Fifty-fourth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 27 - 30, 2016

New Methods for Handling Binary Constraints

Abhay Kumar Yadav, Rajeev Ranjan, Upal Mahbub, and Michael C. Rotkowitz

Abstract— **We consider the problem of handling binary constraints in optimization problems. We review methods for solving binary quadratic programs (BQPs), such as the spectral method and semidefinite programming relaxations. We then discuss two new methods for handling these constraints. The first involves the introduction of an extra unconstrained variable along with the alternating direction method of multipliers (ADMM), which has now also appeared in other recent literature. This allows the effect of the binary variables to be decoupled when they are minimized over. The second involves rewarding the one-norm while restricting the infinity-norm, based on a reformulation of the original problem. The piecewise linearity of the negative penalty results in the problem being convex until it hits a critical point, at which point the parameters of this linear term can be changed. These two methods can be applied to any problems which are convex except for binary constraints. In addition to testing them on BQPs, we show the efficacy of these approaches on point segmentation and image segmentation problems.**

I. Introduction

Clustering is a crucial technique to group similar data together and separate dissimilar data. It finds its use in a lot of scientific fields such as statistics, computer science, biology, etc. Given a set of points $x_1, x_2, ..., x_n$ and some measure of pairwise similarity between them, the goal of clustering is to partition the data points in such a way that points in the same group are similar and points in different groups are dissimilar to each other [1], [2].

The set of points can be represented by weighted graphs $G(\mathbf{V};\mathbf{E})$ in graph based segmentation, where the vertices $\mathbf{V}$ correspond to data points and edges $\mathbf{E}$ capture the pair-wise affinities w_{ij} between the points. Each point can be labeled either $+1$ or -1 based on which group it belongs to. A partition $\mathbf{x} \in \{-1, 1\}^n$ is optimized to cut the minimal edge weights and results into two balanced disjoint groups. The problem of finding the optimal partition can be formulated as a binary quadratic problem.

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{x}^T\mathbf{A}\mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \{-1,1\}^n, \end{aligned} \tag{1}$$

where $\mathbf{A} \in S_n^+$ is the normalizing matrix. This problem is non-convex and NP-hard because of the integrality of constraints.

One solution to the binary quadratic problem in (1) is given by the spectral method (SM) in which the constraint on x is relaxed to formulate the problem as:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{x}^T\mathbf{A}\mathbf{x} \\ \text{subject to} \quad & \|\mathbf{x}\|_2^2 = n, \end{aligned} \tag{2}$$

where $\mathbf{A} \in S_n^+$. While this problem is still non-convex, the solution can be found by eigen-decomposition of $\mathbf{A}$ in $O(n^3)$ time. The eigenvector corresponding to the algebraically smallest eigenvalue of $\mathbf{A}$ is the solution from the spectral relaxation method. However, the bound on the solution is loose and therefore the quality of the solution might be poor [3], [4] and it is difficult to generalize the spectral method to BQPs with linear or quadratic inequality constraints [5]. Also, the solution is highly dependent on threshold to classify a node to binary values $+1$ or -1. Attempts have been made to improve the spectral relaxation methods by normalizing the Laplacian matrix [6], [2]. Considering an additional linear inequality for (2) makes the problem NP-hard in general [7].

Another way to solve (1) is by relaxation to semi-definite programs which is defined as

$$\begin{aligned} \underset{\mathbf{X}\succeq 0}{\text{minimize}} \quad & < \mathbf{X}, \mathbf{A} > \\ \text{subject to} \quad & \text{diag}(\mathbf{X}) = \mathbf{1}; \\ & \text{rank}(\mathbf{X}) = 1, \end{aligned} \tag{3}$$

where $\mathbf{1}$ is a vector of 1's and $\mathbf{A} \in S_n^+$. In this formulation, the variable in (1) is lifted to the space of rank-one positive semi-definite matrices of the form $\mathbf{X} = \mathbf{x}\mathbf{x}^T$. By relaxing the rank constraint, the problem becomes convex and can be solved using semi-definite programming. This relaxation produces better results than the spectral method (2), but the number of variables increases to $n(n+1)/2$ thus increasing the execution time and memory. This method has poor scalability and therefore is impractical for large problems [8], [5]. The spectral and SDP relaxation methods can be regarded as two points on an axis of increasing relaxation performance.

