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

1Abbas Y. Al-Bayati and 2Huda E. Khalid
1Department of Computer Sciences and Mathematics, Mosul University, Mosul, Iraq
2Department 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 \( (\mathbf{a}_0) = m \) If the solution to the dual geometric program has \( \delta^* > 0 \) and if the Jacobian matrix \( J(\delta) \) with components is:

\[
 J_i(\delta) = \sum_{k=1}^{n} \frac{b_k \partial h_i(\delta)}{\partial \delta_k} - \sum_{k=1}^{p} \frac{\lambda_k \partial h_i(\delta)}{\partial \lambda_k(\delta)} , i = 1, ..., d
\]

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

\[
 \frac{dv}{\nu} = \sum_{i=1}^{e} \delta_i \frac{dc_i}{c_i} \tag{2}
\]

\[
 d\delta_i = \sum_{j=1}^{d} \left[ b_j(i) \sum_{k=1}^{n} \left[ J_{ik}(\delta^*) \sum_{l=1}^{e} b_l(k) \frac{dc_l}{c_l} \right] \right] \tag{3}
\]

where, \( J_{ik}(\delta^*) \) represents the components of the inverse of \( J(\delta) \) and:

\[
 d\lambda_k = \sum_{j=1}^{n} b_j(k) \delta_j, k = 1, ..., p \tag{4}
\]

The major restriction of this result, from an applications point of view, is that are no inactive primal constraints at the optimal solution \( \delta^* > 0 \) 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:

\[
 \delta_i^* = \delta_i^* + d\delta_i, i = 1, ..., n \tag{5}
\]

\[
 \nu^* = \nu^* + \nu \sum_{i=1}^{e} \delta_i \frac{dc_i}{c_i} \tag{6}
\]
where, \( d\delta_i \) is given by (3). Once the dual solution is known the estimate of the new primal solution is computed as:

\[
c_i x_i^\ast x_2^\ast \ldots x_m^\ast = \begin{cases} 
\delta_i^\ast v(\delta^\ast) & i \in [0] \\
\delta_i^\ast / \lambda_i(\delta^\ast) & i \in [k] 
\end{cases}
\]

and

\[
(\log x_i^\ast) = [(a_i)^T(a_i)]^{-1}(a_i)^T(K_i)
\]

Where:

\[
k_i = \begin{cases} 
\log \delta_i^0 + \log v(\delta^\ast) - \log(c_i + dc_i) & i = 1, \ldots, n_0 \\
\log \delta_i^0 - \log \lambda_i^0 - \log(c_i + dc_i) & i = n_0 + 1, \ldots, n
\end{cases}
\]

and \( n_0 \) is the number of terms in the primal objective function.

If the sub-matrix \( b_i(j) = 1, \ldots, n_0 \), \( j = 1, \ldots, d \), has rank \( d \) then \( J(\delta) \) is nonsingular for each \( \delta > 0 \).

**Theorem 2:** Suppose the primal GPP has \( d > 0 \) and let \( b(j), j = 1, \ldots, m \) are linearly independent. If the sub-matrix with components \( b_i(j), i = 1, \ldots, n_0 \) and \( j = 1, \ldots, d \) 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 \( \Delta_1 \) and \( \Delta_2 \) where \( i = 1, 2, \ldots, n \) is the number of dual variables.

**Step 2:** Calculate the cofactors of \( \Delta_1 \) and \( \Delta_2 \) in those equations obtained in the step 1, we note that the sign of \( \Delta_1 \) is the opposite to the sign of \( \Delta_2 \) for each \( i = 1, 2, \ldots, n \).

