CHAPTER V

SOLUTION TO SOME ADVANCED PROGRAMMING PROBLEMS

Much of the recent research in mathematical Programming has largely been concentrated on some programming problems such as Separable Programming, Fractional Programming, Geometric Programming and Stochastic Programming Problem etc. Different methods of solution for such problems have been developed during the last four decades. In this chapter we demonstrate through numerical examples, how the Modified Ellipsoid Algorithm could be used to solve all such advanced programming problems.

5.1 Separable Programming

Separable Programming is a class of Mathematical Programming problems in which the function \( g(y) \) can be expressed as the sum of \( n \) single functions \( g_1(y_1), \ldots, g_n(y_n) \).

\[
i.e. \quad g(y_1, \ldots, y_n) = \sum_{i=1}^{n} g_i(y_i) \quad \ldots.. (5.1)
\]

A function \( g(y_1, \ldots, y_n) \) that can be decomposed additively in terms of single variable functions \( g_i(y_i) \) satisfying equation (5.1) is said to be separable. Some times, functions that are not immediately separable can be made so by transformation of variables.
The Separable Programming Problems may be stated as follows:

\[ \text{Minimize} \quad \sum_{i=1}^{n} g_{i0}(y_i) \]

\[ \text{Subject to} \quad \sum_{i=1}^{n} g_{ij}(y_i) \geq b_j \quad (j=1, \ldots, n) \quad \ldots \ldots \quad (5.2) \]

Where \( g_{ij}(y_i) \) are linear or nonlinear.

Different methods have been developed for solving Separable Programming Problems. One of the important methods among them is the Simplex Methods with restricted basis entry. For Nonlinear Programming Problem an approximate solution can be obtained by piece wise linear approximation and the Simplex Method. This approach is found in the works of Charnes and Cooper [1957], Dantzig, Johnson and White [1958] and Markowitz and Manne [1957]. For further discussion on this approach, see Miller [1963] and Wolfe [1963]. In the non convex case, the optimality cannot be claimed with the restricted basis entry route. In this case, choosing a small grid, one can obtain a solution sufficiently close to the global optimal solution. For grid generation scheme see Wolfe [1963].

Here we use the Modified Ellipsoid Method for solving a non-convex separable program.

**Numerical Example**

Minimize \(-x_1^2+6x_1-x_2^2+8x_2^2+\frac{1}{2}x_3\)

Subject to \(x_1+x_2+x_3 \leq 5\) \quad \ldots \ldots (5.3)

\(x_1^2-x_2 \leq 3\)

\(x_1, x_2, x_3 \geq 0.\)
The objective function is concave and thus will have a large number of local minima. The constraint set is convex. The various iterations of the method are given in the following table. The last column of the table gives the status of the point as to it being feasible (F) or Non-feasible (NF).

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$z$</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>12.50000</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>0.49042</td>
<td>-0.16609</td>
<td>0.77626</td>
<td>1.48396</td>
<td>NF</td>
</tr>
<tr>
<td>43</td>
<td>0.12803</td>
<td>0.06775</td>
<td>0.55309</td>
<td>1.45414</td>
<td>F</td>
</tr>
<tr>
<td>46</td>
<td>0.00765</td>
<td>-0.02720</td>
<td>0.94196</td>
<td>0.29087</td>
<td>NF</td>
</tr>
<tr>
<td>66</td>
<td>0.00462</td>
<td>0.02372</td>
<td>-0.10665</td>
<td>0.15895</td>
<td>NF</td>
</tr>
<tr>
<td>89</td>
<td>0.00672</td>
<td>0.00119</td>
<td>0.04771</td>
<td>0.06700</td>
<td>F</td>
</tr>
<tr>
<td>91</td>
<td>0.00141</td>
<td>0.00556</td>
<td>-0.03725</td>
<td>0.03289</td>
<td>NF</td>
</tr>
<tr>
<td>92</td>
<td>-0.00328</td>
<td>0.00398</td>
<td>0.06911</td>
<td>0.04996</td>
<td>NF</td>
</tr>
<tr>
<td>93</td>
<td>0.00483</td>
<td>0.00201</td>
<td>0.03452</td>
<td>0.05747</td>
<td>F</td>
</tr>
</tbody>
</table>

Table 5.1: Solution of the problem by Modified Ellipsoid Method.

