A Method for Binary Quadratic Programming with Circulant Matrix

Daniela Istrati 1, ORCID: 0000-0002-1607-9273
Vasile Moraru 1, ORCID: 0000-0002-6400-1374
Sergiu Zaporojan 2, ORCID: 0000-0001-5928-4229

1 Department of Computer Science and Systems Engineering, Technical University of Moldova, Stefan cel Mare bd., 168, MD2004, Chișinău, Republic of Moldova, www.utm.md, daniela.istrati@ia.utm.md, vasile.moraru@ia.utm.md
2 Scientific Research Dept., Technical University of Moldova, 168, bvd. Stefan cel Mare, MD2004, Chișinău, Republic of Moldova, www.utm.md, sergiu.zaporojan@adm.utm.md

Abstract — Binary quadratic programming is a classical combinatorial optimization problem that has many real-world applications. This paper presents a method for solving the quadratic programming problem with circulant matrix by reformulating and relaxing it into a separable optimization problem. The proposed method determines local suboptimal solutions. To solve the relaxing problem, the DCA algorithm is proposed to calculate the solutions, in the general case, only local suboptimal.

Keywords— Binary nonconvex quadratic problems, circulant matrix, Fourier matrix, separable programming, relaxed problem, DC algorithm

I. INTRODUCTION

Consider the following linearly-constrained binary quadratic programming problem:

\[
\begin{aligned}
\min & \quad f(x) = x^T Q x \\
\text{subject to} & \quad Ax = b, \\
& \quad x \in \{0,1\}^n
\end{aligned}
\]  \hspace{1cm} (1)

where \( Q \) is a symmetric \( n \times n \) real matrix, \( A \) is an \( m \times n \) matrix, \( \text{rank}(A) = m \leq n \), and \( b \) is an \( m \) real vector.

We will briefly describe the notation used in this paper. All vectors are column vectors. The subscript notation \( y_k \) refers to an element of the vector \( y \). A superscript \( ^T \) is used to denote iteration numbers. Superscript \( "T" \) denotes transposition.

Over the years, various methods have been developed to solve the problem (1) by:
- linear reformulations [1], [2], [3];
- convex reformulations [4], [5];
- continuous convex programming [6];
- Lagrangian, semidefinite and convex quadratic relaxation, [7], [8], [9], [10].

In this paper we will consider that the \( Q \) matrix is a symmetric circulant matrix [11]:

\[
Q = \begin{pmatrix}
q_0 & q_1 & q_2 & \cdots & q_{n-2} & q_{n-1} \\
q_1 & q_2 & q_3 & \cdots & q_{n-1} & q_0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
q_{n-1} & q_0 & q_1 & \cdots & q_{n-3} & q_{n-2}
\end{pmatrix}
\]

Circulant matrices appear in a variety of mathematical and engineering applications such as signal processing and error correction of codes [12], [13].

In this context above, we present a method for solving the quadratic programming problem with circulant matrices \( Q \). The problem is converted into a separable programming problem, which consecutively is relaxed to a problem with the objective function represented as the difference of two convex functions, a problem called in the literature DC programming (DC-Difference of Convex functions).

II. EIGENVALUES AND EIGENVECTORS OF CIRCULANT MATRIX

The first row of the circulant matrix \( Q \)

\[
q_0 \quad q_1 \quad q_2 \cdots q_{n-2} \quad q_{n-1}
\]

is called the generator of \( Q \).

The eigenvalues of the symmetric matrix \( Q \) are real numbers and are given by
\( \lambda_j = q_0 + q_1\omega_j + q_2\omega_j^2 + \cdots + q_{n-1}\omega_j^{n-1} \)

\[ j = 1, 2, \ldots, n, \]

where

\[ \omega_j = \exp\left(\frac{2\pi(j-1)}{n}\right). \]

Note: for \( n \) even numbers \((n = 2k)\) we have \( \lambda_j = \lambda_{n-j}. \)

For \( j = 1, 2, \ldots, n, \) the corresponding eigenvectors are given by [11]:

\[ p_j = \left(\omega^0, \omega^{j-1}, \omega^{2(j-1)}, \ldots, \omega^{(j-1)(j-1)}\right)^T \quad (3) \]

Here \( \omega \) is the primitive root of unity:

\[ \omega = \exp\left(\frac{2\pi i}{n}\right), \quad i = \sqrt{-1}. \]

All circulant matrices can be diagonalized by the same matrix \( F \) with the columns \( p_j, j = 1, 2, \ldots, n \) [11]:

\[
F = \frac{1}{\sqrt{n}} \begin{pmatrix}
p_1 & p_2 & \cdots & p_n \\
1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\
1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)}
\end{pmatrix}
\]

The matrix \( F \) is the Fourier matrix (the Discrete Fourier Transform DFT) [6].

\( F \) is a matrix with the outstanding properties:

- \( F^T = F \)
- \( F^2 = I \)
- \( \det F = 1, \quad F^{-1} = F \)
- \( Sp(F) = \{-1, 1\} \)

Moreover, the matrix \( F \) is a well-conditioned matrix \((\text{cond}(F)=1)\). This is important from the point of view of numerical calculation: small perturbations in the input data will not produce large variations in the calculations [14].