**Step 3:** Categorized those equations in two groups:

- The first group is containing the +ive cofactors of \( \Delta_1 \) and the −ive cofactors of \( \Delta_2 \)
- The second group is containing the −ive cofactors of \( \Delta_1 \) and the +ive cofactors of \( \Delta_2 \)

**Step 4:** From the first group, calculate the lower bound of \( \Delta_1 \) and the upper bound of \( \Delta_2 \), while the upper of \( \Delta_1 \) and the lower bound of \( \Delta_2 \) 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 \( \Delta_1 \) will effect on the upper bound of \( \Delta_2 \) similarly, upper bound of \( \Delta_1 \) and lower bound of \( \Delta_2 \) will be effected, this connection gives us an ability to construct the cross-shape in Fig. 1.

**Step 6:** Find the intersection points of \( d\delta_i = 0 \) with \( \Delta_1 \) and \( \Delta_2 \) 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 max\( \Delta_1 \) with min\( \Delta_2 \) simultaneously?

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

- The upper bound on \( \Delta_1 \) 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 $\Delta_1$ being the maximum such $\Delta_{1[1]}^*$, 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 $\Delta_1$ and $\Delta_2$ simultaneously or the lower bounds $\Delta_1$ and $\Delta_2$ 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 $\Delta_1$ is the horizontal axis and $\Delta_2$ is the vertical axis. The equations $\delta_1 + \delta_0 = 0$ are straight lines in $\Delta_1$ and $\Delta_2$ plane.

**Note 2:**

(a) We suggest that the lower bound on $\Delta_1$ don’t exceed the negative value of $c_1$ to maintain the posynomial nature. Also for $\Delta_2$

(b) We make the same steps on the bounds of $\Delta_2$ 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

<table>
<thead>
<tr>
<th>I</th>
<th>(A)</th>
<th>(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.012624766</td>
<td>-0.012624766</td>
</tr>
<tr>
<td>2</td>
<td>-0.012624766</td>
<td>0.012624766</td>
</tr>
<tr>
<td>3</td>
<td>0.052363195</td>
<td>-0.052363195</td>
</tr>
<tr>
<td>4</td>
<td>-0.052363195</td>
<td>0.052363195</td>
</tr>
<tr>
<td>5</td>
<td>0.039272396</td>
<td>-0.039272396</td>
</tr>
<tr>
<td>6</td>
<td>-0.052363195</td>
<td>0.052363195</td>
</tr>
<tr>
<td>7</td>
<td>-0.00209759</td>
<td>0.00209759</td>
</tr>
<tr>
<td>8</td>
<td>-0.018698897</td>
<td>0.018698897</td>
</tr>
<tr>
<td>9</td>
<td>-0.005918973</td>
<td>0.005918973</td>
</tr>
</tbody>
</table>

