CHAPTER 7
ANALYSIS OF ALGORITHMS

7.1 Introduction

An algorithm is essentially a set of instructions to perform calculations in a certain way. More than one algorithm might work to perform the same operation, but some algorithms use more memory and take longer time to perform than others. Hence, the algorithms designed must be analyzed for its efficiency over existing ones. An algorithm can be analyzed[1] by writing and running a computer program. Algorithm analysis is most concerned with finding out how much time a program takes to run[4], and how much memory storage space it needs to execute the program.

An algorithm may run faster on certain data sets than on others. Finding the average case can be very difficult, so typically algorithms are measured by the worst-case time complexity[29].

7.2 Computational aspects

Generally, in solving Linear Programming Problems, researchers compare more than one algorithm with the developed one based on the total number iterations to reach the optimal solution of the problem. But in this thesis, the comparison is based on the number iterations and time for typical problems. The execution time may vary based on the multiply/divide[5] operations in solving problems.

From the execution of sum of ratios linear fractional programming algorithm given in chapter three and multi objective optimization algorithm given in chapter four, it was observed that the computational savings obtained on the
basis of number of iterations and execution time is significant. Also it was observed that both algorithms reduce the computation effort and memory requirements due to unnecessary popping in and out of promising variables by allowing promising variables to enter into basis only if they improve the sum of ratios value.

7.3 Basis for Comparison

Algorithms, in general, may have different computational efficiencies. Generally optimization algorithms are compared based on the total number of iterations required to find optimal solution. The computational efforts required for different methods may vary depending on the procedure adopted. In this thesis, computational efforts required for different methods are presented. Large scale problems are solved using the proposed algorithms. Multiply/divide operations as per formulae derived and as per proposed algorithms are compared. Computation time taken by the computer in finding solution to problems of different sizes are presented. The methods are:

i) Complete ratio algorithm
ii) Modified ratio algorithm
iii) Proposed sum of ratios fractional programming algorithm
iv) Proposed multi objective optimization algorithm

7.4 Computational effort required for Complete Ratio algorithm

The computational efforts required for complete ratio algorithm are given below:
i) To find the most promising variable the multiply/divide operations required are \((2m+1)n = 2mn + n\). This includes the computation of \(z_{nj} - c_j, z_{dj} - d_j\) and their ratios.

ii) To find the leaving variable, the computations \(B^{-1}P_0, B^{-1}P_j\) and ratios \(\left\{ \frac{(B^{-1}P_0)_i}{(B^{-1}P_j)_i} ; (B^{-1}P_j)_i > 0 \right\}\) are to be carried out. This requires \(2m^2 + m\) multiply/divide operations.

iii) The finding of \(\eta_{\text{new}}\) vector \((m+2)\) divisions are required.

iv) The computation of \(M'^{-1}_{\text{new}}\) based on the relation
\[
M'^{-1}_{\text{new}} = E^{-1} M'^{-1}_{\text{current}}
\]
requires \((m+2)^2\) multiply/divide operations.

Hence the total number of multiply/divide operations needed for an iteration is \(3m^2 + 2mn + n + 6m + 6\).

If \(s\) number of iterations are needed to reach the optimal solution of a linear fractional programming problem, then the total number of multiply/divide operations required is \((3m^2 + 2mn + n + 6m)\) s.

**7.5 Computational effort required for Modified Ratio algorithm**

The computational efforts required for modified ratio algorithm are given below:

i) Computation needed for arrangement of promising variables:

To find the maximum value with which each of the decision variable \(x_j\) can enter the basis, the computation
\[
\begin{align*}
\begin{cases}
\frac{b_{ij}}{a_{ij}}; i = 1,\ldots,m, j = 1,\ldots,n
\end{cases}
\end{align*}
\]
requires \( n \times m \) multiply/divide operations.