The method converges to the global minimum $\begin{bmatrix} 0.00483 \\ 0.00201 \\ 0.03452 \end{bmatrix}$ with an objective function value 0.05747.

5.2 Fractional Programming

We consider a problem in which the objective function is the ratio of two linear functions and the constraints are linear. Such problems are called linear Fractional Programming Problems and are much of interest in Management.
The problem may be stated as follows:

\[ \text{Min} \quad g(y) = \frac{p'y + \alpha}{q'y + \beta} \quad \ldots \ldots (5.4) \]

Subject to \[ A y = b \]

\[ y \geq b \]

Where \( p, q \) are \( n \) vectors, \( b \) is an \( m \) vector, \( A \) is an \( mxn \) matrix, and \( \alpha \) and \( \beta \) are scalars. We know that if an optimal solution for a Linear Fractional Program exists, then an extreme point optimum exists. Furthermore, every local minimum is a global minimum. Some important properties of a linear fractional objective are contained in the following well known result.

Let \( g_0(y) = (p'y + \alpha)/(q'y + \beta) \) and let \( S \) be a convex set such that \( q'y + \beta \neq 0 \) over \( S \). Then \( g_0(y) \) is both pseudo convex and pseudo concave over \( S \).

From the fundamental theorems of pseudo & quasi convex(concave) functions and several implications of the above result for linear programming problem, it may be noted that:

If the objective function is both pseudo convex and pseudo concave over \( S \), then it is also a quasi convex, quasi concave, strictly quasi convex and strictly quasi concave. Further,

1. A point satisfying K-T conditions for a maximization problem is also a global maximum over the feasible region.
2. Any local minimum(maximum) is also a global minimum(maximum) over the feasible region.
3. If the feasible region is bounded, then the objective function has a minimum at an extreme point of the feasible region and also has a maximum at an extreme point of the feasible region.

The above facts about the linear fractional objective function give useful results that are being used to develop suitable computational procedures for solving fractional programming problems. In particular, one searches among the extreme points of the polyhedral set \( \{ y : A y = b, x \geq 0 \} \) until a K-T point is reached.


Also, there are some situations when the numerator or denominator or both of the objective function may be nonlinear. Dinkelbach [1967] has developed a solution technique for solving such problems.
The solution obtained by the Modified Ellipsoid method is \( \begin{bmatrix} 0.00002 \\ 2.99953 \end{bmatrix} \) with a maximum value of \( z = 1.28568 \), which is the optimum solution.

**Numerical Example 2.**

Consider the following fractional programming problem with a nonlinear denominator.

\[
\text{Min } Z = \frac{2x_1 + 2x_2 + 1}{x_1^2 + x_2^2 + 3}
\]

Subject to

\[
x_1 + x_2 \leq 3
\]

\[
x_1, x_2 \geq 0
\]

We get the solution shown in the following table

<table>
<thead>
<tr>
<th>Iteration</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( Z )</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.00000</td>
<td>1.00000</td>
<td>0.83333</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-1.12132</td>
<td>-1.12132</td>
<td>-1.72991</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1.11082</td>
<td>0.44589</td>
<td>0.75887</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.97786</td>
<td>-1.47826</td>
<td>-0.00011</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>0.07251</td>
<td>0.38902</td>
<td>0.58339</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>0.07378</td>
<td>0.10615</td>
<td>0.43050</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>-0.16629</td>
<td>0.15534</td>
<td>0.36340</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>0.05877</td>
<td>0.05979</td>
<td>0.39637</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>-0.00339</td>
<td>-0.08659</td>
<td>0.27328</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>0.00460</td>
<td>0.02290</td>
<td>0.35053</td>
</tr>
<tr>
<td>42</td>
<td>2</td>
<td>0.01033</td>
<td>0.00291</td>
<td>0.33982</td>
</tr>
<tr>
<td>44</td>
<td>2</td>
<td>-0.00662</td>
<td>0.00916</td>
<td>0.33650</td>
</tr>
<tr>
<td>45</td>
<td>2</td>
<td>0.00681</td>
<td>0.00198</td>
<td>0.33766</td>
</tr>
<tr>
<td>48</td>
<td>2</td>
<td>0.00491</td>
<td>0.00130</td>
<td>0.33637</td>
</tr>
</tbody>
</table>

