any two coefficients in the objective function by changing them simultaneously so any small change in the lower bound of Δ_1 will effect on the upper bound of Δ_2 similarly, upper bound of Δ_1 and lower bound of Δ_2 will be effected, this connection gives us an ability to construct the cross-shape in Fig. 1.

Step 6: Find the intersection points of $\delta_i + d\delta_i = 0$ with Δ_1 and Δ_2 axis.

Step 7: Determine the pieces of the those lines between the intersection points and study all points at that pieces to find the most important answer on the following most important question:

At which point on the pieces of the 1st and 2nd groups will we find maxΔ_1 with minΔ_2 simultaneously?

Step 8: After finding those points, apply the following rule:

- The upper bound on Δ_1 is then the minimum of $\Delta_1 > 0$ for those i when (14)<0 for which (13) is satisfied. If $\Delta_1<0$ evaluating (13) for those I for which (14)>0 then the lower bound on Δ_1 being the maximum such Δ_I[1], by regarding the observations (a), (b) and (c) in Note 2

Step 9: End.

Some theoretical observations:

Note 1:

- If we attempt to change the upper bounds of Δ_1 and Δ_2 simultaneously or the lower boundsΔ_1 and Δ_2 this will shift the cross-shape right or left respectively. The important thing now, because we have consider the change in two coefficients, this yields two dimensional space for which Δ_1 is the horizontal axis and Δ_2 is the vertical axis .The equations $\delta_i+d\delta_i = 0$ are straight lines in Δ_1 and Δ_2 plane

Note 2:

(a) We suggest that the lower bound on Δ_1 don't exceed the negative value of c_1 to maintain the posynomial nature. Also for Δ_2

(b) We make the same steps on the bounds of Δ_2 with replacing (A) by (B)

(c) At changing in c_1 and c_2 simultaneously we must note that this changing is with respect (the cases if $\Delta_1>0 \rightarrow \Delta_2>0$ and $\Delta_1<0 \rightarrow \Delta_2<0$ are out of our ranges since it is contradict the conditions in the problems)

Note 3:

- The above algorithm is originally designed by us with a numerical evidence we put those results in Table 1-4 which are verified by using our programs writing in Matlab
- If we try to change three coefficients simultaneously, this required to study three dimensional space and this is not the domain of our research in this research but it is a good field to study in future

Table 1: The effect of the sensitivity analysis in 9 problems

I	(A)	(B)
1	0.012624766	-0.012624766
2	-0.012624766	0.012624766
3	0.052363195	-0.052363195
4	-0.052363195	0.052363195
5	0.039272396	-0.039272396
6	-0.052363195	0.052363195
7	-0.002029759	0.002029759
8	-0.018698897	0.018698897
9	-0.005918973	0.005918973

Table 2: Maximum and minimum changes in 9 problems

I	Δ_1	Δ_2
1	-83.892382270	3329233.990000
2	11.911373650	-472697.864300
3	-15.462960660	613641.083500
4	7.751342565	-307608.766300
5	-15.462960660	613641.087500
6	7.635325135	-303004.663300
7	21.496974180	-853094.403500
8	21.496974180	-853098.403500
9	1.056374340	-4192.177244

Table 3: Allowable ranges in the sensitivity analysis for changing in c_1 and c_2 simultaneously

	$c_2' = 0.419$	$c_2' = 0.419$	$c_2' = 0$	$c_2' = 3.47537434$	$c_2' = 3.419$	$c_2' = 3.419$	$c_1' = 0.581$	$c_2' = 4.419$
Dual variable	$c_2' = 1195997$	$c_2' = 91997$	$c_2' = 12227282.17$	$c_2' = 91804,822$	$c_2' = 99997$	$c_2' = 99997$	$c_1' = 1245997$	$c_2' = 90997$
δ_1^*	0.720567740	0.86575708	0.714266600	0.88173357	0.88141408	0.88036199	0.7087731	0.88676459
δ_2^*	0.279443226	0.13424920	0.285733390	0.11826643	0.11858592	0.11963801	0.2912269	0.11323540
δ_3^*	0.026135030	0.62833053	1.005022700	0.69459555	0.69327041	0.68890667	-0.02278494	0.71546250
δ_4^*	0.978887740	0.37669224	1.005022700	0.31042722	0.31175236	0.31611609	1.02780772	0.28956027
δ_4^*	0.019601270	0.47124789	0.999999930	0.52094666	0.51995281	0.51668000	-0.01708871	0.53659686
δ_5^*	0.973864970	0.37166947	0.062025560	0.30540445	0.30672959	0.31109333	1.02278494	0.28453749
δ_6^*	0.061012490	0.03766942	0.062025560	0.03510077	0.03515214	0.03532129	0.06290879	0.03429189
δ_7^*	0.562067060	0.34702305	0.571399860	0.323359810	0.32383303	0.32539131	0.57953638	0.31590822
δ_8^*	0.077886920	0.00981671	0.080841130	{Not allowable	0.00247612	0.00296938	0.08341667	-0.00003239
			{Not allowable	since $\delta_9^* = 0$}			{Not allowable}	{Not allowable}
			since $\delta_3^* = \delta_5^* = 0$}					

