Dhaka Univ. J. Sci. 60(2): 209-215, 2012 (July)

A Technique for Solving Special Type Quadratic Programming Problems

M. Babul Hasan

Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh

Email: mbabulhasan@yahoo.com

Received on 20. 07. 2011. Accepted for Publication on 22. 02. 2012.

Abstract

Because of its usefulness in production planning, financial and corporate planning, health care and hospital planning, quadratic programming (QP) problems have attracted considerable research and interest in recent years. In this paper, we first extend the simplex method for solving QP problems by replacing one basic variable at an iteration of simplex method. We then develop an algorithm and a computer technique for solving quadratic programming problem involving the product of two indefinite factorized linear functions. For developing the technique, we use programming language *MATHEMATICA*. We also illustrate numerical examples to demonstrate our technique.

Key Words: Quadratic programming, simplex method, algorithm, computer technique

I. Introduction

Non-linear Programming is a mathematical technique for determining the optimal solutions to many business problems. In a non-linear programming problem, either the objective function is non-linear, or one or more constraints have non-linear relationship or both.

General Non-linear Programming Problem

Let z be a real valued function of n variables defined

(a) $z = f(x_1, x_2, \dots \dots \dots x_n)$,

Let $\{b_1, b_2, \dots \dots \dots b_m\}$ be a set of constraints such that

(b)
$$\begin{cases} g^1(x_1, x_2, \dots \dots \dots x_n)\{\le, \ge \text{ or } =\}\, b_1 \\ g^2(x_1, x_2, \dots \dots \dots x_n)\{\le, \ge \text{ or } =\}\, b_2 \\ \vdots \\ g^m(x_1, x_2, \dots \dots \dots x_n)\{\le, \ge \text{ or } =\}\, b_m \end{cases}$$

where g^i's are real valued functions of n variables $x_1, x_2, \dots \dots \dots x_n$.

Finally, let

(c) $x_j \ge 0.\quad j = 1, 2, \dots \dots \dots m$

If either $f(x_1, x_2, \dots \dots \dots x_n)$ or

some $g^i(x_1, x_2, \dots \dots \dots x_n), i = 1,2, \dots \dots \dots m$; or both are non-linear, then the problem of determining the n-type $(x_1, x_2, \dots \dots \dots x_n)$ which makes z a maximum or minimum and satisfies (b) and (c), is called a general non-linear programming (GNLP) problem. An NLP may be non-linearly constrained or linearly constrained. If the objective of a linearly constrained NLP is a quadratic function then this problem is called quadratic programming (QP) problem.

That is, quadratic programming is concerned with the NLPP of maximizing (or minimizing) the quadratic objective function, subject to a set of linear inequality constraints.

Quadratic programming problem

Let x^T and $c \in \boldsymbol{R}^n$. Let Q be a symmetric $n \times n$ real matrix. Then,

Maximize $\quad f(x) - cx + \frac{1}{2}x^T Qx$

subject to the constraints $Ax \le b$ and $x \ge 0$

where $b^T \in \boldsymbol{R}^m$ and A in a $m \times n$ real matrix, is called a general quadratic programming problem.

Note: $x^T Qx$ represents a quadratic form. The reader may recall that a quadratic form $x^T Qx$ is said to be positive-definite (negative definite) if $x^T Qx > 0 (< 0)$ for $x \ne 0$ and positive-semi-definite (negative-semi-definite) if $x^T Qx \ge 0 (\le 0)$ for all x such that there is one $x \ne 0$ satisfying $x^T Qx = 0$.

It can easily be shown that

1. If $x^T Qx$ is positive-semi-definite (negative-semi-definite) then it is convex (concave) in x over all of $\boldsymbol{R}^n$, and

2. If $x^T Qx$ is positive-definite (negative-definite) then it is strictly convex (strictly concave) in x over all of $\boldsymbol{R}^n$.

These results help in determining whether the quadratic objective function $f(x)$ is concave (convex) and the implication of the same on the sufficiency of the Karush-Khun-Tucker conditioin for constrained maxima (minima) of $f(x)$.

*Author for correspondence

Special type Quadratic programming problem

In this chapter, we consider the quasi-concave quadratic programming problem (QP) subject to linear constraints as follows:

(QP):Maximize $z = (cx+\alpha)(dx+\beta)$ (1)

subject to $Ax \leq b$, $x \geq 0$ (2)

where A = $(a_1, a_2, \ldots \ldots, a_m, a_{m+1}, \ldots a_n)$ is a $m \times n$ matrix, b $\in \mathfrak{R}^m$, x, c , d $\in \mathfrak{R}^n$, $\alpha, \beta \in \mathfrak{R}$. Here we assume that

(i) $(cx+\alpha)$ and $(dx+\beta)$ are positive for all feasible solution.
(ii) The constraints set $x \in S = \{x : Ax = b, x \geq 0\}$ is non empty and bounded.