The computation of the ratios
\[
\begin{align*}
\begin{cases}
\frac{c_jx_j + c_0}{d_jx_j + d_0}; j = 1,\ldots,n
\end{cases}
\end{align*}
\]
requires \( 3n \) operations.

In the first pass, the total number of multiply/divide operations required for arrangement of promising variables is \((nm + 3n)\) operations. Since the number of promising variables are reduced to \( n - m \) for subsequent passes, the number of multiply/divide operations required for arrangement is \((n - m) m + 3 (n - m)\).

ii) Computation needed for an iteration:

To check the promisibility of an ordered variable \( z_{nj} - c_j \), \( z_{dj} - d_j \) and their ratios are to be computed.

This requires \( 2m + 1 \) multiply / divide operations.

iii) To find the leaving variable, the computations \( B^{-1}P_0 \), \( B^{-1}P_j \) of the promising variable and ratio \( \left\{ \frac{(B^{-1}P_0)i}{(B^{-1}P_j)i}; (B^{-1}P_j)i > 0 \right\} \) are to be carried out. This requires \( 2m^2 + m \) multiply/divide operations.

iv) The find \( \eta_{\text{new}} \) vector \((m+2)\) divisions are required.

v) The computation of \( M^{-1}_{\text{next}} \) based on the relation

182
Hence the total number of multiply/divide operations needed for an iteration is

\[(2m+1) + 2m^2 + m + 2 + (m+2)^2\]
\[= 3m^2 + 8m + 7\]

If \( o \) is the number of passes and \( s \) is the number of iterations/basis changes needed to reach the optimal solution of a linear fractional programming problem, then the total number of multiply/divide operations required is

\[= (nm + 2n) + ((n-m)m+(3n-m)) (o-1)+(3m^2+8m) s.\]

### 7.6 Computational effort required for Sum of ratios Linear Fractional Programming Problem

(i) Computations needed for arrangement:

To find maximum value with which each of the decision variable \( x_j \) can enter the basis, the computation of \( \{ b_i / a_{ij} : i=1..m, j=1..n \} \) requires \( n x m \) multiply/divide operations.

(ii) The computation of the ratio

\[ Z = \sum_{i=1}^{t} \sum_{j=1}^{n} \frac{c_i x_j + c_{0i}}{d_i x_j + d_{0i}} \]

requires \( (3*f)n \) operations are required, where \( f \) represents the number of fractions. Hence the total number of multiply/divide operations required for arrangement is \( (nm+(3*f)n) \) operations.

(iii) Computations needed for an iteration
(a) To find whether an ordered variable is promising \( z_{ij}, c_j, zd_j, d_j \) and their ratio are to be computed. It requires \( 2fm+f \) multiply/divide operations.

(b) To find the leaving variable \( B'P_0, B'P_j \) and the ratios are to be computed.

\[
0 = \min_{i=1..m} \left\{ \frac{(B^{-1}P_0)_i}{(B^{-1}P_j)_i} : (B^{-1}P_j)_i > 0 \right\}
\]

This requires \( 2m^2 + m \) multiply/divide operations.

(c) To obtain \( n_{new} \) vector \( m+2f \) divisions are required.

(d) The computation of \( M^{-1}_{next} \) based on the relation

\[
M^{-1}_{next} = E^{-1} \ast M^{-1}_{current}
\]

needs \( (m+2f)^2 \) multiplication operations.

Hence the total number of multiply/divide operations needed for an iteration is

\[
2fm+f + 2m^2 + m + m + 2f + (m+2f)^2
\]

\[
2fm+f+2m^2+2m+2f+m^2+4f^2+4fm
\]

\[
6fm+3f+3m^2+2m+4f^2
\]

\[
3m^2 + (6f + 2)m + f(4f + 3)
\]

Suppose, \( f=2 \) (fractions), \( m=7 \) (constraints), \( n=12 \) (variables),

184
3*7^2 + (6*2 + 2)*7 +2(4*2 + 3) = 267 multiply/divide operations for an iteration.

