Dhaka Univ. J. Sci. 63(2): 111-117, 2015 (July)

A Proposed Technique for Solving Quasi-Concave Quadratic Programming Problems with Bounded Variables

M. Asadujjaman* **and M. Babul Hasan**

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

(Received : 2 December 2014; Accepted : 12 March 2015)

Abstract

In this paper, a new method is proposed for finding an optimal solution to a Quasi-Concave Quadratic Programming Problem with Bounded Variables in which the objective function involves the product of two indefinite factorized linear functions and constraints functions are in the form of linear inequalities. The proposed method is mainly based upon the primal dual simplex method. The Linear Programming with Bounded Variables (LPBV) algorithm is extended to solve quasi-concave Quadratic Programming with Bounded Variables (QPBV). For developing this method, we use programming language *MATHEMATICA*. We also illustrate numerical examples to demonstrate our method.

Keywords: Quadratic Programming, Quasi-Concave Quadratic Programming, Bounded Variable, Lower & Upper Bound.

I. Introduction

In mathematics, non-linear programming (NLP) is the process of solving an optimization problem defined by a system of equalities and inequalities, collectively termed constraints, over a set of unknown real variables, along with an objective function to be maximized or minimized, where some of the constraints or the objective function is non-linear. It is the sub-field of mathematical optimization that deals with problems that are not linear. A quadratic programming (QP) is a special type of mathematical optimization problem. It is the problem of optimizing (minimizing or maximizing) a quadratic function of several variables subject to linear constraints on these variables. Because of its usefulness in Producing Planning, Financial and Corporate Planning, Health Care and Hospital Planning and Engineering, QP is viewed as a discipline in Operational Research and it has become a fertile area in the field of research in recent years. More importantly, though, it forms the basis of several general non-linear programming algorithms. A large number of algorithms for solving QP problems have been developed. Some of them are extensions of the simplex method and others are based on different principles. In the conversance, a great number of methods (Wolfe[1], Beale[2], Frank and Wolfe[3], Shetty[4], Lemke[5], Best and Ritter[6], Theil and van de Panne[7], Boot[8], Fletcher[9], Swarup[10], Gupta and Sharma[11], Moraru[12,13], Jensen and King[14], Bazaraa, Sherali and Shetty[15]) are designed to solve QP problems in a finite number of steps. Among them, Wolf's method[1], Swarup's simplex method[10] and Gupta and Sharma's method[11] are more popular than the other methods. The above mentioned articles deal with variables of the type ≥ 0 but no upper bound. But when considering real-world applications of QP, it may arrive that one or more unknown variables x_j not only have a non-negativity restriction but also have upper and lower bounds on them. In this case, the above mentioned articles did not consider the upper bounds on the variables. Andrew Whinston[16] developed a method for solving QP problems with bounded variables but not consider quasi-concave QP problems with bounded variables. Also this method is laborious. For this reason we try to find another procedure which takes less computational effort. So, in this paper, we proposed a new method for solving quasi-concave QPBV problems.

The proposed method depends mainly on solving quasi-concave QPBV problems in which the objective function involving the product of two indefinite factorized linear functions and constraints functions are in the form of linear inequalities. We use the concept of LPBV method to solve this problem. For developing this method, use programming language *MATHEMATICA*. We also illustrate numerical examples to demonstrate our method.

The rest of the paper is organized as follows. In section II, we discuss on glossary background of LPBV, QP, quasi-concave QP and quasi-concave QPBV problems. In section III, we discuss the existing method and existing algorithm for LPBV problems. In section IV, we discuss our proposed algorithm for quasi-concave QPBV problems and illustrate the solution procedure with a number of numerical examples. In section V, we develop a computer technique for this method by using programming language *MATHEMATICA* and solve the previous examples through the computer technique. Finally, we draw a conclusion in section VI.

II. Preliminaries

In this section, we briefly discuss definitions of LPBV, QP, quasi-concave QP and quasi-concave QPBV problems.