The solution obtained by the Modified ellipsoid Method is \( \begin{bmatrix} 0.00491 \\ 0.00130 \end{bmatrix} \) with a minimum value of \( Z=0.33637 \), which is the optimum solution.
5.3 GEOMETRIC PROGRAMMING

Geometric Programming is a method for solving a class of nonlinear programming problems where one minimizes functions which are in the form of posynomials subject to constraints of the same type.

A function $h(x)$ is called a **Posynomial** if $h$ can be expressed as the sum of power terms each of the form $c_i x_1^{a_{i1}} x_2^{a_{i2}} \cdots x_n^{a_{in}}$

Where $c_i$ and $a_{ij}$ are constants with $c_i > 0$ and $x_j > 0$. Thus a posynomial can be expressed as $h(x)=c_1 x_1^{a_{11}} x_2^{a_{12}} \cdots x_n^{a_{1n}} + \cdots + c_l x_1^{a_{l1}} x_2^{a_{l2}} \cdots x_n^{a_{ln}}$.

A Geometric Programming Problem is one in which the objective function and constraints are expressed as posynomials in $x$. Thus the problem can be defined as follows:

Find $x$ which minimizes

$$f(x) = \sum_{i=1}^{N_0} c_i \prod_{j=1}^{n} x_j^{P_{ij}}, \quad c_i > 0, \quad x_j > 0$$

Subject to

$$g_j(x) = \sum_{i=1}^{N_j} a_{ij} \prod_{k=1}^{n} x_k^{a_{ik}} = 0, \quad a_{ij} > 0$$

$j = 1, 2, \ldots, m,$

Where $N_0$ and $N_j$ denote the number of posynomial terms in the objective and $j$th constraint functions, respectively.

In an Unconstrained Geometric Optimization problem, the Arithmetic mean-Geometric mean inequality conditions
have been used to yield a solution. For a constrained Geometric Optimization, different methods are in use. Passy-Wilde Method [1967] determines the pseudo minimum of a Geometric Program. Ecker and Zoracki [1976] developed a primal solution method for a prototype Geometric Programming Problem. Avriel and Williams [1976] extended the method of Geometric Program to include any rational function of posynomial terms and called the method as complementary Geometric Programming Method.

**Numerical example**

Minimize \( z = 20x_1x_3 + 40x_2x_3 + 80x_1x_2 \)

Subject to

\[
8x_1^{-1}x_2^{-1}x_3^{-1} \leq 1 \\
\text{and} \\
x_1x_2x_3 > 0
\]  

\( \text{.....(5.7)} \)

The values obtained after the implementation of the Modified Ellipsoid Method are given in the following table.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( z )</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>140.00000</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>2.29904</td>
<td>2.29904</td>
<td>2.29904</td>
<td>739.98067</td>
<td>F</td>
</tr>
<tr>
<td>18</td>
<td>0.72467</td>
<td>1.00517</td>
<td>6.06636</td>
<td>390.10451</td>
<td>F</td>
</tr>
<tr>
<td>19</td>
<td>2.44565</td>
<td>0.78620</td>
<td>4.04637</td>
<td>478.99151</td>
<td>F</td>
</tr>
<tr>
<td>20</td>
<td>1.94778</td>
<td>1.40022</td>
<td>3.41449</td>
<td>542.44230</td>
<td>F</td>
</tr>
<tr>
<td>59</td>
<td>1.94339</td>
<td>1.04576</td>
<td>3.94534</td>
<td>480.96698</td>
<td>F</td>
</tr>
<tr>
<td>61</td>
<td>1.91193</td>
<td>0.99328</td>
<td>4.20065</td>
<td>479.44817</td>
<td>F</td>
</tr>
<tr>
<td>74</td>
<td>1.95839</td>
<td>1.03843</td>
<td>3.93206</td>
<td>480.02895</td>
<td>F</td>
</tr>
<tr>
<td>75</td>
<td>1.91338</td>
<td>1.03149</td>
<td>4.05700</td>
<td>480.53324</td>
<td>F</td>
</tr>
<tr>
<td>99</td>
<td>1.97121</td>
<td>1.00915</td>
<td>4.02065</td>
<td>479.94692</td>
<td>F</td>
</tr>
<tr>
<td>100</td>
<td>2.01823</td>
<td>1.00565</td>
<td>3.94071</td>
<td>479.95545</td>
<td>F</td>
</tr>
<tr>
<td>107</td>
<td>2.01295</td>
<td>0.99776</td>
<td>3.98298</td>
<td>479.98849</td>
<td>F</td>
</tr>
<tr>
<td>110</td>
<td>2.00042</td>
<td>1.00385</td>
<td>3.98383</td>
<td>480.00214</td>
<td>F</td>
</tr>
</tbody>
</table>