**Table 2:** Maximum and minimum changes in 9 problems

<table>
<thead>
<tr>
<th>I</th>
<th>$\Delta_1$</th>
<th>$\Delta_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-83.89238227</td>
<td>3329233.9900</td>
</tr>
<tr>
<td>2</td>
<td>11.91137365</td>
<td>-472697.8643</td>
</tr>
<tr>
<td>3</td>
<td>-15.46296066</td>
<td>613641.0875</td>
</tr>
<tr>
<td>4</td>
<td>7.75134256</td>
<td>-307608.7663</td>
</tr>
<tr>
<td>5</td>
<td>-15.46296066</td>
<td>613641.0875</td>
</tr>
<tr>
<td>6</td>
<td>7.65325135</td>
<td>-303004.6633</td>
</tr>
<tr>
<td>7</td>
<td>21.496974180</td>
<td>-855094.4035</td>
</tr>
<tr>
<td>8</td>
<td>21.496974180</td>
<td>-855094.4035</td>
</tr>
<tr>
<td>9</td>
<td>1.056374340</td>
<td>-4192.172244</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$c_i$</th>
<th>$c_j$</th>
<th>$c_k$</th>
<th>$c_l$</th>
<th>$c_m$</th>
<th>$c_n$</th>
<th>$c_o$</th>
<th>$c_p$</th>
<th>$c_q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 $c_i$</td>
<td>0.419</td>
<td>0.419</td>
<td>0.419</td>
<td>0.419</td>
<td>0.419</td>
<td>0.419</td>
<td>0.419</td>
<td>0.419</td>
</tr>
<tr>
<td>2 $c_j$</td>
<td>0.91997</td>
<td>0.91997</td>
<td>0.91997</td>
<td>0.91997</td>
<td>0.91997</td>
<td>0.91997</td>
<td>0.91997</td>
<td>0.91997</td>
</tr>
<tr>
<td>3 $c_k$</td>
<td>1.222728217</td>
<td>1.222728217</td>
<td>1.222728217</td>
<td>1.222728217</td>
<td>1.222728217</td>
<td>1.222728217</td>
<td>1.222728217</td>
<td>1.222728217</td>
</tr>
<tr>
<td>4 $c_l$</td>
<td>0.81904822</td>
<td>0.81904822</td>
<td>0.81904822</td>
<td>0.81904822</td>
<td>0.81904822</td>
<td>0.81904822</td>
<td>0.81904822</td>
<td>0.81904822</td>
</tr>
<tr>
<td>5 $c_m$</td>
<td>0.0599997</td>
<td>0.0599997</td>
<td>0.0599997</td>
<td>0.0599997</td>
<td>0.0599997</td>
<td>0.0599997</td>
<td>0.0599997</td>
<td>0.0599997</td>
</tr>
<tr>
<td>6 $c_n$</td>
<td>0.1245997</td>
<td>0.1245997</td>
<td>0.1245997</td>
<td>0.1245997</td>
<td>0.1245997</td>
<td>0.1245997</td>
<td>0.1245997</td>
<td>0.1245997</td>
</tr>
<tr>
<td>7 $c_o$</td>
<td>0.90997</td>
<td>0.90997</td>
<td>0.90997</td>
<td>0.90997</td>
<td>0.90997</td>
<td>0.90997</td>
<td>0.90997</td>
<td>0.90997</td>
</tr>
</tbody>
</table>

Dual variable $c_i = 1195997$, $c_j = 0.8657508$, $c_k = 0.7142660$, $c_l = 0.8817335$, $c_m = 0.8814108$, $c_n = 0.88036199$, $c_o = 0.7087731$, $c_p = 0.38676459$.

<table>
<thead>
<tr>
<th>$\delta_i^\prime$</th>
<th>$\delta_j^\prime$</th>
<th>$\delta_k^\prime$</th>
<th>$\delta_l^\prime$</th>
<th>$\delta_m^\prime$</th>
<th>$\delta_n^\prime$</th>
<th>$\delta_o^\prime$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.27943226</td>
<td>0.13429920</td>
<td>0.28573390</td>
<td>0.11826643</td>
<td>0.11858592</td>
<td>0.11963801</td>
<td>0.2912269</td>
</tr>
<tr>
<td>0.02613030</td>
<td>0.62833053</td>
<td>1.00502270</td>
<td>0.69495555</td>
<td>0.69327041</td>
<td>0.68890667</td>
<td>-0.02278494</td>
</tr>
<tr>
<td>0.097887740</td>
<td>0.37669224</td>
<td>1.00502270</td>
<td>0.31047227</td>
<td>0.3175236</td>
<td>0.3161609</td>
<td>1.0278072</td>
</tr>
<tr>
<td>0.019601270</td>
<td>0.47124789</td>
<td>0.99999930</td>
<td>0.52094666</td>
<td>0.51995281</td>
<td>0.51668000</td>
<td>-0.01708871</td>
</tr>
<tr>
<td>0.97386497</td>
<td>0.37669479</td>
<td>0.60205560</td>
<td>0.30540445</td>
<td>0.30672959</td>
<td>0.31103933</td>
<td>1.02278494</td>
</tr>
<tr>
<td>0.061012490</td>
<td>0.03766942</td>
<td>0.06202560</td>
<td>0.03510077</td>
<td>0.03515214</td>
<td>0.03532129</td>
<td>0.06290879</td>
</tr>
<tr>
<td>0.562067060</td>
<td>0.34702305</td>
<td>0.57199980</td>
<td>0.32339810</td>
<td>0.32383303</td>
<td>0.32593131</td>
<td>0.57953638</td>
</tr>
<tr>
<td>0.37886920</td>
<td>0.00981671</td>
<td>0.08084130</td>
<td>[Not allowable]</td>
<td>0.00247612</td>
<td>0.00296938</td>
<td>0.08341667</td>
</tr>
</tbody>
</table>