Hence, total number of multiply/divide operations need for a sum of ratios linear fractional programming problem solved in ‘t’ number of iterations is:

\[ nm+(3*f)n+ [(3m^2 + (6f + 2)m +f(4f + 3))] * t \]

### 7.7 Computational effort required for Multi objective Linear Fractional Programming Problem

(i) Computations needed for arrangement:

To find maximum value with which each of the decision variable \( x_j \) can enter the basis, the computation of \( \{b_i/a_{ij} ; i=1..m, j=1..n\} \) requires \( n \times m \) number of multiply/divide operations.

(ii) The computation of the ratio

\[ Z = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{c_{ij}x_j+c_{0i}}{d_{ij}x_j+d_{0i}} \]

requires \( (a+a/2 )n \) multiply/divide operations, where \( a \) represents the number of objectives. Hence the total number of multiply/divide operations required for arrangement is \( (nm+(a+a/2)n) \) operations.

(iii) Computations needed for an iteration

(a) To find whether an ordered variable is promising, \( z_{nj}=c_j, z_{dj}=d_j \) and their ratios are to be computed.

It requires \( am+a/2 \) multiply/divide operations.

(b) To find the leaving variable \( B^{-1}P_0B^{-1}/P_j \) and the ratios are to be computed.
This requires \(2m^2+m\) multiply/divide operations.

(c) To obtain \(n_{\text{new}}\) vector \(m+a\) divisions are required.

(d) The computation of \(M^{-1}_{\text{next}}\) based on the relation \(M^{-1}_{\text{next}}=E^{-1}_{\text{next}} \cdot M^{-1}_{\text{current}}\) needs \((m+a)^2\) multiply/divide operations.

Hence the total number of multiply/divide operations needed for an iteration is

\[
am + \frac{a}{2} + 2m^2 + 2m + a + (m + a)^2
\]

\[
am + \frac{1}{2}a + 2m^2 + 2m + a + m^2 + a^2 + 2am
\]

\[3am + 3/2a + 3m^2 + 2m + a^2
\]

\[3m^2 + (3a + 2)m + (a + 3/2)a
\]

Suppose, \(a=4\) (objective functions), \(m=8\) (constraints), \(n=10\) (variables),

\[3 \cdot 8^2 + (3 \cdot 4 + 2) \cdot 8 + (4 + 3/2) \cdot 4 = 326\] multiply/divide operations for an iteration.

Hence, total number of multiply/divide operations needed for multiobjective linear programming problem solved in ‘\(t\)’ number of iterations is:

\[\text{(nm + (a + a/2)n) + } [3m^2 + (3a + 2)m + (a + 3/2)a] * t\]

Thus the formula for \(k\) number of permutations of objectives is

\[k[(nm + (a + a/2)n) + [3m^2 + (3a + 2)m + (a + 3/2)a] * t]\] where \(k = \text{p_2}\)
Table 7.1 Multiply/Divide operations in Sum of Ratios Linear Fractional Programming Algorithm

