A Computer Technique for Solving Linear Fractional Programming Problems by Using Dinkelbach's Algorithm

Sajal Chakroborty[1] and Md. Babul Hasan[2*]

[1]*Department of Electronics and Communications Engineering, East West University, Dhaka-1212, Bangladesh*

[2]*Department of Mathematics, Dhaka University, Dhaka-1000, Bangladesh*

Abstract

In this paper, we introduce a computer oriented technique for solving linear fractional programming (LFP) problems by converting it into a single linear programming (LP) problem. We have used the idea of Dinkelbach's algorithm. We use a mathematical programming language (AMPL) to develop computer code. A number of numerical examples are used to demonstrate the technique.

I. Introduction

Linear fractional programming (LFP) is a mathematical programming that consists of linear constraints with an objective function which is also a ratio of two linear functions[1]. Actually it is a linear programming (LP) problem. When denominator of LFP becomes a constant then it can be converted into a LP problem. Hungarian mathematician B. Martors and his associates developed the field of LFP[2] in 1960's. There are many techniques for solving LFP by transforming into LP. Among them Charnes and Cooper' (1962, 1973) method, Bitran and Novae's (1972) method, Swarup's method[3] are noteworthy. In this paper, we have presented a computer oriented technique for solving LFP by using the idea of Dinkelbach's algorithm[4]. It transforms the objective function as a subtraction of two linear functions by multiplying denominator with a parameter. It stops when value of the transformed objective function becomes zero.

LFP has lots of applications in business and economics such as to determine maximum return on investment, minimum risk asset to capital, minimum debt to equity ratio etc. In the next section, we have outlined the paper.

Paper Outline

We have started our work by discussing some basic definitions and preliminaries in Section II. In Section III, we discuss some existing techniques. In Section IV, we present Dinkelbach's algorithm. In Section V and in Section VI, we have presented our developed computer code in AMPL and have demonstrated our technique by analyzing numerical examples respectively. Finally, we have drawn conclusion about our work in Section VII.

II. Preliminaries

Problems of LFP arise when it becomes important to optimize the efficiency of some activities. In this section, we have discussed some basic ideas and definitions about LFP.

Linear Fractional Programming (LFP)

Mathematically LFP problems are defined as follows[1].

$$Maximize \quad z = \frac{c^T x + \alpha}{d^T x + \beta} \tag{2.1}$$

---

* Author for correspondence. e-mail: mbabulhasan@yahoo.com

subject to

$$Ax \le b \tag{2.2}$$

$$x \ge 0 \tag{2.3}$$

Where $x, c, d \in \mathbb{R}^n$, $A \in \mathbb{R}^n \times \mathbb{R}^m$ is an $n \times m$ matrix, $b \in \mathbb{R}^m$ and $\alpha, \beta \in \mathbb{R}$. It is assumed that $d^T x + \beta \neq 0$.

Relation of LFP with LP

There is a close relation of LFP with LP. We have discussed this in details as follows[5].

Case 1: If $d = 0$ and $\beta = 1$ then (2.1) transforms into an LP. Then (2.1) can be written as follows.

$$Maximize \quad z = c^T x + \alpha$$
$$subject \ to$$
$$Ax \le b$$
$$x \ge 0$$

Case 2: If $d = 0$ and $\beta \neq 1$ then (2.1) transforms into the following LP.

$$Maximize \quad z = \frac{c^T}{\beta} x + \frac{\alpha}{\beta}$$
$$subject \ to$$
$$Ax \le b$$
$$x \ge 0$$

Case 3: If $c = 0$ then (2.1) becomes a linear function as follows.

$$z = \frac{\alpha}{d^T x + \beta}$$

If $c \neq 0$ and $d \neq 0$ then these methods will not be applicable. For solving such a case we have used idea of Dinkelbach's algorithm and have developed an AMPL code.

III. Existing Techniques

In this section, we have discussed some existing techniques for solving LFP. These techniques have been discussed below.

Hasan and Acharjee's method (2011)

Hasan and Acharjee considered the LFP problem (2.1), (2.2), (2.3) on the basis of the following assumptions[5].

(i)   The feasible region X is non empty and bounded and $d^T x + \beta > 0$.

(ii) $p = (c^T - d^T \frac{\alpha}{\beta}), y = \frac{x}{d^T x + \beta}$ and $g = \frac{\alpha}{\beta}$.

Then the objective function of the LFP becomes $z = py + g$.