Table 4: Allowable ranges in the sensitivity analysis and increment analysis for changing in c_1 and c_2 simultaneous

Dual variable	C1 = 0.01092549664174 C2 = 1.98907450335826 Sensitivity analysis	C1 = 0.01092549664174 C2 = 1.98907450335826 Increment analysis	C1 = 0.5534959906535 C2 = 1.44650400934649 Sensitivity analysis	C1 = 0.5534959906535 C2 = 1.44650400934649 Increment analysis	C1 = 0.1 C2 = 1.9 Sensitivity analysis	C1 = -1 C2 = 1 Sensitivity analysis	C1 = -1 C2 = 1 Sensitivity analysis	C1 = -1 C2 = 1 Increment analysis
δ_1^*	0.57324699633083	0.38599011429509	0.86890197867858	0.28971437867349	0.59159174061681	0.48497242931080	0.57099690865665	0.36942007548640
δ_2^*	0.42675300366917	0.61400988570491	0.13109802132142	0.71028562132651	0.40840825938319	0.51502757068920	0.42900309134335	0.63057992451360
δ_3^*	0.13349054427694	0.24416928708291	0.06525799179181	0.29342482843379	0.12925686098397	0.19512609389964	0.13400982634819	0.25256707117627
δ_4^*	0.33702897400910	0.67942180780158	-0.00000000000000	0.84378174399595	0.31299736329592	0.51454582589908	0.33624254943152	0.70749528278640
δ_5^*	057324699633083	0.38599011429509	0.86890197867858	0.28971437867349	0.59159174061681	0.48497242931080	0.57099690865665	0.36942007548640
δ_6^*	0.00000000000000	0.00845459440089	0.14221703003489	0.00349107343137	0.00882425531404	0.01683442979603	-0.00108234531952	0.00743631805382
δ_7^*	0.28661924200770	0.18452264364437	0.29223579843941	0.14135637128838	0.28696790792462	0.22564594540339	0.28657668068146	0.17726556918633
δ_8^*	0.33440663034118	0.34089955855327	0.15249480842802	0.35656420153804	0.32311940010716	0.32222071317365	0.33579107360347	0.34372388482617
δ_9^*	0.30572287516258	0.58011527000410	0.04415222355412	0.70886423045173	0.28949298896763	0.45032064286015	0.30771356341156	0.60214600194424
δ_{10}^*	0.21919054316968	0.32349066748720	0.12804144830696	0.36814278031812	0.21353494141429	0.27683821249629	0.20117234285314	0.33123284006549
δ_{11}^*	0.20023408123380	0.44376968413946	0.07694923074260	0.55391518337738	0.19258452620639	0.33629195167420	0.33624254943152	0.46241960303047
δ_{12}^*	0.33370289740091	0.67942180780158	-0.00000000000000 {Not allowable}	0.84378174399595	0.31299736329592	0.51454582589908	{Not allowable}	0.70749528278640

RESULTS

Example 1: Consider the following GPP with degree of difficulty two:

$$\min g_o(x) = 2.419x_1x_2x_3 + 95997x_1^{-1.8}x_3x_4^{-4.8}$$

Subject to:

$$28867x_1^{-0.875}x_2^{-,75}x_3^{-1}x_5^{-,75} \le 1$$
$$25819x_1^{-0.2}x_3^{-1}x_4^{,8}x_6^{-1} \le 1$$
$$0.03866x_5 + 0.03866x_6 \le 1$$
$$0.0081666x_2^{-1} + x_2^{-1}x_4 \le 1$$
$$0.0834x_2^{-1} \le 1 x_i > 0 \quad i = 1,2,3,4,5,6$$