Table 5.2: Solution of the Geometric Programming Problem by the Modified Ellipsoid Method.
The solution obtained is \[ \begin{bmatrix} 2.00042 \\ 1.00385 \\ 3.98383 \end{bmatrix} \] with a minimum value of \( z = 480.002 \), which is the optimum feasible point.

5.5 STOCHASTIC PROGRAMMING

Stochastic programming deals with situations where some or all the parameters of the problem are described by random variables. Such cases seem typical of real-life problems, where it is found difficult to determine the values of the parameters exactly. We know that in the case of linear programming, sensitivity analysis can be used to study the effect of changes in problem’s parameters on optimal solution. This, however, represents only a partial answer to the problem especially when the parameters are actually random variables. The objective of stochastic programming is to consider these random effects explicitly in the solution of the model.

The basic idea of all stochastic programming models is to convert the probabilistic nature of the problem into an equivalent deterministic situation. Several models have been developed to handle special cases of the general problem. In this section, the idea of employing deterministic equivalence is illustrated with the introduction of the interesting technique of chance constrained programming. Once it is converted, the problem can be easily solved by the Modified Ellipsoid Method.
A chance constrained model is defined generally as

\[ \text{Maximize } Z = \sum_{j=1}^{n} c_j x_j \]

S.t.

\[ \left\{ \sum_{j=1}^{n} a_{ij} x_j \leq b_i \right\} \geq 1 - \alpha_i \quad \ldots \ldots (5.8) \]

\[ i=1, \ldots, m \]

\[ x_j \geq 0 \text{ for all } j \]

The name "Chance-constrained" follows from each constraint

\[ \sum_{j=1}^{n} a_{ij} x_j \leq b_i \]

being realized with a minimum probability of \( 1 - \alpha_i \), \( 0 < \alpha_i < 1 \)

In the general case, it is assumed that \( c_j \), \( a_{ij} \) and \( b_i \) are all random variables. The fact that \( c_j \) is a random variable can always be treated by replacing it by its expected value. Here we consider three cases. The first two correspond to the separate considerations of \( a_{ij} \) and \( b_i \) as random variables. The third case combines the random effects of \( a_{ij} \) and \( b_i \). In all the cases, it is assumed that the parameters are normally distributed with known means and variances.