They calculated variable values of LFP by $x = \frac{\beta y}{1 - d^T y}$. Although this technique is very interesting but it's laborious and time consuming because one has to do a lots of calculations to convert LFP into LP.

Charnes and Copper's method (1962)

Charnes and Cooper considered LFP problem (2.1), (2.2), (2.3) on the basis of the following assumptions[6].

(i) $c^T x + \alpha$ and $d^T x + \beta$ cannot vanish simultaneously.

(ii)   The feasible region X is non empty and bounded.

The authors used the transformation $y = tx, \ t \geq 0$ and transformed LFP as follows.

$$Maximize \ z = c^T y + \alpha t$$
$$subject \ to$$
$$Ay - bt \leq 0$$
$$d^T y + \beta z = 1$$
$$y \geq 0, z \geq 0$$

The above problem arises for $d^T x + \beta > 0$. But if $d^T x + \beta < 0$ then LFP have to transform as follows.

$$Minimize \ z = -c^T y - \alpha t$$
$$subject \ to$$
$$Ay - bt \leq 0$$
$$-d^T y - \beta z = 1$$
$$y \geq 0, z \geq 0$$

Bitran and Novae's Method (1972)

In this section, we have summarized Bitran and Novae's method. Assuming that the constraints set is nonempty and bounded and the denominator $d^T x + \beta > 0$ for all feasible solutions, the authors proceeds as follows[7].

(i)   Convert the LFP into a sequence of LP.
(ii)   Then solve these LPs until two of them give identical solution.

In the next section, we have discussed the steps of Dinkelbach's algorithm.

IV. Dinkelbach's Algorithm

In the current section, we have discussed Dinkelbach's algorithm[4]. We have discussed the algorithm by considering the LFP presented in Section III. It consists of the following steps.

Step 1 : Choose $x_i^{(0)} = 0, i = 1, ..., n$, and consider iteration $k = 1$. Then compute the following.

$$\lambda^{(k)} = \frac{c^T x_i^{(k-1)} + \alpha}{d^T x_i^{(k-1)} + \beta}$$

Step 2: Solve the following LP.

$$Maximize \ F(\lambda^{(k)})$$
$$subject \ to$$
$$Ax_i \leq b$$
$$x_i \geq 0$$

Where $F(\lambda^{(k)}) = (c^T x_i + \alpha) - \lambda^{(k)}(d^T x_i + \beta)$.

Step 3: If $F(\lambda^{(k)}) = 0$ then $x_i^k, i = 1, .., n$ is the required optimal solution.

Step 4: If $F(\lambda^{(k)}) \neq 0$ then repeat the Steps 1 to 2.

V. Computer Code

In the current section, we have presented our developed computer code. We have developed this code by using a mathematical programming language[8] (AMPL). It has three different parts these are AMPL model file, AMPL data file and AMPL run file. In this paper, we have presented model file and run file only. These are presented below.

```
#------------------------------------------------------------------
#AMPL Model File
#------------------------------------------------------------------
param n;              # no of variables
param m;              # no of rows
param alpha;          # numerator consatant
param beta;           # denominator consatant
param lembda;         # a parameter
param c{i in 1..n};   #numerator's coefficients
param d{i in 1..n};   #denominator's coefficients
param a{i in 1..m, j in 1..n};   # constraint's coefficients
param b{i in 1..m};   # r.h.s constants
var x{i in 1..n}>=0;  # no of variables
maximize obv: sum{i in 1..n} (c[i]*x[i])+alpha-lembda*(sum{i in 1..n}(d[i]*x[i])+beta);
subject to cost{j in 1..m}: sum{i in 1..n} a[j,i]*x[i]<=b[j];
subject to limit{i in 1..n}:x[i]>=0;


#------------------------------------------------------------------
# AMPL Run File
#------------------------------------------------------------------
reset;
param iteration default 0;
```

param stop;
param p;
param q;
param u;
problem prob: x, obv, cost, limit;
let stop:=0;
for{i in 1..n}
{let x[i]:=0;
}
repeat {
let p:= sum{i in 1..n}(c[i]*x[i])+alpha;
let q:= sum{i in 1..n}(d[i]*x[i])+beta;
let lembda:= p/q;
solve prob;
let u:=abs(obv-lembda);
if stop=u then break;
let stop:=u;
let iteration:=iteration+1;
display iteration, lembda, x ;}

VI. Numerical Examples

In this section, we have demonstrated our developed computer technique by using numerous examples.