Here the degree of difficulty is d = 9-6-1 = 2.
The dual objective function is:

$$z(\delta) = \delta_1 \log\frac{c_1}{\delta_1} + \delta_2 \log\frac{c_2}{\delta_2} + \delta_3 \log c_3 + \delta_4 \log c_4$$
$$+ \delta_5 \log\frac{c_5.\lambda_3}{\delta_5} + \delta_6 \log\frac{c_6.\lambda_3}{\delta_6} \delta_7 \log\frac{c_7.\lambda_4}{\delta_7}$$
$$+ \delta_8 \log\frac{c_8.\lambda_4}{\delta_8} + \delta_9 \log c_9$$

First this system can be solved by:

"NLPSolve" Maple function gives following results:

>with (optimization):

$$\mathrm{NLPsolve}\left(\frac{(1-\delta_2)\,.\mathrm{In}\left(\frac{2.419}{1-\delta_2}\right)}{\mathrm{In}(10)} + \frac{\delta_2\,.\mathrm{In}\left(\frac{95997}{\delta_2}\right)}{\mathrm{In}(10)}\right)$$

$$+\frac{\delta_3.\mathrm{In}(288670)}{\mathrm{In}(10)} + \frac{(1-\delta_3).\mathrm{In}(25819)}{\mathrm{In}(10)}$$

$$> + \frac{\delta_5.\mathrm{In}\left(\frac{0.03866(\delta 5+\delta_6}{\delta_5}\right)}{\mathrm{In}(10)} + \frac{\delta_6.\mathrm{In}\left(\frac{0.03866(\delta_5+\delta_6}{\delta 6}\right)}{\mathrm{In}(10)}$$

$$+\frac{\delta_7.\mathrm{In}\left(\frac{0.0081666(\delta_7+\delta_8}{\delta_7}\right)}{\mathrm{In}(10)} + \frac{\delta_8.\mathrm{In}\left(\frac{(\delta_7+\delta_8}{\delta_8}\right)}{\mathrm{In}(10)}$$

$$-1.078833949\delta_9, \{2.8\,.\delta_2 + 0.675.\delta_3$$
$$= 0.8, \delta_2 + 0.75.\delta_3 + \delta_7 + \delta_8 + \delta_9 = 1, -4.8\delta_2 - 0.8\delta_3$$
$$+\delta_8 = -0.8, -0.75.\delta_3 + \delta_5 = 0, \delta_3 + \delta_6 = 1\},$$
$$\text{assume} = \text{nonnegative}, \text{max imize})$$

[5.26927101441275526,
[δ_8 = 0.332342173499054206,
δ_5 = 0.502081435651985375,
δ_6 = 0.330558085797352685,
δ_7 = 0.360758067560167442,
δ_2 = 0.124330967111861854,
δ_3 = 0.6694419142026204,
δ_9 = 0.00516961698108152346]]

Suppose c_1 change to $c_1+\Delta_1$ and c_2 change $c_2+\Delta_2$ and consider $0 = \delta^*_1$ $0=\delta_i^{'}\equiv\delta_i^*+d\delta_i$ fore:

$$\begin{aligned} d\delta_i &= b_i(1)\left\{\begin{array}{l} J_{11}^{-1}\left[b_1(1)\dfrac{\Delta_1}{c_1}+b_2(1)\dfrac{\Delta_2}{c_2}\right]+ \\ J_{12}^{-1}\left[b_1(2)\dfrac{\Delta_1}{c_1}+b_2(2)\dfrac{\Delta_2}{c_2}\right] \end{array}\right\} \\ &+ b_i(2)\left\{\begin{array}{l} J_{21}^{-1}\left[b_1(1)\dfrac{\Delta_1}{c_1}+b_2(1)\dfrac{\Delta_2}{c_2}\right]+ \\ J_{22}^{-1}\left[b_1(2)\dfrac{\Delta_1}{c_1}+b_2(2)\dfrac{\Delta_2}{c_2}\right] \end{array}\right\} \end{aligned} \quad (12)$$

We have:

$$\begin{aligned} 0 = \delta_i^* &\left(\left[\begin{array}{l} b_i(1)\left(b_1(1)J_{11}^{-1}+b_1(2)J_{12}^{-1}\right)+ \\ b_i(2)\left(b_1(1)J_{21}^{-1}+b_1(2)J_{22}^{-1}\right) \end{array}\right]\frac{\Delta_1}{c_1}\right) \\ &+\left(\left[\begin{array}{l} b_i(1)\left(b_2(1)J_{11}^{-1}+b_2(2)J_{12}^{-1}\right)+ \\ b_i(2)\left(b_2(1)J_{21}^{-1}+b_2(2)J_{22}^{-1}\right) \end{array}\right]\frac{\Delta_2}{c_2}\right) \end{aligned} \quad (13)$$