[Not allowable] since $\delta_{i^\prime} = 0$

[Not allowable] since $\delta_{i^\prime} = 0$

[Not allowable] since $\delta_{i^\prime} = 0$

483
### RESULTS

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

\[
\min g_c(x) = 2.419x_1x_2x_3 + 95997x_4^{-13}x_4^{-14}
\]

Subject to:

\[
\begin{align*}
28867x_1^{-0.875}x_2^{-1.75}x_3^{-1.75} & \leq 1, \\
25819x_1^{-2}x_2x_3 & \leq 1, \\
0.03866x_4 + 0.03866x_6 & \leq 1, \\
0.008166x_4^{-1} + x_4^{-1} & \leq 1, \\
0.0834x_4^{-1} & \leq 1, x_i > 0, i = 1, 2, 3, 4, 5, 6
\end{align*}
\]

Here the degree of difficulty is \(d = 9-6-1 = 2\).

The dual objective function is:

\[
z(\delta) = \delta_1 \log \frac{c_1}{\delta_1} + \delta_2 \log \frac{c_2}{\delta_2} + \delta_3 \log c_3 + \delta_4 \log c_4
\]

\[
+ \delta_5 \log \frac{c_1 \lambda_1}{\delta_5} + \delta_6 \log \frac{c_2 \lambda_2}{\delta_6} + \delta_7 \log \frac{c_3 \lambda_3}{\delta_7}
\]

First this system can be solved by:

"NLP Solve" Maple function gives following results:

\[
\begin{align*}
\lambda_1 &= 0.57324699633083, \\
\lambda_2 &= 0.38599101429599, \\
\lambda_3 &= 0.86891978675858, \\
\lambda_4 &= 0.28971437867349
\end{align*}
\]

\[
\begin{align*}
\delta_1 &= 0.59159174061681, \\
\delta_2 &= 0.48497242931080, \\
\delta_3 &= 0.57099608366566, \\
\delta_4 &= 0.36942007548640
\end{align*}
\]

\[
\begin{align*}
\delta_5 &= 0.00882425514040, \\
\delta_6 &= 0.16483429786038, \\
\delta_7 &= 0.00010823451952, \\
\delta_8 &= 0.00742631805382
\end{align*}
\]

\[
\begin{align*}
\delta_9 &= 0.12217030034899, \\
\delta_{10} &= 0.00349107343137, \\
\delta_{11} &= 0.00000000000000, \\
\delta_{12} &= 0.00000000000000
\end{align*}
\]

\[
\begin{align*}
\delta_1 + \delta_2 + \delta_3 + \delta_4 &= -0.008166. \\
\delta_1 + \delta_2 + \delta_3 + \delta_4 &= -0.008166. \\
\delta_1 + \delta_2 + \delta_3 + \delta_4 &= -0.008166. \\
\delta_1 + \delta_2 + \delta_3 + \delta_4 &= -0.008166.
\end{align*}
\]

\[
\begin{align*}
\delta_5 + \delta_6 + \delta_7 + \delta_8 &= 0.00516961698108, \\
\delta_5 + \delta_6 + \delta_7 + \delta_8 &= 0.50208143565198, \\
\delta_5 + \delta_6 + \delta_7 + \delta_8 &= 0.332342173499054236, \\
\delta_5 + \delta_6 + \delta_7 + \delta_8 &= 0.37699294513160
\end{align*}
\]