**Case I**: In this case each \( a_{ij} \) is normally distributed with mean \( E\{a_{ij}\} \) and variance \( \text{Var}\{a_{ij}\} \). Also, the covariance of \( a_{ij} \) and \( a_{ij}' \) is given by \( \text{cov}\{a_{ij}, a_{ij}'\} \).
Consider the \( i \)th constraint
\[
P\left\{ \sum_{j=1}^{n} a_{ij}x_j \leq b_i \right\} \geq 1 - a_i
\]
and define \( h_i = \sum_{j=1}^{n} a_{ij}x_j \)
Then \( h_i \) is normally distributed with
\[
E\{h_i\} = \sum_{j=1}^{n} E\{a_{ij}\}x_j \quad \text{and} \quad \text{Var}\{b_i\} = X'D_iX
\]
where, \( X=(x_1, x_2, \ldots, x_n)' \)
\[
D_i=\text{ith covariance matrix} = \begin{bmatrix} \text{Var}\{a_{i1}\} & \cdots & \text{Cov}\{a_{i1}, a_{in}\} \\ \text{Cov}\{a_{in}, a_{i1}\} & \cdots & \text{Var}\{a_{in}\} \end{bmatrix}
\]
Now,
\[
P\{h_i \leq b_i\} = P\left\{ \frac{h_i - E\{h_i\}}{\sqrt{\text{Var}\{h_i\}}} \leq \frac{b_i - E\{h_i\}}{\sqrt{\text{Var}\{h_i\}}} \right\} \geq 1 - a_i
\]
where, \( (h_i - E\{h_i\})/\sqrt{\text{Var}\{h_i\}} \) is standard normal with mean zero and variance one. This means that
\[
P\{h_i \leq b_i\} = \phi \left( \frac{b_i - E\{h_i\}}{\sqrt{\text{Var}\{h_i\}}} \right)
\]
where \( \phi \) represents the CDF of the standard normal distribution.

Let \( K_{a_i} \) be the standard normal value such that
\[
\phi (K_{a_i}) = 1 - a_i
\]
Then the statement $P(h_i \leq b_i) \geq 1 - a_i$ is realized if and only if
\[ b_i - E(h_i) \geq K a_i / \sqrt{\text{Var}(h_i)} \]

This yields the following nonlinear constraint
\[ \sum_{j=1}^{n} E(aij)x_j + K a_i \sqrt{x' D_i X} \leq b_i \]
which is equivalent to the original stochastic constraint.

The problem (5.8) now reduces to the Non linear Programming problem

\[
\text{Maximize } Z = \sum_{j=1}^{n} C_j x_j \\
\text{S.t. } \sum_{j=1}^{n} E(aij)x_j + K a_i \sqrt{x' D_i X} \leq b_i \\
i=1, \ldots, m \\
x_j \geq 0 \text{ for all } j
\]

For the special case where the normal distributions are independent,
\[ \text{Cov}\{aij, aij, a'i a'j\} = 0 \]
and the last constraint reduces to
\[ \sum_{j=1}^{n} E(aij)x_j + K a_i \left( \sum_{j=1}^{n} \text{Var}(aij)x_j^2 \right) \leq b_i \]
This constraint can now be put in the Separable Programming form using substitution

\[ y_i = \sum_{j=1}^{n} \text{Var}(a_{ij})x_j^2, \text{for all } i \]

Thus the original constraint is equivalent to

\[ \sum_{j=1}^{n} E(a_{ij})x_j + K_i y_i \leq b_i \]

and

\[ \sum_{j=1}^{n} \text{Var}(a_{ij})x_j^2 - y_i^2 = 0 \]

where \( y_i \geq 0 \).

Case II

In this case only \( b_i \) is normal with mean \( E(b_i) \) and variance \( \text{Var}(b_i) \). The analysis in this case is very similar to that of case 1. Consider the stochastic constraint

\[ P\left\{ b_i \geq \sum_{j=1}^{n} a_{ij} \right\} \geq a_i . \]

As in case 1,

\[ P\left\{ \frac{b_i - E(b_i)}{\sqrt{\text{Var}(b_i)}} \geq \frac{\sum_{j=1}^{n} a_{ij}x_j - E(b_i)}{\sqrt{\text{Var}(b_i)}} \right\} \geq a_i \]
This can hold if only if
\[
\sum_{j=1}^{n} a_{ij}x_j - E(b_i) \leq K_{a_i} \sqrt{\text{Var}(b_i)}
\]

Thus the stochastic constraint is equivalent to the deterministic linear constraint
\[
\sum_{j=1}^{n} a_{ij}x_j \leq E(b_i) + K_{a_i} \sqrt{\text{Var}(b_i)}
\]

Thus, in case 2, the chance constrained model can be converted into the following equivalent linear programming problem.

\[
\begin{align*}
\text{Maximize } Z &= \sum_{j=1}^{n} C_j x_j \\
\text{S.t. } \sum_{j=1}^{n} a_{ij} x_j &\leq E(b_i) + K_{a_i} \sqrt{\text{Var}(b_i)} \\
&i = 1, \ldots, m & \quad & x_j \geq 0 \text{ for all } j
\end{align*}
\]