LPBV Problems

In Linear Programming (LP) models[17], variables may have explicit positive upper and lower bounds. For example, in production facilities, lower and upper bounds can represent the minimum and maximum demands for certain products. Bounded variables also arise prominently in the course of solving integer programming problems by the branch and bound algorithm.

Consider the following LP problems,

$$\text{Maximize,}\quad Z = CX$$

$$\text{Subject to,}\quad (A, I)X = b \tag{1}$$

$$L \leq X \leq U \tag{2}$$

where, $U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n+m} \end{pmatrix}$ & $L = \begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_{n+m} \end{pmatrix}$, $U \geq L \geq 0$.

*Author for correspondence. *email:* asad132@du.ac.bd

The elements of L and U for an unbounded variable 0 and ∞.

QP Problems

The general QP problem can be written as

$$\text{Maximize,} \quad Z = cx + \frac{1}{2}x^T Qx$$

$$\text{Subject to,} \quad Ax \leq b \text{ and } x \geq 0$$

Where c is an n-dimensional row vector describing the coefficients of the linear terms in the objective function, and Q is an $(n \times n)$ symmetric real matrix describing the coefficients of the quadratic terms. If a constant term exists it is dropped from the model. As in LP, the decision variables are denoted by the n-dimensional column vector x, and the constraints are defined by an $(m \times n)$ A matrix and an m-dimensional column vector b of right-hand side coefficients. We assume that a feasible solution exists and that the constraints region is bounded. When the objective function Z is strictly convex for all feasible points the problem has a unique local maximum which is also the global maximum. A sufficient condition to guarantee strictly convexity is for Q to be positive definite.

Quasi-Concave QP Problems

In this paper, we consider the quasi-concave QPBV problems subject to linear constraints.

The quasi-concave QP problems[18] can be written as

$$\text{Maximize,} \quad Z = (cx + \alpha)(dx + \beta)$$

$$\text{Subject to,} \quad Ax \leq b \text{ and } x \geq 0$$

where, A is an $(m \times n)$ matrix, $b \in \Re^m$, and x, c, $d \in \Re^n$ and α, $\beta \in \Re$. Here we assume that

(i) $(cx + \alpha)$ and $(dx + \beta)$ are positive for all feasible solution.

(ii) The constraints set $S = \{ x : Ax = b, x \geq 0 \}$ is non-empty and bounded.

Quasi-Concave QPBV Problems

Let us consider the general quasi-concave QPBV problem

Max, $Z = z^1(x) \cdot z^2(x)$

$$= \left(\sum_{j=1}^{n} c_j x_j + \alpha \right) \cdot \left(\sum_{j=1}^{n} d_j x_j + \beta \right) \qquad (3)$$

Subject to,

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1,2,\cdots\cdots,m \qquad (4)$$

$$l_j \leq x_j \leq u_j, \quad j = 1,2,\cdots\cdots,n \qquad (5)$$

where, $l_j \leq u_j$, $j = 1,2,\cdots\cdots\cdots,n$. Here l_j and u_j are usually called lower-bound and upper-bound of the constraints.

Let us assume that $z^1(x)$, $z^2(x) > 0$ for all $x = (x_1, x_2, \cdots\cdots, x_m, x_{m+1}, \cdots\cdots, x_n)^T \in S$, where S denotes a feasible set defined by the constraints (4) and (5). Also assume that S is non-empty and bounded.

III. Existing Method for LPBV Problems

In this section, we discuss the existing method[17] and existing algorithm[17,19] for LPBV problems.

One can solve LPBV problems by regular simplex method by considering the lower and upper bound constraints explicitly which is not computationally efficient as the number of constraints as well as the number of variables become large and studied LP problems with upper bounded variables, which uses smaller basis to solve LPBV problems. In which case, from (1) and (2), the constraints are put in the form,

$$(A, I)X = b$$
$$X + X' = U$$
$$X - X'' = L$$
$$X, X', X'' \geq 0$$

Where X' and X'' are slack and surplus variables. This problem includes $3(m + n)$ variables and $(3m + 2n)$ constraints equations. However, the size can be reduced considerably through the use of special techniques that ultimately reduce the constraints to the set $(A, I)X = b$.