\[
\begin{align*}
\delta_9 &= 0.15249480842802, \\
\delta_{10} &= 0.91309821231242, \\
\delta_{11} &= 0.00349107343137, \\
\delta_{12} &= 0.00000000000000
\end{align*}
\]
Suppose \( c_1 \) change to \( c_1 + \Delta_1 \) and \( c_2 \) change \( c_2 + \Delta_2 \) and consider \( 0 = \delta_1^* = 0 = \delta_1 = \delta_i^* + d\delta_i \) for:

\[
\delta_i = b_i(1) \begin{cases} 
J_{1i}^1 \left[ b_i(1) \frac{\Delta_1}{c_i} + b_i(2) \frac{\Delta_2}{c_2} \right] + \\
J_{1i}^2 \left[ b_i(2) \frac{\Delta_1}{c_i} + b_i(2) \frac{\Delta_2}{c_2} \right] 
\end{cases} \\
+ b_i(2) \begin{cases} 
J_{2i}^1 \left[ b_i(1) \frac{\Delta_1}{c_i} + b_i(2) \frac{\Delta_2}{c_2} \right] + \\
J_{2i}^2 \left[ b_i(2) \frac{\Delta_1}{c_i} + b_i(2) \frac{\Delta_2}{c_2} \right] 
\end{cases}
\]

(12)

We have:

\[
0 = \delta_i^* = \delta_1^* = \delta_1 = \delta_i = \delta_i^* + d\delta_i
\]

(13)

Let:

\[
A = \left[ b_i(1) \left( b_i(1) J_{1i}^{11} + b_i(2) J_{1i}^{12} \right) + b_i(2) \right]
\]

(14)

\[
B = \left[ b_i(1) \left( b_i(2) J_{1i}^{21} + b_i(2) J_{1i}^{22} \right) + b_i(2) \right]
\]

(15)

Evaluating (A) and (B) for \( i = 1, 2, \ldots, 9 \) we will get Table 1. Substitute these values in (13) and solve the following nine optimization problems:

**Problem 1 (for \( i = 1 \)):**

\[
\max \Delta_1 \text{ which is the min } \Delta_2 \\
\text{s.t.} \quad 0.005219002067\Delta_1 - 0.0000001315120889\Delta_2 + 0.8756690329 = 0 \\
\text{with } \Delta_1 < 0; \Delta_2 > 0
\]

**Problem 2 (for \( i = 2 \)):**

\[
\min \Delta_1 \text{ which is the max } \Delta_2 \\
\text{s.t.} \quad -0.005219002067\Delta_1 + 0.0000001315120889\Delta_2 + 0.1243309671 = 0 \\
\text{with } \Delta_1 > 0; \Delta_2 < 0
\]

**Problem 3 (for \( i = 3 \)):**

\[
\max \Delta_1 \text{ which is the min } \Delta_2 \\
\text{s.t.} \quad 0.021646628\Delta_1 - 0.000000545669938\Delta_2 + 0.6694449142 = 0 \\
\text{with } \Delta_1 < 0; \Delta_2 > 0
\]

**Problem 4 (for \( i = 4 \)):**

\[
\min \Delta_1 \text{ which is the max } \Delta_2 \\
\text{s.t.} \quad -0.021646628\Delta_1 + 0.0000005454669938\Delta_2 + 0.3355808580 = 0 \\
\text{with } \Delta_1 > 0; \Delta_2 < 0
\]

**Problem 5 (for \( i = 5 \)):**

\[
\max \Delta_1 \text{ which is the min } \Delta_2 \\
\text{s.t.} \quad 0.016234971\Delta_1 - 0.0000004091002427\Delta_2 + 0.502081435 = 0 \\
\text{with } \Delta_1 < 0; \Delta_2 > 0
\]

**Problem 6 (for \( i = 6 \)):**

\[
\min \Delta_1 \text{ which is the max } \Delta_2 \\
\text{s.t.} \quad -0.021646628\Delta_1 + 0.000000545669938\Delta_2 + 0.3305580857 = 0 \\
\text{with } \Delta_1 > 0; \Delta_2 < 0
\]