In general, a quadratic programming problem is solved starting with some x which satisfies the constraints but does not necessarily maximize Z. Then x is replaced by a series of other $x's$ (not necessarily basic feasible solutions) all of which satisfy the constraints until the required solution is obtained.There are many methods for solving (QP). Among them Wolf's [4] method, Swarup's [5] simplex type method and Gupta and Sharm's [1] 2-basic variable replacement method are noteworthy. In this paper, we develop a computer technique for solving special type (quasi-concave) of QP problem in which the objective function can be factorized. Then compare the values of the objective function at these points in order to choose the optimal solution. We will introduce a computational technique to solve the QP problems by using computer algebra Mathematica [11].

The rest of the paper is organized as follows.

In Section II, we present the simplex type method

of Swarup [5] with numerical example.

In Section III, we present our technique along with

algorithm. We then conclude the paper in Section IV.

Swarup's Simplex Type Method for Solving QP

Adding slack variables $X_{n+i}, i = 1,2,\ldots\ldots m$ to the constraints (2) of (QP), we have the standard (QP) as follows:

(QPI): Maximize $z = (cx+\alpha)(dx+\beta)$

subject to $Ax = b$, $x \geq 0$

Where $A, b, c, d, x, \alpha, \beta$ are defined as in Section 1.3

Let x_B be the initial basic feasible solution to the (QP). Then BX_B where $(b_1, b_2, b_3, \ldots \ldots b_m)$, $X_B \geq 0$,

$$=> X_B = B^{-1}b$$

Also let $Z^1 = c_B x_B + \alpha$, and $Z^2 = d_B x_B + \beta$

$$\therefore Z = Z^1 . Z^2$$

In addition we assume that $y_u = B^{-1}a_u$, $Z_u^1 = C_B y_u$ and $Z_u^2 = d_B y_u$

Now let $\hat{X}_B$ be another basic feasible solution where $\hat{B} = (\hat{b}_1, \hat{b}_2, \hat{b}_3, \ldots \hat{b}_m)$ is the basis in which b_r is replaced by a_u of A but not in B.Then the columns of $\hat{B}$ are given by

$\hat{b}_i = b_i$ for $i \neq r$, and $\hat{b}_r = a_u$

Then the new value of the basic variables in terms of the original ones and y_{iu} are as follows:

$\hat{X}_{B_i} = X_{B_i} - X_{B_r}(\frac{y_{iu}}{y_{ru}}, i \neq r)$, and $\hat{X}_{B_r} = \frac{X_{B_r}}{y_{ru}} - \theta_u$ (Say)

Where $a_u = \sum_{i=1}^{i=m} y_{iu}\, b_i$

New optimizing value

$$\hat{Z}^1 = \sum_{i=1}^{i=m} \hat{C}_{B_i}\hat{X}_{B_i} + \alpha$$

$$= \sum_{i=1}^{i=m} \hat{C}_{B_i}\hat{X}_{B_i} + \hat{C}_{B_r}\hat{X}_{B_r} + \alpha$$

$$= \sum_{i \neq r}^{i=m} \hat{C}_{B_i}(X_{B_i} - X_{B_r}\frac{y_{iu}}{y_{ru}}) + C_u\frac{X_{B_r}}{y_{ru}} + \alpha$$

where $\hat{C}_{B_i} = C_{B_i}$ and $\hat{C}_{B_r} = C_u$

$$-\sum_{i-1}^{i-m} C_{B_i}X_{B_i} - C_{B_r}X_{B_r} - \sum_{i=r}^{i-m} C_{B_i}X_{B_i}\frac{y_{iu}}{y_{ru}} + C_u\frac{X_{B_r}}{y_u} + \alpha$$

$$= Z^1 + C_u\frac{X_{B_r}}{y_{ru}} - \sum_{i=1}^{i=m} C_{B_r}\, y_{iu}\frac{X_{B_r}}{y_{ru}}$$

$$=> \hat{Z}_1 = Z^1 + (c_u - Z_u{}^1)\frac{X_{B_r}}{y_{ru}}$$

$$\therefore \hat{Z}_1 = Z^1 + (C_u - Z_u{}^1)\hat{X}_{B_r}$$

Similary, $\hat{Z}^2 = Z^2 + (d_u - Z_u{}^2)\hat{X}_{B_r}$

Hence

$$\hat{Z} = \hat{Z}_1\hat{Z}^2 = \{Z^1 + (C_u - Z_u{}^1)\hat{X}_{B_r}\}\{Z^2 + (d_u - Z_u{}^2)\hat{X}_{B_r}\}$$