The circulant matrices are diagonalized by the Fourier matrix \( F \), i.e. we can write

\[ Q = F\Lambda F \quad (4) \]

where \( \Lambda \) is the diagonal matrix:

\[
\Lambda = \text{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) =
\begin{pmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_n
\end{pmatrix}
\]

Thus the symmetric matrix \( Q \) is expressed in terms of matrices that contain its eigenvalues (2) and the components of the eigenvectors (3). Using the Fourier matrix \( F \), resulting from (4) and (5), the diagonalization of the circulant matrix can be performed \( Q = F\Lambda F = \Lambda \).

### III. Reformulation of the Quadratic Problem as a Separable Programming Problem

The objective function \( f(x) \) can be rewritten as:

\[ f(x) = x^T Q x = x^T F \Lambda F x = (Fx)^T \Lambda F x. \]

We note

\[ y = Fx = (y_1, y_2, \ldots, y_n)^T. \]

As the matrix \( F \) is orthogonal \((F^{-1} = F)\), we have

\[ x = F y. \]

Then problem (1) becomes a separable programming problem:

\[
\varphi(y) = y^T \Lambda y = \sum_{k=1}^{n} \lambda_k y_k^2 \rightarrow \max \\
\text{subject to} \\
AFy = b, \\
Fy \in \{0,1\}^n
\]

Among the eigenvalues of the \( Q \) matrix are both positive and negative numbers. The function \( \varphi(y) \) can be rewritten as the difference between two convex functions:

\[ \varphi(y) = \varphi_1(y) - \varphi_2(y) \]

where

\[ \varphi_1 = \sum_{\lambda_k > 0} \lambda_k y_k^2 \]

\[ \varphi_2 = \sum_{\lambda_k < 0} \lambda_k y_k^2 \]
\[ \varphi_2 = \sum_{\lambda_k < 0} (-\lambda_k) y_k^2 \]

The last constraints of problem (6) make it very difficult to solve it. If constraints are not taken into account \( Fy \in [0,1]^n \), then the problem (6) becomes a nonconvex separable quadratic programming problem. A practical approach would be to relax the conditions

\[ Fy \in [0,1]^n, \]

by replacing them with

\[ 0 \leq Fy \leq 1, \]

i.e. with

\[ 0 \leq p_j^T y_j \leq 1, j = 1,2,\ldots,n \]

Thus we obtain the relaxed problem

\[
\begin{aligned}
\varphi_1(y) - \varphi_2(y) & \rightarrow \text{max} \\
\text{subject to} & \\
AFy &= b, \\
0 & \leq Fy \leq 1 \\
\end{aligned}
\]

which is a DC programming problem [15].

IV. DC ALGORITHM

As it is mentioned above, to solve the relaxed problem (7) we will use the DCA method [15].

We denote the set of indices \( i_s \) for which the eigenvalues \( \lambda_{i_s} > 0 \):

\[ I = \{ i | \lambda_i > 0 \} = \{ i_1, i_2, \ldots, i_s \}. \]

The DCA method is of the primal-dual type and is based on the construction of two strings

\[ \{ y^{(k)} \}, \{ u^{(k)} \} \]

which are calculated at each iteration as follows:

Step 1. \( y^{(k)} \) - the initial state approximation, \( k = 0 \).

Step 2. It is determined

\[
\begin{align*}
\varphi_1(y^{(k)}) &= \frac{\partial \varphi_1(y^{(k)})}{\partial y_{i_1}} v, \\
\varphi_2(y^{(k)}) &= \frac{\partial \varphi_2(y^{(k)})}{\partial y_{i_2}} v, \\
& \vdots \\
\varphi_{i_s}(y^{(k)}) &= \frac{\partial \varphi_{i_s}(y^{(k)})}{\partial y_{i_s}} v,
\end{align*}
\]

Step 3. It is established \( y^{(k+1)} \) the solution of the convex separable programming problem:

\[
\begin{aligned}
\sum_{\lambda_k < 0} (-\lambda_k) y^2 - \sum_{\lambda_k > 0} \lambda_k u^{(k)} & \rightarrow \text{min} \\
\text{subject to} & \\
AFy &= b, \\
0 & \leq p_j^T y_j \leq 1, \\
& j = 1,2,\ldots,n.
\end{aligned}
\]

Step 4. If the stop criterion is checked, then STOP. Otherwise, \( k = k + 1 \) will be taken and it is proceed to Step 2.

V. CONCLUSIONS

In this paper, the 0-1 quadratic nonconvex programming problem with circulant matrices was considered. Such problems are NP-hard [16]. The diagonalization of the circulant matrix using the Fourier matrix allows reducing the considered problem to a separable programming problem.

To solve the relaxing problem, the DCA algorithm is proposed to calculate the solutions, in the general case, only local suboptimal. In order to find the optimal global solutions, other methods must be used, such as the branch and bound method [17].

These methods are slow and require many calculations that grow exponentially with the size of the problem. DC Numerical simulations show that in the case of non-convex quadratic programming problems, it is more advantageous to apply the DC Algorithm than the branch and bound method.

REFERENCES