**Problem 7 (for \( i = 7 \)):**

\[
\min \Delta_1 \text{ which is the max } \Delta_2 \\
\text{s.t.} \quad -0.00083909034\Delta_1 + 0.0000000211439892\Delta_2 + 0.036075807 = 0 \\
\text{with } \Delta_1 > 0; \Delta_2 < 0
\]

**Problem 8 (for \( i = 8 \)):**

\[
\min \Delta_1 \text{ which is the max } \Delta_2 \\
\text{s.t.} \quad -0.007730011162\Delta_1 + 0.000000211439892\Delta_2 + 0.3323421734 = 0 \\
\text{with } \Delta_1 > 0; \Delta_2 < 0
\]

**Problem 9 (for \( i = 9 \)):**

\[
\min \Delta_1 \text{ which is the max } \Delta_2 \\
\text{s.t.} \quad -0.00244686793\Delta_1 + 0.00000061657901\Delta_2 + 0.00516961698 = 0 \\
\text{with } \Delta_1 > 0; \Delta_2 < 0
\]
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 \( \Delta_2 = \min \{ \Delta_{2i}, \ i = 2,4,6,7,8,9 \} = -4192,177244 \)
- 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 \( \Delta_i = c_i - c_i \rightarrow c_i = \Delta_i + c_i > 0 \) and this occur at \( \Delta_i > -2,419 \); therefore we replace the lower bound of \( \Delta_1 \) from -15,46296066 to -2,419, but we realize this value will effect on the upper bound of \( \Delta_2 \). We evaluate \( \Delta_2 \) at the constraints in the problems 1,3 and 5 mentioned above and we select the minimum value of \( \Delta_2 \) (facing to \( \Delta_1 = -2,419 \)) this implies that:

\[-2,419 < \Delta_2 < 1,05637434 \rightarrow 0 > c_i < 0,347537434 \]
\[-4192,177244 < \Delta_1 < 1131285,17 \rightarrow 91804,822p76 < c_i < 122728,17 \]

This will give the upper and lower bounds of the coefficient \( c_i \).

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

Subject to:

\[0.1(x_2^9 + x_4)x_7^{-1} - 1 \leq x_3^{1.5}x_4^{-1} \leq 1 \]
\[(x_6 + x_7)x_5^{-1} \leq 1(x_7 + x_8^{-2.9})x_2^{-1} \leq 1 \]
\[x_7x_8x_9^{-1} \leq 1 \]
\[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 > 0 \]

Here \( d = 12 - 8 - 1 = 3 \).

To find \( b_m(0) \) set \( \delta_1 = \delta_2 = \delta_3 = 0 \) in the linear equations of the dual constraints, \( b_m(1) \) can be obtained similarly by setting \( \delta_1 = 1, \delta_2 = \delta_3 = 0 \) \( b_m(2) \) is the value obtained by substituting \( \delta_1 = 0, \delta_2 = 1, \delta_3 = 0 \). Finally \( b_m(3) \) will be obtained by substituting \( \delta_1 = 0, \delta_2 = 0, \delta_3 = 1 \) where \( m = 1,2,\ldots,12 \). So we get:

\[
\begin{bmatrix}
0 \\
1 \\
-0.2386 \\
0 \\
0 \\
0 \\
0 \\
-0.9887 \\
0.9887 \\
-1.5 \\
0 \\
0 \\
0 \\
-1.499 \\
-0.7499 \\
0
\end{bmatrix}
\begin{bmatrix}
1 \\
-1 \\
0 \\
0 \\
1 \\
0 \\
0 \\
1.3182 \\
-0.8181 \\
1.5 \\
0 \\
2 \\
0 \\
0 \\
0 \\
1 \\
2
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
0 \\
0.9887 \\
-0.9887 \\
-1.5 \\
0 \\
-1.499 \\
-0.7499 \\
0 \\
1 \\
2 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0.3295 \\
-1 \\
0 \\
2.9 \\
-0.9499 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