Optimality condition

The value of the objective function will improve If $\hat{Z} > Z$

$$\Rightarrow \{Z^1 + (C_u - Z_u{}^1)\hat{X}_{B_r}\}\{Z^2 + (d_u - Z_u{}^2)\hat{X}_{B_r}\} > Z_1Z_2$$

$$\Rightarrow Z^2 + (C_u - Z_u{}^1)\hat{X}_{B_r} + Z^1 + (d_u - Z_u{}^2)\hat{X}_{B_r} + (C_u - Z_u{}^1)(d_u - Z_u{}^2)\hat{X}_{B_r}.\hat{X}_{B_r}$$

$$\Rightarrow Z^2(C_u - Z_u{}^1) + Z^1(d_u - Z_u{}^2) + \hat{X}_{B_r}(C_u - Z_u{}^1)(d_u - Z_u{}^2)\hat{X}_{B_r}.\hat{X}_{B_r} > 0$$

But for non degenerate case $\hat{X}_{B_r} > 0$

$$\therefore Z^2(C_u - Z_u{}^1) + Z^1(d_u - Z_u{}^2) + \theta_u(C_u - Z_u{}^1)(d_u - Z_u{}^2) > 0$$

where $\theta_u = \hat{X}_{B_r}$

Hence we must have,

$$R_j(say) = Z^2(C_j - Z_j{}^1) + Z^1(d_j - Z_j{}^2) + \theta_j(C_j - Z_j{}^1)(d_j - Z_j{}^2) > 0$$

Thus the solution can be improved until $R_j \leq 0$ for all $j = 1,2,\ldots..n$. Whenever all $R_j \leq 0$ for all $j = 1,2,\ldots..n$ at a simplex table, the solution becomes optimal and the process terminates.

Criterion1: (Choice of the entering variable)

To identify the entering variable, choose the uth column of A for which R_u is the greatest positiveR_j; j = 1, 2,…, n

Criterion 2: (Choice of the outgoing variable)

Choose X_{B_r} for which $\frac{x_{Br}}{y_{ru}} = \min\left(\frac{x_{B_i}}{y_{ui}}, y_{ui} > 0\right)$

Numerical Example

Maximize

$$Z = (2x_1 + 4x_2 + x_3 + 1)(x_1 + x_2 + 2x_3 + 2)$$

subject to $x_1 + 3x_2 \leq 4$, $2x_1 + x_2 \leq 3$, $x_2 + 4x_3 \leq 3$, $x_1, x_2, x_3 \geq 0$

Solution

Introduction the slack variables to the constraints we get the initial table as follows:

Table. 1. (Initial table)

C_B	d_B								b
		$C_j \rightarrow$	2	4	1	0	0	0	
		$d_j \rightarrow$	1	1	2	0	0	0	
		x_B	x_1	x_2	x_3	x_4	x_5	x_6	
0	0	x_4	1	3	0	1	0	0	4
0	0	x_5	2	1	0	0	1	0	3
0	0	x_6	0	1	4	0	0	1	3
$Z^1 = 1$	$Z^2 = 2$	$Z = 2$							
		$C_j - Z_j{}^1$	2	4	1	0	0	0	
		$d_j - Z_j{}^2$	1	1	3	0	0	0	
		θ_j	3/2	4/3	¾	-	-	-	
		R_j	8	43/3	11/2	-	-	-	

Table.2

C_B	d_B	C_j d_j	2 1	4 1	1 2	0 0	0 0	0 0	b
		x_B	x_1	x_2	x_3	x_4	x_5	x_6	
4 0 0	1 0 0	x_2 x_5 x_6	1/3 5/3 -1/3	1 0 0	0 0 4	1/3 -1/3 -1/3	0 1 0	0 0 1	4/3 5/3 5/3
$Z^1 = 19/3$	$Z^2 = 10/3$	$Z = 190/9$							
		$C_j - Z_j^1$ $d_j - Z_j^2$	2/3 2/3	0 0	1 2	-4/3 -1/3	0 0	0 0	
		θ_j R_j	1 62/9	- -	5/12 101/6	4 -43/9	- -	- -	