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>Objective Type</th>
<th>Number of Objectives</th>
<th>Number of Fractions</th>
<th>Number of Constraints</th>
<th>Number of Variables</th>
<th>No. of Iterations</th>
<th>Multiply Divide Operations</th>
<th>Execution Time in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Max</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>12</td>
<td>4</td>
<td>1224</td>
<td>144</td>
</tr>
<tr>
<td>2</td>
<td>Min</td>
<td>32</td>
<td>16</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1668</td>
<td>718</td>
</tr>
<tr>
<td>3</td>
<td>Max</td>
<td>12</td>
<td>6</td>
<td>15</td>
<td>56</td>
<td>12</td>
<td>18732</td>
<td>1668</td>
</tr>
<tr>
<td>4</td>
<td>Max</td>
<td>16</td>
<td>8</td>
<td>20</td>
<td>96</td>
<td>12</td>
<td>33984</td>
<td>3477</td>
</tr>
<tr>
<td>5</td>
<td>Max</td>
<td>20</td>
<td>10</td>
<td>20</td>
<td>100</td>
<td>44</td>
<td>131280</td>
<td>4273</td>
</tr>
<tr>
<td>6</td>
<td>Min</td>
<td>18</td>
<td>9</td>
<td>25</td>
<td>144</td>
<td>12</td>
<td>9261308</td>
<td>5555</td>
</tr>
<tr>
<td>7</td>
<td>Min</td>
<td>20</td>
<td>10</td>
<td>36</td>
<td>180</td>
<td>136</td>
<td>902680</td>
<td>7892</td>
</tr>
<tr>
<td>8</td>
<td>Min</td>
<td>30</td>
<td>15</td>
<td>52</td>
<td>192</td>
<td>103</td>
<td>1444247</td>
<td>13035</td>
</tr>
<tr>
<td>9</td>
<td>Max</td>
<td>30</td>
<td>15</td>
<td>50</td>
<td>400</td>
<td>362</td>
<td>4760290</td>
<td>23700</td>
</tr>
<tr>
<td>10</td>
<td>Max</td>
<td>60</td>
<td>30</td>
<td>50</td>
<td>400</td>
<td>362</td>
<td>7400980</td>
<td>45399</td>
</tr>
<tr>
<td>11</td>
<td>Min</td>
<td>30</td>
<td>15</td>
<td>60</td>
<td>800</td>
<td>611</td>
<td>10632915</td>
<td>53134</td>
</tr>
<tr>
<td>12</td>
<td>Max</td>
<td>60</td>
<td>30</td>
<td>60</td>
<td>800</td>
<td>632</td>
<td>16179120</td>
<td>90529</td>
</tr>
<tr>
<td>13</td>
<td>Min</td>
<td>60</td>
<td>30</td>
<td>65</td>
<td>1000</td>
<td>752</td>
<td>21357640</td>
<td>113795</td>
</tr>
<tr>
<td>14</td>
<td>Max</td>
<td>60</td>
<td>30</td>
<td>254</td>
<td>1000</td>
<td>237</td>
<td>58045442</td>
<td>99929</td>
</tr>
</tbody>
</table>
Table 7.2 Multiply/Divide Operations in Multi objective Linear Programming Algorithm

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>Objective Type</th>
<th>Number of Constraints</th>
<th>Number of Variables</th>
<th>No. of Iterations for 12 combinations</th>
<th>Multiply divide operations</th>
<th>Execution Time in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>max</td>
<td>7</td>
<td>12</td>
<td>12</td>
<td>5076</td>
<td>1206</td>
</tr>
<tr>
<td>2</td>
<td>max</td>
<td>15</td>
<td>56</td>
<td>128</td>
<td>130208</td>
<td>7159</td>
</tr>
<tr>
<td>3</td>
<td>min</td>
<td>15</td>
<td>56</td>
<td>66</td>
<td>73974</td>
<td>6196</td>
</tr>
<tr>
<td>4</td>
<td>min</td>
<td>20</td>
<td>96</td>
<td>84</td>
<td>156120</td>
<td>10888</td>
</tr>
<tr>
<td>5</td>
<td>max</td>
<td>25</td>
<td>144</td>
<td>170</td>
<td>455781</td>
<td>18369</td>
</tr>
<tr>
<td>6</td>
<td>min</td>
<td>36</td>
<td>180</td>
<td>151</td>
<td>757234</td>
<td>17613</td>
</tr>
<tr>
<td>7</td>
<td>max</td>
<td>52</td>
<td>192</td>
<td>188</td>
<td>1799688</td>
<td>23164</td>
</tr>
<tr>
<td>8</td>
<td>max</td>
<td>50</td>
<td>400</td>
<td>291</td>
<td>2661402</td>
<td>38902</td>
</tr>
<tr>
<td>9</td>
<td>min</td>
<td>50</td>
<td>400</td>
<td>257</td>
<td>2381854</td>
<td>38955</td>
</tr>
<tr>
<td>10</td>
<td>max</td>
<td>60</td>
<td>800</td>
<td>587</td>
<td>7479194</td>
<td>85572</td>
</tr>
<tr>
<td>11</td>
<td>max</td>
<td>60</td>
<td>800</td>
<td>372</td>
<td>4971864</td>
<td>80137</td>
</tr>
<tr>
<td>12</td>
<td>max</td>
<td>65</td>
<td>1000</td>
<td>475</td>
<td>7315325</td>
<td>132527</td>
</tr>
<tr>
<td>13</td>
<td>max</td>
<td>254</td>
<td>1000</td>
<td>1306</td>
<td>260566556</td>
<td>201448</td>
</tr>
</tbody>
</table>