\[
\begin{aligned}
\Delta_1 &= \delta_1 \log \frac{c_1}{\delta_1} + \delta_2 \log \frac{c_2}{\delta_2} + \delta_3 \log \frac{c_3}{\delta_3} \\
&+ \delta_1 \log \frac{c_4}{\delta_1} + \delta_2 \log c_5 + \delta_3 \log \frac{c_6}{\delta_3} \\
&+ \delta_1 \log \frac{c_7}{\delta_1} + \delta_2 \log \frac{c_8}{\delta_2} + \delta_3 \log \frac{c_9}{\delta_3} \\
&+ \delta_1 \log \frac{c_{10}}{\delta_1} + \delta_2 \log \frac{c_{11}}{\delta_2} + \delta_3 \log c_{12}
\end{aligned}
\]

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

\[
\text{with (optimization):}
\]

\[
\text{NLPsolve} \begin{bmatrix}
\delta_1 \ln \left( \frac{1}{\delta_1} \right) + \left( 1 - \delta_1 \right) \ln \left( \frac{1}{1 - \delta_1} \right) \\
\delta_2 \ln \left( \frac{0.1(\delta_1 + \delta_2)}{\delta_2} \right) + \delta_1 \ln \left( \frac{0.1 + \delta_1}{\delta_1} \right) + \delta_2 \ln \left( \frac{\delta_1 + \delta_2}{\delta_1} \right) \\
\delta_3 \ln \left( \frac{\delta_1 + \delta_2}{\delta_2} \right) + \delta_1 \ln \left( \frac{\delta_2 + \delta_3}{\delta_2} \right) + \delta_2 \ln \left( \frac{\delta_1 + \delta_2}{\delta_3} \right) \\
\delta_4 \ln \left( \frac{\delta_1 + \delta_2}{\delta_1} \right) + \delta_1 \ln \left( \frac{\delta_2 + \delta_3}{\delta_3} \right) + \delta_2 \ln \left( \frac{\delta_1 + \delta_2}{\delta_1} \right) \\
\delta_5 \ln \left( \frac{\delta_1 + \delta_2}{\delta_1} \right) + \delta_1 \ln \left( \frac{\delta_2 + \delta_3}{\delta_1} \right) + \delta_2 \ln \left( \frac{\delta_1 + \delta_2}{\delta_2} \right)
\end{bmatrix}
\]

\[
\frac{0.36901663595333999,8}{0.776945228258277830}
\]
\[ \delta_3 = 0.15639129886847561, \]
\[ \delta_{11} = 0.11529417638566357, \]
\[ \delta_6 = 0.086480172705934083, \]
\[ \delta_4 = 0.1037906688075505330, \]
\[ \delta_6 = 0.09798366106092626, \]
\[ \delta_7 = 0.290488953113046455, \]
\[ \delta_8 = 0.2907433864.034649, \]
\[ \delta_9 = 0.125507818972236702, \]
\[ \delta_2 = 1 - \delta_1, \delta_5 = \delta_1 \text{ and } \delta_{12} = \delta_6. \]

But \[ \delta_2 = 1 - \delta_1, \delta_5 = \delta_1 \text{ and } \delta_{12} = \delta_6. \]

Again by applying the algorithm we will get:

\[ -0.98907450335862 < \Delta_1 < 0.446504009 < 0.0109259664174 < \Delta_2 < c_1 < 0.98907450335862 < 0.446504009 < \Delta_2 < c_1 < 0.98907450335862 < 0.55495990635 < c_1 < 1.98907450335862 < 0.98907450335862 < \]

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

\[ d\delta^i = \sum_{j=1}^{d} b_j (\sum_{k=1}^{d} \sum_{l=1}^{n} j l_j (\delta^i)^{d_j} b_l (k) \Delta_l c_i k + \Delta_i (k_i - 1) \quad (16) \]

\[ d = \text{The degree of difficulty} \]
\[ n = \text{Number of coefficients that we will change} \]

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**