Let:

$$\begin{aligned} A = {} & [b_i(1)(b_1(1)J_{11}^{-1}+b_1(2)J_{12}^{-1})+b_i(2) \\ & +(b_i(2)(b_i(1)J_{21}^{-1}+b_1(2)J_{22}^{-1}] \end{aligned} \quad (14)$$

$$\begin{aligned} B = {} & [b_i(1)(b_2(1)J_{11}^{-1}+b_2(2)J_{12}^{-1})+b_i(2) \\ & +(b_2(2)(b_2(1)J_{21}^{-1}+b_2(2)J_{22}^{-1}] \end{aligned} \quad (15)$$

Evaluating (A) and (B) for i = 1, 2,….9 we will get Table 1.

Substitute these values in (13) and solve the following nine optimization problems:

Problem 1 (for i = 1):

max$\Delta_1$which is the minΔ_2
s.t.
$0.005219002067\Delta_1-0.0000001315120889\Delta_2$
$+0.8756690329 = 0$
with $\Delta_1<0$; $\Delta_2<0$

Problem 2 (for i = 2):

minΔ_1 which is the maxΔ_2
$-0.005219002067\Delta_1+0.0000001315120889\Delta_2$
$+0.1243309671 = 0$
with $\Delta_1>0$; $\Delta_2<0$

Problem 3 (for i = 3):

max$\Delta_1$which is the minΔ_2
s.t.
$0.021646628\Delta_1-0.000000545669938$
$+0.6694419142 = 0$
with $\Delta_1<0$; $\Delta_2>0$

Problem 4 (for i = 4):

minΔ_1 which is the maxΔ_2
s.t.
$-0.021646628\Delta_1$
$+0.0000005454669938\Delta_2+0.3355808580 = 0$
with $\Delta_1>0$; $\Delta_2<0$

Problem 5 (for i = 5):

maxΔ_1 which is the minΔ_2
s.t.
$0.016234971\Delta_1$
$-0.0000004091002427\Delta_2+0.502081435 = 0$
with $\Delta_1<0$; $\Delta_2>0$

Problem 6 (for i = 6):

minΔ_1 which is the MaxΔ_2
s.t.
$-0.021646628\Delta_1+0.000000545669938\Delta_2$
$+0.3305580857 = 0$
with $\Delta_1>0$; $\Delta_2<0$

Problem 7 (for i = 7):

minΔ_1 which is the maxΔ_2
s.t.
$-0.00083909034\Delta_1+0.0000000211439892\Delta_2$
$+0.036075807 = 0$
with $\Delta_1>0$; $\Delta_2<0$

Problem 8 (for i = 8):

minΔ_1 which is the maxΔ_2
s.t.
$-0.007730011162\Delta_1+0.007730011162\Delta_2$
$+0.3323421734 = 0$
with $\Delta_1>0$; $\Delta_2<0$

Problem 9 (for i = 9):

minΔ_1 which is the maxΔ_2
s.t.
$-0.00244686793\Delta_1+0.00000061657901\Delta_2$
$+0.00516961698 = 0$
with $\Delta_1>0$; $\Delta_2<0$

The solutions of these problems can be tabulated as Table 2.

Depending on the step 8 of the original algorithm we conclude that:

- Lower bound of $\Delta_1 = \max\ \{\Delta_{1i},\ i = 1,3,5\} = -15.46296066$
- Upper bound of $\Delta_1 = \min\ \{\Delta_{1i},\ i = 2,4,6,7,8,9\} = 1.05637434$
- Lower bound of $\Delta_1 = \max\ \{\Delta_{2i},\ i = 2,4,6,7,8,9\} = -4192.17724$
- Upper bound of $\Delta_2 = \min\ \{\Delta_{2i},\ i = 1,2,3,5\} = -61361.0835$

This implies that:

$-15.46296066 < \Delta_1 < 1.05637434$
$-4192.177244 < \Delta_2 < 613641.0835$

To satisfy the Note 2 part (a) we must observe that this means that $\Delta_1 = c_1^{'} - c_1 \rightarrow c_1^{'} = \Delta_1 + c_1 > 0$ and this occur at $\Delta_1 > -2.419$; therefore we replace the lower bound of Δ_1 from -15.46296066 to -2.419, but we realize this value will effect on the upper bound of Δ_2. We evaluate Δ_2 at the constraints in the problems 1,3 and 5 mentioned above and we select the minimum value of Δ_2 (facing to Δ_1 = -2.419) this implies that:

$-2.419 < \Delta_1 < 1.056374334 \rightarrow 0 > c_1^{'} < 0.347537434$
$-4192.177244 < \Delta_1 < 1131285.17 \rightarrow$
$91804.822p76 < c_2^{'} < 122728.17$

This will give the upper and lower bounds of the coefficient $c_2^{'}$.

