https://doi.org/10.52326/ic-ecco.2022/CS.01

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 it 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:

$$\left.\begin{aligned} f(x) = x^T Qx \to max \\ \text{subject to} \\ Ax = b, \\ x \in \{0,1\}^n \end{aligned}\right\} \qquad (1)$$

where Q is a symmetric $n \times n$ real matrix, A is an $m \times n$ matrix, $rank(A) = m \leq n$, 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 k 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]:

$$Q = \begin{pmatrix} q_0 & q_1 & q_2 & \cdots & q_{n-2} & q_{n-1} \\ q_1 & q_2 & q_3 & \cdots & q_{n-1} & q_0 \\ q_2 & q_3 & q_n & \cdots & q_0 & q_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{n-1} & q_0 & q_1 & \cdots & q_{n-3} & q_{n-2} \end{pmatrix}.$$

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

$$\lambda_j = q_0 + q_1\omega_j + q_2\omega_j^2 + \cdots + q_{n-1}\omega_j^{n-1}$$
(2)

$j = 1,2,\cdots,n,$

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 $j = 1,2,\cdots,n$, the corresponding eigenvectors are given by [11]:

$$p_j = \left(\omega^0, \omega^{j-1}, \omega^{2(j-1)}, \cdots, \omega^{(j-1)(j-1)}\right)^T \qquad (3)$$

Here ω is the primitive root of unity :

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

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

$$F = \frac{1}{\sqrt{n}}(p_1 \quad p_2 \quad \cdots \quad p_n\ ) =$$

$$= \frac{1}{\sqrt{n}}\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{pmatrix}.$$

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

F is a matrix with the outstanding properties:

- $F^T = F$
- $F^2 = I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$
- $\det F = 1,$
- $F^{-1} = F$
- $Sp(F) = \{-1,1\}$

Moreover, the matrix F is a well-conditioned matrix ($cond(F)$=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

$$Q = F\Lambda F \qquad (4)$$

where Λ is the diagonal matrix:

$$\Lambda = Diag\ (\lambda_1, \lambda_2, \cdots, \lambda_n) =$$

$$= \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \lambda_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \lambda_n \end{pmatrix} \qquad (5)$$

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: $FQF = \Lambda$.

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

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

$$f(x) = x^TQx = x^TF\Lambda Fx = (Fx)^T\Lambda Fx.$$

We note

$$y = Fx = (y_1 \quad y_2 \quad \cdots \quad y_n)^T.$$

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

$$x = Fy.$$

Then problem (1) becomes a separable programming problem:

$$\left.\begin{array}{c} \varphi(y) = y^T\Lambda y = \sum_{k=1}^{n} \lambda_k y_k^2 \to max \\ \text{subject to} \\ AFy = b, \\ Fy \in \{0,1\}^n \end{array}\right\} \qquad (6)$$

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

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

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

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

$$0 \le Fy \le 1,$$

i.e. with

$$0 \le p_j^T y_j \le 1, j = 1,2,\cdots.n$$

Thus we obtain the relaxed problem

$$\left.\begin{array}{c} \varphi_1(y) - \varphi_2(y) \to max \\ \text{subject to} \\ AFy = b, \\ 0 \le Fy \le 1 \end{array}\right\} \qquad (7)$$

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 $\lambda_{i_s} > 0$:

$$I = \{i | \lambda_i > 0\} = \{i_1, i_2, \cdots, i_s\}.$$

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

$$\{y^{(k)}\}, \{v^{(k)}\}$$

which are calculated at each iteration as follows:

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

Step 2. It is determined

$$u^{(k)} = \nabla\varphi_1(y^{(k)}) = \begin{pmatrix} \dfrac{\partial\varphi_1(y^{(k)})}{\partial y_{i_1}} \\ \dfrac{\partial\varphi_1(y^{(k)})}{\partial y_{i_2}} \\ \vdots \\ \dfrac{\partial\varphi_1(y^{(k)})}{\partial y_{i_s}} \end{pmatrix}$$

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

$$\left.\begin{array}{c} \sum_{\lambda_k<0} (-\lambda_k) y^2 - \sum_{\lambda_k>0} \lambda_k u^{(k)} \to min \\ subject\ to \\ AFy = b, \\ 0 \le p_j^T y_j \le 1, \\ j = 1,2,\cdots.n. \end{array}\right\}$$

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

[1] W.P. Adams, R. Forrester, and F. Glover. „Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs". *Discrete Optimization*, 1(2):99–120, 2004.
[2] F. Glover and E. Woolsey. „ Further reduction of 0-1 polynomial programming problems to 0-1 linear programming problems" . *Operations Research*, 21:156–161, 1973

[3] H.D. Sherali and W.P. Adams. „A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems”. Kluwer Academic Publishers, Norwell, M., 1999.

[4] A. Billionnet, S. Elloumi, Marie-Christine Plateau. „Convex Quadratic Programming for Exact Solution of 0-1 Quadratic Programs”. Published 2005 Mathematics, Computer Science, 23 pp.

[5] M.W. Carter. „The indefinite zero-one quadratic problem”. *Discrete Applied Mathematics*, 7:23–44, 1984

[6] B. Kalantari and A. Bagchi. „An algorithm for quadratic zero-one programs”. *Naval Research Logistics*, 37:527–538, 1999

[7] A. Billionnet and E. Soutif. „An exact method based on lagrangian decomposition for the 0-1 quadratic knapsack problem”. *European J. of Operational Research*, 157(3):565–575, 2004

[8] M.X. Goemans. „Semidefinite programming in combinatorial optimization’. *Mathematical Programming*, pages 143–161, 1997

[9] C. Helmberg and F. Rendl. „Solving quadratic (0,1)-problems by semidefinite programs and cutting planes”. *Mathematical. Programming*, 8(3):291–315, 1998..

[10] P. Michelon and N. Maculan. „Lagrangian decomposition for integer non-linear programming with linear constraints”. *Mathematical Programming*, 52(2):303–314, 1991.

[11] Philip J. Davis Circulant matrices. Wiley, New York, 1979, xv + 250 pp. *‚2nd ed. Providence (RI): AMS Chelsea Publishing*; 1994

[12] A.V. Ramakrishna and T.V.N. Prasanna, „Symmetric circulant matrices and publickey cryptography”, *Int. J. Contemp. Math. Sciences* 8 (12) (2013), 589–593.

[13] Dhashna T. Pillai and Briji J. Chathely “A Study on Circulant Matrices and its Application in Solving Polynomial Equations and Data Smoothing*"*, *International Journal of Mathematics Trends and Technology* , Volume 66, Issue 6, (2020), 275-283.

[14] G Starng. „Linear algebra and its applications”, *4th Edition, Academic Press,* 2022, 487 pp.

[15] W. de Oliveria. “The ABC of DC Programming”. *Theory and Applications and Variational Analysis*, Volume 28, Issue 4, 2020, pp. 679-706.

[16] P.M. Pardalos, S.A. Vavasis. “Quadratic programming with one negative eigenvalue is NP-hard”. *Journal of Global Optimization*, Volume 1, 1991, pp.15-22

[17] D.R. Morrison,Sh. H. Jacobson, J.J. Sauppe and E. C. Sewell. „Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning”. *DiscreteOptimization*, Volume 19, 2016, pp. 79-102