Numerical Example 1

This example has taken from Erik B. Bajalinov[4].

$$Maximize \quad Q(x) = \frac{8x_1+9x_2+4x_3+4}{2x_1+3x_2+2x_3+7}$$

subject to

$$x_1 + x_2 + 2x_3 \leq 3$$
$$2x_1 + x_2 + 4x_3 \leq 4$$
$$5x_1 + 3x_2 + x_3 \leq 15$$
$$x_1, x_2, x_3 \geq 0$$

Then we have to determine the followings.

$$\lambda^{(k)} = \frac{8x_1^{(k-1)} + 9x_2^{(k-1)} + 4x_3^{(k-1)} + 4}{2x_1^{(k-1)} + 3x_2^{(k-1)} + 2x_3^{(k-1)} + 7}$$

We start our calculations by considering $k = 1$ and $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$. Now we transform the problem into a single LP by converting $Q(x)$ as follows.

$F(\lambda^{(k)}) = (8x_1 + 9x_2 + 4x_3 + 4) - \lambda^{(k)}(2x_1 + 3x_2 + 2x_3 + 7)$

All the constraints remain same. Now we have presented details calculations into the following steps.

Iteration 1: $\lambda^{(1)} = \frac{4}{7}$. Corresponding LP has given below.

$$Maximize \ F(\lambda^{(1)}) = (8x_1 + 9x_2 + 4x_3 + 4) - \frac{4}{7}(2x_1 + 3x_2 + 2x_3 + 7)$$

$$= \frac{48}{7}x_1 + \frac{61}{7}x_2 + \frac{20}{7}x_3$$

subject to

$$x_1 + x_2 + 2x_3 \leq 3$$
$$2x_1 + x_2 + 4x_3 \leq 4$$
$$5x_1 + 3x_2 + x_3 \leq 15$$
$$x_1, x_2, x_3 \geq 0$$

Solving this problem by using **LINDO** we get $x_1 = 0, x_2 = 3, x_3 = 0$, and $F_{max} = \frac{153}{7} = 21.8571(approx)$.

Since $F_{max} \neq 0$ so we have to go to the next step.

Iteration 2: $\lambda^{(2)} = \frac{31}{16}$. Corresponding LP has given below.

$$Maximize \ F(\lambda^{(2)}) = (8x_1 + 9x_2 + 4x_3 + 4) - \frac{31}{16}(2x_1 + 3x_2 + 2x_3 + 7)$$

$$= \frac{33}{8}x_1 + \frac{51}{16}x_2 + \frac{1}{8}x_3 - \frac{153}{16}$$

subject to

$$x_1 + x_2 + 2x_3 \leq 3$$
$$2x_1 + x_2 + 4x_3 \leq 4$$
$$5x_1 + 3x_2 + x_3 \leq 15$$
$$x_1, x_2, x_3 \geq 0$$

Solving this problem by using **LINDO** we get $x_1 = 1, x_2 = 2, x_3 = 0$, and $F_{max} = \frac{15}{16} = 0.9375(approx)$. Since $F_{max} \neq 0$ so we have to go to the next step.

Iteration 3: $\lambda^{(3)} = 2$. Corresponding LP has given below.

$$Maximize \ F(\lambda^{(3)}) = (8x_1 + 9x_2 + 4x_3 + 4) - 2(2x_1 + 3x_2 + 2x_3 + 7)$$

$$= 4x_1 + 3x_2 - 10$$

subject to

$$x_1 + x_2 + 2x_3 \leq 3$$
$$2x_1 + x_2 + 4x_3 \leq 4$$
$$5x_1 + 3x_2 + x_3 \leq 15$$
$$x_1, x_2, x_3 \geq 0$$

Solving by using **LINDO** we get $x_1 = 1, x_2 = 2, x_3 = 0$, and $F_{max} = 0$. Since $F_{max} = 0$ then $x_1 = 1, x_2 = 2, x_3 = 0$ are the required optimal solutions of $Q(x)$. Then the objective function value of $Q(x)$ is $Q_{max} = 2$.