Example 2: $\min g_0(x) = x_1 + x_2^{1.5}$

Subject to:

$$0.1(x_4^{pi} + x_5)x_6^{-1}x_7^{-1} \le 1 x_3^{0.5}x_1^{-1} \le 1$$
$$(x_6 + x_7^{-1})x_3^{-1} \le 1(x_7 + x_8^{-2.9})x_2^{-1} \le 1$$
$$x_6^2 x_8^2 x_5^{-1} \le 1$$
$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 > 0$$

Here d = 12-8-1 = 3.

To find $b_m(0)$ set $\delta_1 = \delta_4 = \delta_9 = 0$ in the linear equations of the dual constraints, $b_m(1)$ can be obtained similarly by setting $\delta_1 = 1$, $\delta_4 = \delta_9 = 0$ $b_m(2)$ is the value obtained by substituting $\delta_1 = 0$, $\delta_4 = 1$, $\delta_9 = 0$. Finally $b_m(3)$ will be obtained by substituting $\delta_1 = 0$, $\delta_4 = 0$, $\delta_9 = 1$ where m = 1,2,….12. So we get:

$$bm(0) = \begin{bmatrix} 0 \\ 1 \\ -0.2386 \\ 0 \\ 0 \\ -0.9887 \\ 0.9887 \\ 1.5 \\ 0 \\ -1.499 \\ -0.7499 \\ 0 \end{bmatrix} bm(1) = \begin{bmatrix} 0 \\ -1 \\ -0.31818 \\ 0 \\ 1 \\ 1.3182 \\ -0.8181 \\ 1.5 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} bm(2) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \\ -2 \\ 2 \\ 1 \end{bmatrix} bm(3) = \begin{bmatrix} 0 \\ 0 \\ 0.6205 \\ 0 \\ 0 \\ -0.3295 \\ 0.3295 \\ -1 \\ 1 \\ 2.9 \\ -0.9499 \\ 0 \end{bmatrix}$$

$$\therefore z(\delta) = \delta_1 \log\frac{c_1}{\delta_1} + \delta_2 \log\frac{c_2}{\delta_2} + \delta_3 \log\frac{c_3.\lambda_1}{\delta_3}$$
$$+ \delta_4 \log\frac{c_4.\lambda_1}{\delta_4} + \delta_5 \log c_5 + \delta_6 \log\frac{c_6.\lambda_3}{\delta_6}$$
$$+ \delta_7 \log\frac{c_7.\lambda_3}{\delta_7} + \delta_8 \log\frac{c_8.\lambda_4}{\delta_8} + \delta_9 \log\frac{c_9.\lambda_4}{\delta_9}$$
$$+ \delta_{10} \log\frac{c_{10}.\lambda_5}{\delta_{10}} + \delta_{11} \log\frac{c_{11}.\lambda_5}{\delta_{11}} + \delta_{12} \log c_{12}$$

The value of the objective function can be evaluated as follow:

> with (optimization):