First, we consider the lower-bounds. Given $X \geq L$, we can use the substitution $X = L + X''$, $X'' \geq 0$, $L \geq 0$. Throughout and solve the problem in terms of X''. The original X is determined by back-substitution which is legitimate because it guarantees that $X = L + X''$ will remain non-negative for all $X'' \geq 0$. Next, we consider the upper-bounding constraints, $X \leq U$. The idea of direct substitution (i.e. $X = U - X'$, $X' \geq 0$) is not correct because back substitution, $X = U - X'$, does not ensure that X will remain non-negative. This difficulty is over come by using a simplex method variation that accounts for the upper bounds implicitly. Define the upper bounded LP model as

$$\text{Maximize, } Z = \{CX | (A, I)X = b, 0 \leq X \leq U\}$$

The bounded primal simplex method uses only the constraints $(A, I)X = b, X \geq 0$, while accounting for $X \leq U$ implicitly by modifying the simplex feasibility condition. Let $X_B = B^{-1}b$ be a current basic feasible solution of $(A, I)X = b, X \geq 0$ and suppose that according to the regular optimality condition, P_j is the entering vector. In developing the new feasibility condition, two main points must be considered. First one, the non-negativity and upper-bound constraints for the entering variable and secondly, for those basic variables that may be affected by introducing the entering variables.

Existing Algorithm for LPBV Problems

Step 1: If R.H.S of any constraint is negative, make it positive by multiplying the constraint by -1.

Step 2: Convert the inequalities of the constraints into equations by the addition of suitable slacks and/or surplus variables and obtain an initial basic feasible solution.

Step 3: If any variable is at a positive lower bound, it should be substituted at its lower bound.

Step 4: Calculate the net evaluation $C_j - Z_j$. For a maximization problem if $C_j - Z_j \leq 0$ for the non-basic variables at their upper bound, optimum basic feasible solution is attained. It not, go to step-5. Reverse is true for a minimization problem.

Step 5: Select the most positive $C_j - Z_j$.

Step 6: Let x_j be a non-basic variable at zero level which is selected to enter the solution. Compute the quantities,

$$x_j \leq \theta_1 = \min_i \left\{ \frac{(B^{-1}b)_i}{(B^{-1}P_j)_i} \middle| (B^{-1}P_j)_i > 0 \right\}$$

$$x_j \leq \theta_2 = \min_i \left\{ \frac{(B^{-1}b)_i - (U_B)_i}{(B^{-1}P_j)_i} \middle| (B^{-1}P_j)_i < 0 \right\}$$

Final condition is satisfied simply it $x_j \leq u_j$ and $\theta = \min(\theta_1, \theta_2, u_j)$, where $\theta =$ value of the entering variable and u_j is the upper bound for the variable x_j. Let $(X_B)_r$ be the leaving variable corresponding to $\theta = \min(\theta_1, \theta_2, u_j)$ and then we have the following rules:

Rule 1: If $\theta = \theta_1$, $(X_B)_r$ leaves the basic solution (because non-basic) at level zero and x_j enter by using the regular row operation of the simplex method.

Rule 2: If $\theta = \theta_2$, $(X_B)_r$ leaves the basic solution at level zero and x_j enters then $(X_B)_r$ being non-basic at its upper bound must be substituted out by using $(X_B)_r = (U_B)_r - (X'_B)_r$, where $0 \leq (X'_B)_r \leq (U_B)_r$.

Rule 3: If $\theta = u_j$, x_j is substituted at its upper bound $u_j - x'_j$ but remain non-basic.

A tie among θ_1, θ_2 and u_j may be broken arbitrarily. However, it is preferable to implement the rule for $\theta = u_j$ because it entails less computation.