Now we have presented iteration by iteration AMPL output of numerical example 1 as follows.

```
#-------------------------------------------------------------------
# AMPL Output of Example 1
#-------------------------------------------------------------------
iteration = 1
obv = 21.8571
lembda = 0.571429
x [*] :=
1  0
2  3
3  0;
iteration = 2
obv = 0.9375
```

lembda = 1.9375
x [*] :=
1 1
2 2
3 0;
iteration = 3
obv = -7.99361e-15
lembda = 2
x [*] :=
1 1
2 2
3 0;

Here -7.99361e-15 means -7.99361×10^{-15} which is very close to zero.

Numerical Example 2

This example has also taken from Erik B. Bajalinov[4].

$$Maximize \quad Q(x) = \frac{6x_1 + 3x_2 + 6}{5x_1 + 2x_2 + 5}$$

subject to

$$4x_1 - 2x_2 \leq 20$$

$$3x_1 + 5x_2 \leq 25$$

$$x_1, x_2 \geq 0$$

Now consider $\lambda^{(k)} = \frac{6x_1^{(k-1)} + 3x_2^{(k-1)} + 6}{5x_1^{(k-1)} + 2x_2^{(k-1)} + 5}$. We have to start the iterations by considering $k = 1$ and $x_1^{(0)} = x_2^{(0)} = 0$. Then the objective function $Q(x)$ will be transformed into the following form.

$$F(\lambda^{(k)}) = (6x_1 + 3x_2 + 6) - \lambda^{(k)}(5x_1 + 2x_2 + 5)$$

Manual output and **AMPL** output of the Numerical example 2 has been presented in the following table.

Table 1. Output of numerical example 2

Iterations	Manual Output	AMPL output
1	Parameter, $\lambda^{(1)} = \frac{6}{5} = 1.2$ Maximum value of LP, $F_{max} = 3$ Variable values, $x_1 = 0$ $x_2 = 5$	obv = 3 lembda = 1.2 x [*] := 1 0 2 5;
2	Parameter, $\lambda^{(2)} = \frac{21}{15} = 1.4$ Maximum value of LP, $F_{max} = 0$ Variable values, $x_1 = 0$ $x_2 = 5$	obv= 8.88178e-16 lembda = 1.4 x [*] := 1 0 2 5;

Here 8.88178e-16 means 8.88178×10^{-16} which is very close to zero. From Table 1, we observe that at Iteration 2 value of the transformed LP becomes zero. Therefore $x_1 = 0, x_2 = 5$ are the optimal solutions and $Q_{max} = \frac{7}{5}$.

Numerical Example 3

This example has taken from Hasan[9] (2008).

$$Maximize \quad Q(x) = \frac{2x_1 + 3x_2}{x_1 + x_2 + 1}$$

subject to

$$x_1 + x_2 \leq 3$$

$$x_1 + 2x_2 \leq 3$$

$$x_1, x_2 \geq 0$$

Consider $\lambda^{(k)} = \frac{2x_1^{(k-1)} + 3x_2^{(k-1)}}{x_1^{(k-1)} + x_2^{(k-1)} + 1}$. To start iteration let $k = 1$, and $x_1^{(0)} = x_2^{(0)} = 0$. Then the objective function $Q(x)$ will be transformed into the following form.

$$F(\lambda^{(k)}) = (2x_1 + 3x_2) - \lambda^{(k)}(x_1 + x_2 + 1)$$

Manual output and **AMPL** output of the Numerical example 3 has presented in the following table.

Table 2. Output of numerical example 3

Iterations	Manual Output	AMPL output
1	Parameter, $\lambda^{(1)} = 0$ Maximum value of LP, $F_{max} = 6$. Variable values, $x_1 = 3, x_2 = 0$.	obv = 6 lembda = 0 x [*] := 1 3 2 0;
2	Parameter, $\lambda^{(2)} = \frac{3}{2} = 1.5$ Maximum value of LP, $F_{max} = \frac{3}{4} = 0.75$ Variable values, $x_1 = 0, x_2 = \frac{3}{2}$.	obv = 0.75 lembda = 1.5 x [*] := 1 0 2 1.5;
3	Parameter, $\lambda^{(3)} = \frac{9}{5} = 1.8$ Maximum value of LP, $F_{max} = 0$ Variable values, $x_1 = 0, x_2 = 1.5$.	obv = -2.22045e-16 lembda = 1.8 x [*] := 1 0 2 1.5;

Here -2.22045e-16 means -2.22045×10^{-16} which is very close to zero. Therefore $x_1 = 0, x_2 = \frac{3}{2}$ is the required optimal solutions and objective function value is $Q_{max} = \frac{9}{5}$.