188
Table 7.3 Results of Proposed sum of ratios fractional programming algorithm for the problem given in numerical example

<table>
<thead>
<tr>
<th>No. of Fractions</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>Obj. Function Value</th>
<th>No. of Iterations</th>
<th>Execution Time in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>1.66667</td>
<td>0</td>
<td>0.250000</td>
<td>1</td>
<td>.078</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>0</td>
<td>3.33333</td>
<td>-0.279894</td>
<td>1</td>
<td>.079</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>3.33333</td>
<td>0</td>
<td>1.792981</td>
<td>1</td>
<td>.078</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>3.33333</td>
<td>0</td>
<td>2.205069</td>
<td>1</td>
<td>.079</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>3.33333</td>
<td>3.33333</td>
<td>3.859076</td>
<td>1</td>
<td>.110</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1.486711</td>
<td>1</td>
<td>.078</td>
</tr>
</tbody>
</table>

Results produced by EM image space approach and KNITRO numerical software given in Stochastic Search Algorithm[61] for sum of ratios fractional programming problem given in numerical example 1.

Table 7.4 EM image space and KNITRO results for the same Problem

<table>
<thead>
<tr>
<th>No. of fractions</th>
<th>EM image space approach (Objective Value)</th>
<th>Run Time in Seconds</th>
<th>KNITRO - multistart (Objective Value)</th>
<th>Run Time in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.23617</td>
<td>30.29</td>
<td>0.23609</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0.0275</td>
<td>27.64</td>
<td>0.02750</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>-0.6513</td>
<td>25.43</td>
<td>-0.65126</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1.4083</td>
<td>16.54</td>
<td>1.40828</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1.6128</td>
<td>25.37</td>
<td>1.61279</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>2.2427</td>
<td>17.375</td>
<td>2.24275</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>2.4813</td>
<td>29.4688</td>
<td>2.310583</td>
<td>0</td>
</tr>
<tr>
<td>16*</td>
<td>1.6348</td>
<td>7091</td>
<td>1.48671</td>
<td>0.156</td>
</tr>
</tbody>
</table>

Maximum No. of Iterations: 30

* EM image space approach with 100 iterations.
Sum of Ratios Linear Fractional Programming Algorithm

Problem Number

Multiply / Divide Operations

0 200000 400000 600000 800000 1000000 1200000 1400000 1600000

5 6 7 8

As per Formula
As per Package
7.8 Conclusion

The computational aspects of sum of ratios linear fractional programming algorithm and multi objective linear programming algorithm were presented in this chapter. Formulae were derived to compute the multiply/divide operations required for the developed algorithms. Computer packages were written and run in C compiler for small scale problems and Visual C++ compiler for large scale problems. The number of multiply/divide operations based on the formulae derived is compared with the ones obtained as from the package. The results are tabulated and plotted. The package uses lesser number of multiply/divide operations due to elimination of an iteration when the promising variables do not improve the solution. Hence, unnecessary popping in and popping of variables to and from the basis is avoided. Next chapter, presents the software package developed to implement the proposed algorithms.