**Case III**

In this case all \( a_{ij} \) and \( b_i \) are normal random variables. Consider the constraint
\[
\sum_{j=1}^{n} a_{ij}x_j \leq b_i
\]

This may be written
\[
\sum_{j=1}^{n} a_{ij}x_j - b_i \leq 0
\]
Since all $a_{ij}$ and $b_i$ are normal, it follows from the theory of statistics that $\sum_{j=1}^{n} a_{ij} x_j - b_i$ is also normal. This shows that chance constraint reduces in this case also to an equivalent Nonlinear Programming problem.

Numerical Example:

Consider the chance constrained problem

Maximize $z = 5x_1 + 6x_2 + 3x_3$

Subject to

$P\{a_{11} x_1 + a_{12} x_2 + a_{13} x_3 \leq 8\} \geq 0.95 \quad \ldots (5.11)$

$P\{5x_1 + x_2 + 6x_3 \leq b_2\} \geq 0.10$

with all $x_j \geq 0$. Suppose that the $a_{ij}$’s are independent normally distributed random variables with the following means and variances.

$E\{a_{11}\}=1, \quad E\{a_{12}\}=3, \quad E\{a_{13}\}=9$

$\text{Var}\{a_{11}\}=-25, \quad \text{Var}\{a_{12}\}=16, \quad \text{Var}\{a_{13}\}=4$

The parameter $b_2$ is normally distributed with mean 7 and variance 9.

From the standard normal tables,

$K_{a_1}=K_{0.05} = 1.645, \quad K_{a_2} = K_{0.01} = 1.285$

For the first constraint, the equivalent deterministic constant is given by

$x_1 + 3x_2 + 9x_3 + 1.645\sqrt{25x_1^2 + 16x_2^2 + 4x_3^2} \leq 8.$

and for the second constraint,

$5x_1 + x_2 + 6x_3 \leq 7 + 1.285(3) = 10.855$

If we let $x_4^2 = 25x_1^2 + 16x_2^2 + 4x_3^2$

the complete problem becomes
Maximize \[ z = 5x_1 + 6x_2 + 3x_3 \]
Subject to \[ x_1 + 3x_2 + 9x_3 + 1.645x_4 \leq 8 \]
\[ 25x_1^2 + 16x_2^2 + 4x_3^2 - x_4^2 = 0 \] ...(5.12)
\[ 5x_1 + x_2 + 6x_3 \leq 10.855 \]
\[ x_1, x_2, x_3, x_4 \geq 0 \]

This is a problem with non-convexity occurring in the feasible set due to equality sign in the non-linear constraint. The problem may be solved by considering an inequality sign in the non-linear constraint. If at the final solution, the non-linear constraint is active, then this obviously is the optimal solution. Otherwise we reverse the inequality sign of the non-linear constraint. For implementing the Modified Ellipsoid Method to the above problem, first we try to solve the following problem.

Maximize \[ z = 5x_1 + 6x_2 + 3x_3 \]
Subject to \[ x_1 + 3x_2 + 9x_3 + 1.645x_4 \leq 8 \]
\[ 25x_1^2 + 16x_2^2 + 4x_3^2 - x_4^2 \geq 0 \] ...(5.13)
\[ 5x_1 + x_2 + 6x_3 \leq 10.855 \]
\[ x_1, x_2, x_3, x_4 \geq 0 \]

It is found that the constraint set is not feasible. Then we reverse the inequality to get the following problem.