In the next section, we will develop a method for solving quasi-concave QPBV problems and also illustrate the solution procedure with a number of numerical examples.

IV. Proposed Algorithm for Quasi-Concave QPBV Problems

In this section, we propose an algorithm on quasi-concave QPBV problem and also include numerical examples to demonstrate our method.

Step 1: If the R.H.S. of any constraints is negative make it positive by multiplying the constraint by -1.

Step 2: Convert the inequalities of the constraints into equations by the addition of suitable slacks and/or surplus variables. If the constraint set is in a canonical form, then go to step-3. If the constraint set is not in a canonical form, then go to step-9.

Step 3: If any variable is at positive lower bound, it should be substituted at its lower bound.

Step 4: Now, compute z^1, z^2, relative profit factor $(c_j - z_j^1)$, relative cost factor $(d_j - z_j^2)$ and the ratio Δ_j, where

$$z^1 = c_B x_B + \alpha$$
$$z^2 = d_B x_B + \beta$$
$$z_j^1 = c_B a_j$$
$$z_j^2 = d_B a_j$$
$$\text{and } \Delta_j = z^2(c_j - z_j^1) - z^1(d_j - z_j^2)$$

Step 5: For maximization problem, if $\Delta_j \leq 0$ for all non-basic variables at their upper bound then optimal solution is attained. If, not go to step-6.

Step 6: Select the most positive Δ_j.

Step 7: Let x_j be the non-basic variable at zero level, which is selected to enter the solutions. Compute the quantities,

$$\theta_1 = \min_i \left\{ \frac{(x_B)_i^*}{\alpha_i^j} \middle| \alpha_i^j > 0 \right\} \quad or$$

$$\theta_1 = \min_i \left\{ \frac{(x_B)_i^*}{\alpha_i^j} \middle| (x_B)_i^* < 0 \ \& \ \alpha_i^j < 0 \right\}$$

$$\theta_2 = \min_i \left\{ \frac{(x_B)_i^* - (U_B)_i^*}{\alpha_i^j} \middle| \alpha_i^j < 0 \right\}$$

$$\theta = \min(\theta_1, \theta_2, U_j)$$

Where $\theta =$ value of the entering variable and U_j is the upper bound for the variable x_j.

Step 8: Set $(x_B)_r$ be the leaving variable corresponding to $\theta = \min(\theta_1, \theta_2, U_j)$ and then follow the following rules:

Rule 1: If $\theta = \theta_1$, $(x_B)_r$ leaves the basic solution at level zero and x_j enter by using the regular row operation of the simplex method.

Rule 2: If $\theta = \theta_2$, $(x_B)_r$ leaves the basic solution at level zero and x_j enters then $(x_B)_r$ being non-basic at its upper bound must be substituted out by using $(x_B)_r = (U_B)_r - (x'_B)_r$, where $0 \leq (x'_B)_r \leq (U_B)_r$.

Rule 3: If $\theta = U_j$, x_j is substituted at its upper bound $U_j - x'_j$ but remain non-basic.

A tie among θ_1, θ_2 and U_j may be broken arbitrarily. However, it is preferable to implement the rule for $\theta = U_j$ because it entails less computation.

Step 9: If the constraints set is not in a canonical form then follows the following sub-steps:

Sub-step 1: Introduce artificial variables wherever it is required. Consider all variables are non-negative.

Sub-step 2: Then write it as an artificial linear objective function as in minimization type (minimization: $w_1 + w_2 + \cdots$). In phase-I, solve the problem as a regular linear program.

Sub-step 3: Compute relative profit factor $c_j - z_j$.

Sub-step 4: For minimization problem, if $c_j - z_j \geq 0$ for all non-basic variables and the objective function (i.e. minimization: $w_1 + w_2 + \cdots$) equal to zero and also all artificial variables leave the basis then the original quasi-concave QPBV problem has a basic feasible solution. If, not then the problem has no optimal solution.