Table.3

C_B	d_B	C_j d_j	2 1	4 1	1 2	0 0	0 0	0 0	b
		x_B	x_1	x_2	x_3	x_4	x_5	x_6	
4 0 0	1 0 0	x_2 x_5 x_3	1/3 5/3	1 0 0	0 0 1	1/3 -1/3 -1/12	0 1 0	0 0 1/4	4/3 5/3 5/1
$Z^1 = 27/4$	$Z^2 = 25/6$	$Z = 225/8$							
		$C_j - Z_j^1$ $d_j - Z_j^2$	3/4 5/6	0 0	0 2	-5/4 -1/6	0 0	-1/4 -1/2	
		θ_j R_j	1 75/8	- -	- -	4 11/2	- -	- 101/24 -	

Table.4

C_B	d_B	C_j d_j	2 1	4 1	1 2	0 0	0 0	0 0	b
		x_B	x_1	x_2	x_3	x_4	x_5	x_6	
4 2 1	1 1 2	x_2 x_1 x_3	0 1 0	1 0 0	0 0 1	2/5 -1/5 -1/10	-1/5 3/5 1/20	0 0 1/4	1 1 1/2
$Z^1 = 15/2$	$Z^2 = 5$	$Z = 75/2$							
		$C_j - Z_j^1$ $d_j - Z_j^2$	0 0	0 0	0 0	-11/10 0	-9/20 -1/2	-1/4 -1/2	
		θ_j R_j	- -	- -	- -	5/2 -11/2	5/3 -45/8	2 -19/4	

Since all R_j is non positive in Table-4, it gives the global optimal solution.

Hence the optimal solution is
$x_1 = 1, x_2 = 1, x_3 = 1/2$ with global maximum, $Z = 75/2$.

III. Algorithm for Solving Special Type QP

In this section, we present the computational technique

for solving the QP problems. In 2000, Hasan et. al. [10] developed a computer oriented solution method for solving the LP problems. In this study, we will extend that method for solving the QP problems. We will see that, this method will help us to solve QP problems easily and to avoid the clumsy and time consuming methods of Swarup [5] and other methods.

Step 1: Express the QP to its standard form.

Step 2: Find all $m \times m$ sub-matrices of the coefficient matrix A by setting $n - m$ variables equal to zero.

Step 3: Test whether the linear system of equations has unique solution or not.

Step 4: If the system of linear equations has got any unique solution, find it.

Step 5: Dropping the solutions with negative elements, determine all basic feasible solutions.

Step 6: Calculate the values of the objective function for the basic feasible solution found in step 5.

Step7: For the maximization of QP the maximum value of Z is the optimal value of the objective function, and the basic feasible solution which yields the optimal value is the optimal Solution.

Computer code for solving special type QP

In this section, we present our method for solving QP problems using computer algebra Mathematica.

```
<< LinearAlgebra`MatrixManipulation`

Clear[basic, sset, AA, bb]
```

```
basicfeasible[AA_, bb_] :=
Block[{m, n, pp, ss, ns, B, v, vv, var, vplus, vzero, BB, RBB, sol, new, sset, bs},
 {m, n} = Dimensions[AA]; pp = Permutations[Range[n]];
   ss = Union[Table[Sort[Take[pp[[k]], m]], {k, 1, Length[pp]}]];
   ns = Length[ss]; B = {};
   For[k = 1, k <= ns, k = k + 1,
        v = Table[TakeColumns[AA, {ss[[k]][[j]]}], {j, 1, m}];
        vv = Transpose[Table[Flatten[v[[i]]], {i, 1, m}]];
            B = Append[B, vv]];
     var = Table[x[i], {i, 1, n}];
     vplus[k_] := var[[ss[[k]]]];
     vzero[k_] := Complement[var, vplus[k]];
     sset = {}; For[k = 1, k <= ns, k = k + 1, BB = B[[k]]; RBB = RowReduce[BB];
        If[RBB == IdentityMatrix[m], sol = LinearSolve[BB, bb], sol = {}];
        If[Length[sol] == 0 || Min[sol] < 0, new = {}, new = sol;
           sset = Append[sset, {vplus[k], new}]]];
     bs[k_] := Block[{u, v, w, zf1, f2},
             u = sset[[k, 1]]; v = sset[[k, 2]]; w = Complement[var, u];
             z = Flatten[ZeroMatrix[Length[w], 1]];
           f1 = Transpose[{u, v}]; f2 = Transpose[{w, z}];
           Transpose[Union[f1, f2]][[2]]];
     Table[bs[k], {k, 1, Length[sset]}]]
```

```
qpoptimal[AA_, bb_, cc_] := Block[{vertex, val, opt, pos, optsol, qpsoln},
                              vertex = basicfeasible[AA, bb];
                              val = Table[((vertex[[k]].c) + α) / ((vertex[[k]].d) + β),
    {k, 1, Length[vertex]}];
                          opt = Max[val];
                              pos = Flatten[Position[val, opt]];
                              optsol = vertex[[pos[[1]]]];
                              qpsoln = {optsol, opt};
                         Print["The optimal value of the objective function of the QP is ",
   qpsoln[[2]]];
                         Print["The optimal solution of the QP is ", qpsoln[[1]]]]
```