97
Maximize \[ z = 5x_1 + 6x_2 + 3x_3 \]

Subject to \[ x_1 + 3x_2 + 9x_3 + 1.645x_4 \leq 8 \]
\[ 25x_1^2 + 16x_2^2 + 4x_3^2 - x_4^2 \leq 0 \] \(...(5.14)\)
\[ 5x_1 + x_2 + 6x_3 \leq 10.855 \]
\[ x_1, x_2, x_3, x_4 \geq 0 \]

The following table gives the various iterations. The method finally converges to a solution where the second constraint is active. This solution, therefore will also represent the final solution of the problem. (5.12)

<table>
<thead>
<tr>
<th>Iteration</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( z )</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.00000</td>
<td>14.00000</td>
<td>NF</td>
</tr>
<tr>
<td>2</td>
<td>0.53250</td>
<td>-0.40250</td>
<td>-3.20750</td>
<td>0.23096</td>
<td>9.37498</td>
<td>F</td>
</tr>
<tr>
<td>30</td>
<td>0.29654</td>
<td>1.56545</td>
<td>-0.44278</td>
<td>-4.73189</td>
<td>9.54707</td>
<td>F</td>
</tr>
<tr>
<td>31</td>
<td>0.24616</td>
<td>0.49468</td>
<td>0.29812</td>
<td>-6.20622</td>
<td>5.09324</td>
<td>F</td>
</tr>
<tr>
<td>32</td>
<td>-0.05148</td>
<td>0.17578</td>
<td>-0.03432</td>
<td>-2.34427</td>
<td>0.69436</td>
<td>F</td>
</tr>
<tr>
<td>64</td>
<td>0.39263</td>
<td>0.15390</td>
<td>0.08289</td>
<td>3.21153</td>
<td>3.13524</td>
<td>F</td>
</tr>
<tr>
<td>66</td>
<td>0.49511</td>
<td>0.47319</td>
<td>-0.16380</td>
<td>2.55742</td>
<td>4.82329</td>
<td>F</td>
</tr>
<tr>
<td>67</td>
<td>0.40414</td>
<td>0.46738</td>
<td>-0.30021</td>
<td>4.29543</td>
<td>3.92435</td>
<td>F</td>
</tr>
<tr>
<td>68</td>
<td>0.28603</td>
<td>0.26052</td>
<td>0.26700</td>
<td>2.63357</td>
<td>3.79432</td>
<td>F</td>
</tr>
<tr>
<td>115</td>
<td>0.33349</td>
<td>0.64826</td>
<td>0.06853</td>
<td>3.19288</td>
<td>5.76259</td>
<td>F</td>
</tr>
<tr>
<td>116</td>
<td>0.30742</td>
<td>0.66600</td>
<td>0.04563</td>
<td>3.00523</td>
<td>5.67000</td>
<td>F</td>
</tr>
<tr>
<td>117</td>
<td>0.51235</td>
<td>0.49219</td>
<td>0.02191</td>
<td>3.39851</td>
<td>5.58061</td>
<td>F</td>
</tr>
<tr>
<td>119</td>
<td>0.53304</td>
<td>0.48826</td>
<td>0.07793</td>
<td>3.23817</td>
<td>5.82858</td>
<td>F</td>
</tr>
<tr>
<td>169</td>
<td>0.50149</td>
<td>0.58752</td>
<td>0.00614</td>
<td>3.42309</td>
<td>6.05100</td>
<td>F</td>
</tr>
<tr>
<td>170</td>
<td>0.42337</td>
<td>0.65240</td>
<td>0.00257</td>
<td>3.38175</td>
<td>6.03895</td>
<td>F</td>
</tr>
<tr>
<td>173</td>
<td>0.40113</td>
<td>0.67234</td>
<td>0.00864</td>
<td>3.31086</td>
<td>6.06563</td>
<td>F</td>
</tr>
<tr>
<td>174</td>
<td>0.48949</td>
<td>0.59845</td>
<td>0.00584</td>
<td>3.41506</td>
<td>6.05565</td>
<td>F</td>
</tr>
<tr>
<td>209</td>
<td>0.44344</td>
<td>0.64570</td>
<td>0.00073</td>
<td>3.40677</td>
<td>6.09361</td>
<td>F</td>
</tr>
</tbody>
</table>

Table 5.2: Solution of the Stochastic Programming Problem by the Modified Ellipsoid Method.

The final solution obtained is given by

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  x_4
\end{bmatrix}
= \begin{bmatrix}
  0.44344 \\
  0.64570 \\
  0.00073 \\
  3.40677
\end{bmatrix}
\]

with maximum \( z = 6.0936 \).