Sub-step 5: When it is feasible then remove all columns corresponding to the artificial variables and construct a new table to solve original quasi-concave QPBV problem with initial solution found at the end of phase-I. Then, repeat step 3 to step 8.

Numerical Example 1

Consider the following quasi-concave QPBV problem:

$$\text{Max, } Z = (x_1 + 3x_2 + 6) \cdot (2x_1 + 3x_2 + 12)$$
$$\text{Subject to, } \quad x_1 + 2x_2 \geq 10$$
$$2x_1 + 3x_2 \leq 60$$
$$5 \leq x_1 \leq 15, \quad 4 \leq x_2 \leq 30$$

Solution: Using Our Proposed Method

In our problem, the constraints are not in a canonical form. So apply step 9. Then our problem becomes and also we get the following simplex table.

Min, $U = w$

$$\text{Subject to, } \quad x_1 + 2x_2 - s_1 + w = 10$$
$$2x_1 + 3x_2 + s_2 = 60$$
$$x_1, x_2, s_1, s_2, w \geq 0$$

Table 1. Final table for finding basic variables

	c_j	0	0	0	0	1	
c_B	Basis	x_1	x_2	s_1	s_2	w	b
0	x_2	$\cdot 5$	1	$-\cdot 5$	0	$\cdot 5$	5
0	s_2	$\cdot 5$	0	$1 \cdot 5$	1	$-1 \cdot 5$	45
$c_j - z_j$		0	0	0	0	1	$U = 0$

Since all $c_j - z_j \geq 0$ and Min $U = 0$ and all artificial variables leave the basis. So the original quasi-concave QPBV problem has a basic feasible solution. After the above calculation, we take

$$x_1 = 5 + y_1, 0 \leq y_1 \leq 10 \quad \text{and}$$
$$x_2 = 4 + y_2, 0 \leq y_2 \leq 26.$$

Now, solve the original quasi-concave QPBV problem with initial solution found at the end of phase-I. Then the original quasi-concave QPBV problem becomes

Max, $Z = (y_1 + 3y_2 + 23) \cdot (2y_1 + 3y_2 + 34)$

$$\text{Subject to, } \quad \frac{1}{2}y_1 + y_2 - \frac{1}{2}s_1 = -\frac{3}{2}$$
$$\frac{1}{2}y_1 + \frac{3}{2}s_2 = \frac{85}{2}$$

$0 \leq y_1 \leq 10, 0 \leq y_2 \leq 26$ and $s_1, s_2 \geq 0$

Table 2. Initial table

c_B ↓	d_B ↓	$c_j \rightarrow$	1	3	0	0
		$d_j \rightarrow$	2	3	0	0
		Basis ↓	y_1	y_2	s_1	s_2
3	3	$y_2 = -3/2$	$1/2$	1	$-1/2$	0
0	0	$s_2 = 85/2$	$1/2$	0	$3/2$	1
$z^1 = 14$	$z^2 = 25$	$Z = 350$				
		$c_j - z_j^1$	$-1/2$	0	$3/2$	0
		$d_j - z_j^2$	$1/2$	0	$3/2$	0
		Δ_j	$-29/2$	0	$\mathbf{33/2}$ ↑	0

Here, $\theta_1 = \min\{\frac{-3/2}{-1/2}, \frac{85/2}{3/2}\} = 3$, $\theta_2 = \infty$, since $s_1 \geq 0$. So $\theta = \min\{\theta_1, \theta_2, U_1\} = 3 = \theta_1$.

Table 3.

c_B ↓	d_B ↓	$c_j \rightarrow$	1	3	0	0
		$d_j \rightarrow$	2	3	0	0
		Basis ↓	y_1	y_2	s_1	s_2
0	0	$s_1 = 3$	-1	-2	1	0
0	0	$s_2 = 38$	2	3	0	1
$z^1 = 23$	$z^2 = 34$	$Z = 782$				
		$c_j - z_j^1$	1	3	0	0
		$d_j - z_j^2$	2	3	0	0
		Δ_j	-12	$\mathbf{33}$ ↑	0	0

Here, $\theta_1 = 12 \cdot 66$, $\theta_2 = \infty$, since $s_1 \geq 0$. So $\theta = \min\{\theta_1, \theta_2, U_2\} = 12 \cdot 66 = \theta_1$.