INPUT:

```
Clear[A, b, c]
A = {{1, 3, 0, 1, 0, 0}, {2, 1, 0, 0, 1, 0}, {0, 1, 4, 0, 0, 1}};
b = {4, 3, 3};  c = {2, 4, 1, 0, 0, 0};  d = {1, 1, 2, 0, 0, 0};
α = 1;  β = 2;
basicfeasible[A, b]
qpoptimal[A, b, c]
```

OUTPUT:

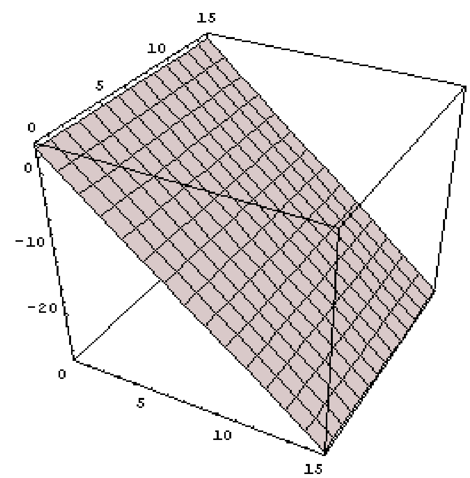
$$\left\{\left\{1, 1, \frac{1}{2}, 0, 0, 0\right\}, \{1, 1, 0, 0, 0, 2\}, \left\{\frac{3}{2}, 0, \frac{3}{4}, \frac{5}{2}, 0, 0\right\}, \left\{\frac{3}{2}, 0, 0, \frac{5}{2}, 0, 3\right\}, \left\{0, \frac{4}{3}, \frac{5}{12}, 0, \frac{5}{3}, 0\right\}, \left\{0, \frac{4}{3}, 0, 0, \frac{5}{3}, \frac{5}{3}\right\}, \left\{0, 0, \frac{3}{4}, 4, 3, 0\right\}, \{0, 0, 0, 4, 3, 3\}\right\}$$

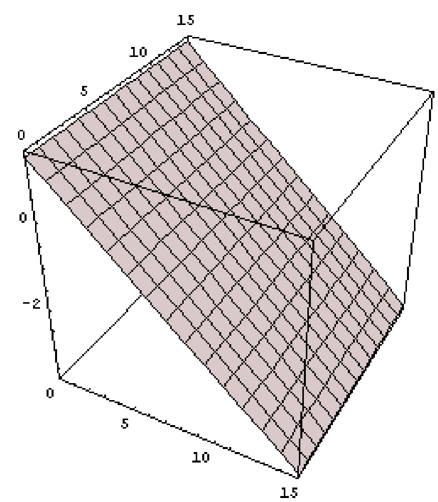
The optimal value of the objective function of the QP is $\frac{75}{2}$

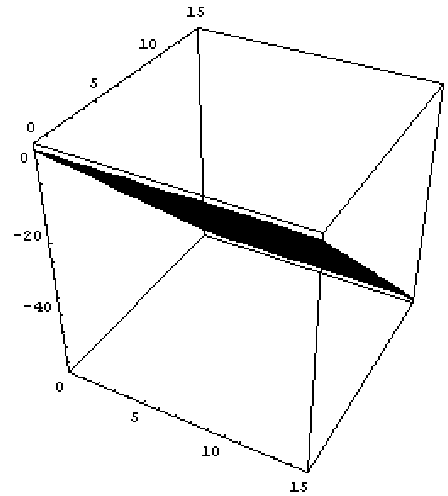
The optimal solution of the QP is $\left\{1, 1, \frac{1}{2}, 0, 0, 0\right\}$