$$\mathrm{NLPsolve}\left(\frac{\delta_1.\mathrm{In}\left(\frac{1}{\delta_1}\right)}{\mathrm{In}(10)} + \frac{(1-\delta_1).\mathrm{In}\left(\frac{1}{1-\delta_1}\right)}{\mathrm{In}(10)} \right)$$
$$+ \frac{\delta_3.\mathrm{In}\left(\frac{0.1(\delta_3+\delta_4}{\delta_3}\right)}{\mathrm{In}(10)} + \frac{\delta_4.\mathrm{In}\left(\frac{0.1+\delta_7}{\delta_7}\right)}{\mathrm{In}(10)} + \frac{\delta_6.\mathrm{In}\left(\frac{\delta_6+\delta_7}{\delta_6}\right)}{\mathrm{In}(10)}$$
$$> + \frac{\delta_7.\mathrm{In}\left(\frac{\delta_6+\delta_7}{\delta_7}\right)}{\mathrm{In}(10)} + \frac{\delta_8.\mathrm{In}\left(\frac{\delta_8+\delta_9}{\delta_8}\right)}{\mathrm{In}(10)} + \frac{\delta_9.\mathrm{In}\left(\frac{\delta_8+\delta_9}{\delta_9}\right)}{\mathrm{In}(10)}$$
$$+ \frac{\delta_{10}.\mathrm{In}\left(\frac{\delta_{10}+\delta_{11}}{\delta_{10}}\right)}{\mathrm{In}(10)} + \frac{\delta 11.\mathrm{In}\left(\frac{\delta_{10}+\delta_{11}}{\delta_{11}}\right)}{\mathrm{In}(10)}, \{1, 5.(1-\delta_1)$$
$$-\delta_8 - \delta_9 = 0\ \pi\delta_3 - \delta_{10} - \delta_{11} = 0, -\delta_3 - \delta_4 - \delta_7 + \delta_8$$
$$+\delta_{10} + \delta_{11} = 0\ 0.5.\delta_1 - \delta_6 - \delta_7 = 0,\ \delta_3 + \delta_4 +$$
$$\delta_6 - \delta_{11} = 0, -2.9.\delta_9 + \delta_{10} + 2.\ \delta_4 = 0\},$$
$$\mathrm{assume} = \mathrm{nonnegative}, \max\mathrm{imize})$$

[0.36901663595333999,8
[δ_1 = 0.776945228258277830

δ_3 = 0.15639129886847561,0
δ_{11} = 0.11529417638566357
δ_6 = 0.086480172705934083,0
δ_4 = 0.1037906688075505330
δ_6 = 0.09798366106092626,8
δ_7 = 0.290488953113046455
δ_8 = 0.2907433864.034649,6
δ_9 = 0125507818972236702]

But $\delta_2 = 1-\delta_1$, $\delta_5 = \delta_1$ and $\delta_{12} = \delta_4$.
Again by applying the algorithm we will get:

$-0.98907450335862 < \Delta_1 < 0.446504009$
$0.0109259664174 < c_1^{'} < 1.4465$
$-0.446504009 < \Delta_2 < c_1^{'} < 0.98907450335862$
$0.55495990635 < c_1^{'} < 1.9890745033586$

We have developed the formula of the increment analysis for single coefficient [1] to multiple coefficients as follow:

$$d\delta_i^{k_1} = \sum_{j=1}^{d} b_i(j) \sum_{k=1}^{d} \sum_{l=1}^{n} J_{jk}^{-1}(\delta^{k_1-1}) b_l(k) \frac{\Delta_l}{c_l k_l + \Delta_l(k_l-1)} \qquad (16)$$

d = The degree of difficulty
n = Number of coefficients that we will change

Finally, we tested the efficiency of our formula by making Matlab program and fettered the results by Table 4.

DISCUSSION

In this study, we have proposed an original algorithm associated to the geometric programming problem (GPP) and signomial programming problem (SPP) by changing two coefficients in their objective function simultaneously to study the effect of ranging analysis of these algorithms without resolving the algorithms again. The original algorithm given in this study has been proved both theoretically and numerically by using high degree of difficulty test problems.

CONCLUSION

This study deals with geometric programming problem where exponent matrix a_{ij} is of full rank, the degree of difficulty is greater than zero and the constraints at the case of less than inequalities, but we made the changes in two coefficients of the objective function simultaneously. In the given examples we show in Table 3 and 4 containing numerical results to test the effectiveness of our original algorithm.

REFERENCES

1. Al-Bayati, A.Y. and C. Mohan, 1980. A numerical procedure for sensitivity analysis of generalized geometric programming problems. Eng. Optimiz., 5: 51-58. DOI: 10.1080/03052158008902433
2. Dinkel, J.J. and G.A. Kochenberger, 1977. On sensitivity analysis in geometric programming. Operat. Res., 25: 155-163. http://www.jstor.org/stable/169555
3. Dinkel, J.J., G.A. Kochenberger and S.N. Wong, 1978. Sensitivity analysis procedures for geometric programs: Computational aspects. ACM. Trans. Math. Software, 4: 1-14. http://portal.acm.org/citation.cfm?id=355770
4. Dinkel, J.J. and G.A. Kochenberger, 1974. A note on substitution effects in geometric programming. Manage. Sci., 20: 1141-1143. http://www.jstor.org/stable/2629745