Table 4. Optimal table

c_B ↓	d_B ↓	$c_j \rightarrow$	1	3	0	0
		$d_j \rightarrow$	2	3	0	0
		Basis ↓	y_1	y_2	s_1	s_2
0	0	$s_1 = 85/3$	$1/3$	0	1	$2/3$
3	3	$y_2 = 38/3$	$2/3$	1	0	$1/3$
$z^1 = 61$	$z^2 = 72$	$Z = 4392$				
		$c_j - z_j^1$	-1	0	0	-1
		$d_j - z_j^2$	0	0	0	-1
		Δ_j	-72	0	0	-11

Since all $\Delta_j \leq 0$ in Table-4, this table gives the optimal solution. The optimal solution in term of the original variables x_1, x_2 is found as follows: $x_1 = 5 + y_1 = 5 + 0 = 5$ and $x_2 = 4 + y_2 = 4 + (38/3) = 50/3$ with $Z_{max} = 4392$.

Numerical Example 2

Consider the following quasi-concave QPBV problem:

$$\text{Max, } Z = (5x_1 + x_2 + 10) \cdot (4x_1 + 2x_2 + 12)$$
$$\text{Subject to, } \quad 5x_1 + x_2 + x_3 = 20$$
$$4x_1 - x_3 + x_4 = 14$$

$2 \leq x_1 \leq 5, 4 \leq x_2 \leq 12, 0 \leq x_3 \leq 25, 0 \leq x_4 \leq 18$

Solution: Using Our Proposed Method

Since x_1 and x_2 has positive lower bound, it must be substituted at its lower bound. Let $x_1 = 2 + y_1$, $0 \le y_1 \le 3$ and $x_2 = 4 + y_2$, $0 \le y_2 \le 8$. Then our problem becomes and also we get the following simplex table.

$$\text{Max, } Z = (5y_1 + y_2 + 24) \cdot (4y_1 + 2y_2 + 28)$$
$$\text{Subject to, } \quad 5y_1 + y_2 + x_3 = 6$$
$$4y_1 - x_3 + x_4 = 6$$

$$0 \le y_1 \le 3, 0 \le y_2 \le 8, 0 \le x_3 \le 25, 0 \le x_4 \le 18$$

Table 5. Initial table

c_B ↓	d_B ↓					
		$c_j \rightarrow$	5	1	0	0
		$d_j \rightarrow$	4	2	0	0
		Basis ↓	y_1	y_2	x_3	x_4
1	2	$y_2 = 6$	5	1	1	0
0	0	$x_4 = 6$	4	0	-1	1
$z^1 = 30$	$z^2 = 40$	$Z = 120$				
		$c_j - z_j^1$	0	0	-1	0
		$d_j - z_j^2$	-6	0	-2	0
		Δ_j	**180** ↑	0	20	0

Here, $\theta_1 = \min\{\frac{6}{5}, \frac{6}{4}\} = \frac{6}{5}$, $\theta_2 = \infty$, since $y_1 \ge 0$. So $\theta = \min\{\theta_1, \theta_2, U_1\} = \frac{6}{5} = \theta_1$.