Numerical Example 4

This example has taken from I.M. Stancu-Minasian[10].

$$Maximize \quad Q(x) = \frac{x_1 - 3x_2 + 3x_3}{x_1 + 2x_2 + x_3 + 1}$$

subject to

$$x_1 + x_2 - x_3 \leq 6$$
$$-2x_1 - x_2 + 2x_3 \leq 2$$
$$2x_1 - 3x_2 + x_3 \leq 3$$
$$x_1, x_2, x_3 \geq 0$$

Consider $\lambda^{(k)} = \frac{x_1^{(k-1)} - 3x_2^{(k-1)} + 3x_3^{(k-1)}}{x_1^{(k-1)} + 2x_2^{(k-1)} + x_3^{(k-1)} + 1}$. To start iteration let

$k = 1$, and $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$. Then the objective function $Q(x)$ will be transformed into the following form.

$$F(\lambda^{(k)}) = (x_1 - 3x_2 + 3x_3) - \lambda^{(k)}(x_1 + 2x_2 + x_3 + 1)$$

Manual output and **AMPL** output of the Numerical example 4 has been presented in the following table.

Table 3. Output of numerical example 4

Iterations	Manual Output	AMPL output
1	Parameter, $\lambda^{(1)} = 0$. Maximum value of LP, $F_{max} = \frac{94}{3}$ $= 31.33\ (approx)$ Variable values, $x_1 = \frac{37}{3}$ $x_2 = 14$ $x_3 = \frac{61}{3}$	obv = 31.3333 lembda = 0 x [*] := 1 12.3333 2 14 3 20.3333;
2	Parameter, $\lambda^{(2)} = \frac{94}{185}$ $= 0.5081\ (approx)$ Maximum value of LP, $F_{max} = \frac{147}{37} =$ $3.9729\ (approx)$ Variable values, $x_1 = \frac{2}{3}$ $x_2 = 0$ $x_3 = \frac{5}{3}$	obv = 3.97297 lembda = 0.508108 x [*] := 1 0.666667 2 0 3 1.66667;
3	Parameter, $\lambda^{(3)} = \frac{17}{10} = 1.7$ Maximum value of LP, $F_{max} = 0$ Variable values, $x_1 = \frac{2}{3}$ $x_2 = 0$ $x_3 = \frac{5}{3}$	obv = 4.44089e-16 lembda = 1.7 x [*] := 1 0.666667 2 0 3 1.66667;

Here 4.44089e-16 means 4.44089×10^{-16} which is very close to zero. Therefore, $x_1 = \frac{2}{3}, x_2 = 0, x_3 = \frac{5}{3}$ are the required optimal solutions of the given problem and $Q_{max} = \frac{17}{10}$.

In the next section, we have drawn a conclusion about our work.

VII. Conclusion

In this paper, we presented a computer oriented technique for solving a mathematical programming called LFP. We developed this technique by using idea of Dinkelbach's algorithm. We used AMPL to develop computer code and illustrated several problems to demonstrate the technique. We hope that this technique can be extended to solve other linear and non-linear programming problems.

References

1. Hillier, F. S., G. J., Lieberman, 2001. Introduction to Operations Research, MacGraw-Hill series in Industrial Engineering and Management Science.

2. Martos, B., 1964. Hyperbolic Programming, Naval Research Logistics Quarterly, **37**, 135-155.

3. Swarup, K., 1964. Linear Fractional Functional Programming, Operations Research, **13**(6), 1029-1036.

4. Bajalinov, E. B., 2003. Linear Fractional Programming: Theory, Methods, Applications, and Software, Boston: Kluwer Academic Publishers.

5. Hasan, M. B., S., Acharjee, 2011. Solving LFP by converting it into a Single LP, International Journal of Operations Research, **8**(3), 1-14.

6. Charnes, A., W.W., Cooper, 1973. An Explicit General Solution in Linear Fractional Programming, Naval Research Logistics Quarterly, **20**(3), 449-467.

7. Bitran, G. R., and A. G., Novaes, 1972. Linear Programming with Fractional Objective Function, University of Sao Paulo, Brazil, **9**,181-186.

8. www.ampls.com

9. Hasan, M. B., 2008. Solution of Linear Fractional Programming Problems through Computer Algebra, The Dhaka University Journal of Science, **57**(1), 23-28.

10. Stanchu-Minasian, I. M., Fractional Programming: Theory, Method, and Applications, Kluwer Academic Publishers.