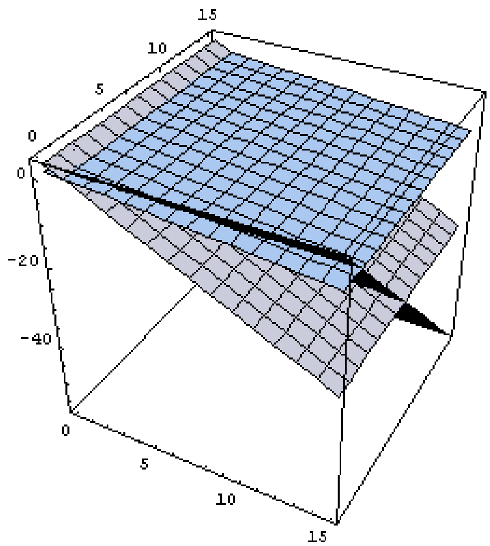
Graphical representation gives the actual concept of the problems. As, this is an optimization problem, we present graphical representation of this problem in the next section in order to achieve a more visual concept of the solutions of the given problem.

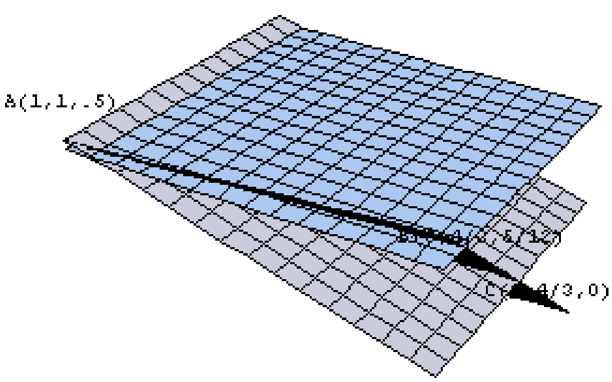
Graphical representation of numerical example of Section 2:











We observed that with Swarup's [5] method we experienced complex calculations for the simplex tables. For solving problems with "$\geq$ type" or "= type" constraints, one needs to solve that problem by Two-phase or Big-M simplex method which is clumsy and time consuming. But by our method, we just had to compute the coefficient matrix *A*, right hand side constant *b*, cost coefficient vectors *c* and *d* and the constants α and β in the same program and easily obtained the optimal solution. Our method is also able to solve degenerate and cycling problems easily whereas the simplex method is proved to be inconvenient for solving such problems.

IV. Conclusion

In this paper, we developed a computer technique for solving special type (Quasi-Concave) of quadratic programming in which the objective function can be factorized. We illustrated the solution procedure developed by us with a numerical example. We observed that the result obtained by our procedure is completely identical with that of the other methods which are laborious and time consuming and clumsy. We therefore, hope that our program can be used as an effective tool for solving special type QP problems in which the objective function can be factorized and hence our time and labor can be saved.

…………..

1. Swarup, K., P. k. Gupta and Man Mohan, 2003,"Tracts in Operations Research", Eleventh Thoroughly Revised Edition,ISBN:81-8054-059-6
2. Frederick S. Hillier and Gerald J. Lieberman, McGraw-Hill International Edition 1995, " Introduction to Operations Research"
3. Wayne L. Winston, (1994), "Application and algorithms," Dexbury press, Belmont, California, U.S.A.
4. Wolfe, P., (1959), "The simplex method for quadratic programming," *Econometrica*, **27**(3), 382-398.
5. Swarup, K. 1966, "Quadratic Programming" CCERO (Belgium), **8**(2), 132-136.
6. Swarup, K., 1964, "Linear fractional functional programming," Operations Research, **13 (6)**, 1029-1036.
7. Dantzig, G.B., 1962, "Linear programming and extension," Princeton University press, Princeton,N.J.
8. Hasan, M.B. and M.A.Islam, 2001, "Generalization of simplex method for solving fractional programming problems," *The Dhaka University Journal of Science*, **53**(1), 111-118.
9. Hasan, M.B. (2008), "Solution of Linear Fractional Programming Problems through Computer Algebra", *The Dhaka University Journal of Science*, **57**(1),
10. Hasan, M. B., A.F.M.K Khan and M. A. Islam, 2001, "Basic solutions of linear system of equations through computer algebrs," Ganit, Journal of Bangladesh Mathematical Society, **21 (1)**, 39-45.
11. Wolfram, S. 2000, "Mathematica," Addision-Wesly Publishing Company, Melno Park, California, New York.