Table 6. Optimal table

c_B ↓	d_B ↓					
		$c_j \rightarrow$	5	1	0	0
		$d_j \rightarrow$	4	2	0	0
		Basis ↓	y_1	y_2	x_3	x_4
5	4	$y_1 = 6/5$	1	$1/5$	$1/5$	0
0	0	$x_4 = 6/5$	0	$-4/5$	$-4/5$	0
$z^1 = 30$	$z^2 = 164/5$	$Z = 984$				
		$c_j - z_j^1$	0	0	-1	0
		$d_j - z_j^2$	0	$6/5$	$-4/5$	0
		Δ_j	0	-36	$-44/5$	0

Since all $\Delta_j \le 0$ in Table-6, this table gives the optimal solution. The optimal solution in term of the original variables x_1, x_2, x_3 and x_4 is found as follows: $x_1 = 2 + y_1 = 2 + (6/5) = 16/5$, $x_2 = 4 + y_2 = 4 + 0 = 4$, $x_3 = 0$ and $x_4 = 6/5$ with $Z_{max} = 984$.

V. Algorithm and Computer Technique

In this section, we present algorithm and computational technique for solving quasi-concave QPBV problems. Hasan[18] developed a computer oriented solution method for solving the LP problems. In this study, we extend that method for solving quasi-concave QPBV problems.

Algorithm for solving Quasi-Concave QPBV problems

Step 1: Express the quasi-concave QPBV problem to its standard form.

Step 2: Find an $m \times m$ sub-matrix 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 linear system of 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 solutions found in step-5.

Step 7: For maximization of quasi-concave QPBV problem, the maximum value of Z is the optimal value of the objection function and the basic feasible solution which yields the optimal value is the optimal solution.

Computer code for solving Quasi-Concave QPBV problems

In this section, we present a computer technique for solving quasi-concave QPBV problems using the programming language *MATHEMATICA*[20,21].

```
<<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 quasi-concave QP is ", qpsoln[[2]]];
Print ["The optimal solution of the quasi-concave QP is ", qpsoln[[1]]]]
```

Numerical Examples

In this section, solve the same problems which were solved in section IV by above computer technique.

Input for Numerical Example 1

```
A= {{1,2,-1,0,0,0,0,0},{2,3,0,1,0,0,0,0},
    {1,0,0,0,-1,0,0,0},{1,0,0,0,0,1,0,0},
    {0,1,0,0,0,0,1,0},{0,1,0,0,0,0,0,1}};
B = {10,60,5,15,4,30};
c = {1,3,0,0,0,0,0,0};
d = {2,3,0,0,0,0,0,0};
α = 6;
β = 12;
basicfeasible[A,b]
qpoptimal[A, b, c]
```

Output for Numerical Example 1

```
The possible all basic solution is:
{{15,4,13,18,10,0,0,26},{5,4,3,38,0,10,0,26
},{15,10,25,0,10,0,6,20},{5,50/3,85/3,0,0,1
0,38/3,40/3}}
The optimal value of the objective function
of the quasi-concave QP is   4392
The optimal solution of the quasi-concave
QP is {5,50/3,85/3,0,0,10,38/3,40/3}
```

Input for Numerical Example 2

```
A = {{5,1,1,0,0,0,0,0,0,0},
    {4,0,-1,1,0,0,0,0,0,0},
    {1,0,0,0,-1,0,0,0,0,0},
    {1,0,0,0,0,1,0,0,0,0},
    {0,1,0,0,0,0,-1,0,0,0},
    {0,1,0,0,0,0,0,1,0,0},
    {0,0,1,0,0,0,0,0,1,0},
    {0,0,0,1,0,0,0,0,0,1}};
B = {20,14,2,5,4,12,25,18};
c = {5,1,0,0,0,0,0,0,0,0};
d = {4,2,0,0,0,0,0,0,0,0};
α = 10;
β = 12;
basicfeasible[A,b]
qpoptimal[A, b, c]
```

Output for Numerical Example 2

```
The possible all basic solution is:
{{2,4,6,12,0,3,0,8,19,6},{16/5,4,0,6/5,6/5,
9/5,0,8,25,84/5},{2,10,0,6,0,3,6,2,25,12}}
The optimal value of the objective function
of the quasi-concave QP is   984
The optimal solution of the quasi-concave
QP is {16/5,4,0,6/5,6/5,9/5,0,8,25,84/5}
```

We observed that the result obtained by computer technique is completely identical with the result obtained by our proposed method for solving quasi-concave QPBV problems. In fact, it converges quickly. In our computer technique, 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. Also we observed that our computer oriented method can solve any quasi-concave QPBV problems.

VI. Conclusions

The aim of the research was to develop an easy technique for solving quasi-concave QPBV problems. So in this paper, we developed a new method for finding an optimal solution to a Quasi-Concave Quadratic Programming Problem with Bounded Variables in which the objective function involving the product of two indefinite factorized linear functions and constraints functions are in the form of linear inequalities. We illustrate some numerical examples to demonstrate our proposed method. We also developed a computer technique by using programming language *MATHEMATICA* for quasi-concave QPBV problems. We therefore, hope that our proposed method and computer technique can be used as an effective tool for solving quasi-concave QPBV problems and hence our time and labor can be saved.

References

1. Wolfe, P., 1959. The simplex method for quadratic programming, Econometrica, **27**, 3. 382-398.
2. Beale, E.M.L., 1959. On quadratic programming, Naval Research Logistics Quarterly, **6**, 227-243.
3. Frank, M., P. Wolfe, 1956. An algorithm for quadratic programming, Naval Research Logistics Quarterly, **3**, 95-110.
4. Shetty C.M., 1963. A simplified procedure for quadratic programming, Operations Research, **11**, 248-260.
5. Lemke C.E., 1965. Bi-matrix equilibrium points and mathematical programming, Management Science, **11**, 681-689.
6. Best M.J., K. Ritter K, 1988. A quadratic programming algorithm, Zeitschrift for Operational Research,**32**, 271-297.
7. Theil H., C. Van de Panne, 1961. Quadratic programming as an extension of conventional quadratic maximization, Management Science, **7**, 1-20.

8. J.C.G., Boot, 1961. Notes on quadratic programming: The kuhn-Tucker and Theil-van de Panne conditions, degeneracy and equality constraints, Management Science, **8**, 85-98.

9. Fletcher R., 1971. A general quadratic programming algorithm, J. Inst., Maths. Applics, **7**, 76-91.

10. Swarup K., 1966. Quadratic programming, CCERO (Belgium), **8**, 132-136.

11. Gupta A.K., J.K. Sharma, 1983. A generalized simplex technique for solving quadratic programming problem, Indian Journal of Technology, **21**, 198-201.

12. Moraru V., 1997. An algorithm for solving quadratic programming problems, Computer science Journal of Moldova, **5**, 223-235.

13. Moraru V., 2000. Primal-dual method for solving convex quadratic programming problems, Computer science Journal of Moldova, **8**, 209-220.

14. Jensen D.L., A.J. King, 1992. A decomposition method for quadratic programming, IBM Systems Journal, **31**, 39-48.

15. Bazaraa M., H. Sherali, C.M. Shetty, 2006. Nonlinear programming: Theory and algorithm (John Wiley, New York).

16. Whinston A., 1956. The Bounded Variable Problem- An Application of The Dual Method For Quadratic Programming, Naval Research Logistics Quarterly, **12(2)**, 173-179.

17. Hamadhy A. Taha, 2007. Operation Research: An Introduction, 8th edition, Pearson Prentice Hall, Pearson Education, Inc., Upper Saddle River, NJ 07458.

18. Hasan M.B., 2012. A Technique for solving Special Type Quadratic Programming Problems, The Dhaka University journal of science, **60(2)**, 209-215.

19. Gupta P.K., D.S. Hira, Problem in Operation Research, S. Chand & Company Ltd, Ram Nagar, New Delhi-110055.

20. Don, Eugene, 2001. Theory and Problems of Mathematica, Schaum's Outline Series, McGraw-Hill, New York San Francisco Washington, D.C.

21. Wolfram S., 2000. Mathematica, Addision-wesley Publication Company, Melno Park, California, New York.