We propose two different methods to address this problem. The first approach is based on ADMM, in which an auxiliary variable is introduced which is set equal to the binary variable, but with domain covering the whole space. The effect of the binary variables on the augmented Lagrangian is then decoupled, such that the step in which we minimize over them scales linearly, rather than exponentially, in the number of variables. The second one adds a negative L1 penalty to the objective function of (2) while constraining the ∞-norm. This arises as the minimization of a generalized Lagrangian for an equivalent formulation of the problem, and serves to push the variables towards either ± 1 while keeping the first term

A.K. Yadav, R. Ranjan, and U. Mahbub are with the Department of Electrical and Computer Engineering, The University of Maryland, College Park, MD 20742 USA, {ayadav,rranjan1,umahbub}@umd.edu.

M. C. Rotkowitz is with the Institute for Systems Research and the Department of Electrical and Computer Engineering, The University of Maryland, College Park, MD 20742 USA, mcrotk@umd.edu.

of the function as small as possible. The piecewise linearity of the penalty results in the problem being convex until the variable hits a critical point, at which point the parameters of the linear function can be adjusted.

The paper is organized as follows. Section II provides a brief background on Alternating Direction Method of Multipliers (ADMM). Section III describes the ADMM-BQP and L1-BQP methods in detail. Section IV provides the experiments and results of the proposed methods. Finally, Section V concludes the paper with a brief summary and discussion.

II. Background

A. Alternating Direction Method of Multipliers (ADMM)

In this section, we briefly describe general Alternating Direction Method of Multipliers (ADMM) approach for solving optimization problems. The ADMM is a powerful optimization technique that combines the benefits of dual decomposition and method of multipliers. The basic idea revolves around decomposing a large (and probably difficult) problem into a set of (simpler to solve) subproblems and cleverly combining their solutions in a principled manner to recover the solution of the original problem. To describe ADMM, consider an optimization over convex functions f and g defined as

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n;\mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \end{aligned} \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{p\times n}$, $\mathbf{B} \in \mathbb{R}^{p\times m}$ and $\mathbf{c} \in \mathbb{R}^p$.

Notice that the variables x and z are coupled through the linear constraint while the objective function is separable. This problem can be turned into an unconstrained minimization problem by introducing the augmented Lagrangian

$$\begin{aligned} L_t(\mathbf{x},\mathbf{z},\boldsymbol{\lambda}) = f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) \\ +(t/2)\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2, \end{aligned} \tag{5}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^p$ is the Lagrange multiplier associated with the constraint $t > 0$ is a fixed parameter.

ADMM performs the following procedure at k^{th} iteration

$$\begin{aligned} \mathbf{x}^{k+1} &= \underset{\mathbf{x}\in\mathbb{R}^n}{\text{argmin}}\ L_t(\mathbf{x},\mathbf{z}^k,\boldsymbol{\lambda}^k) \\ \mathbf{z}^{k+1} &= \underset{\mathbf{z}\in\mathbb{R}^m}{\text{argmin}}\ L_t(\mathbf{x}^{k+1},\mathbf{z},\boldsymbol{\lambda}^k) \\ \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + t(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}). \end{aligned} \tag{6}$$

Using the scaled dual variable ($\boldsymbol{\mu} = \frac{\boldsymbol{\lambda}}{t}$), Lagrangian for ADMM can be expressed as

$$L_t(\mathbf{x},\mathbf{z},\boldsymbol{\mu}) = f(\mathbf{x}) + g(\mathbf{z}) + (t/2)\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} + \boldsymbol{\mu}\|_2^2, \tag{7}$$

and also, the update equations for the variables are

$$\begin{aligned} \mathbf{x}^{k+1} &= \underset{\mathbf{x}\in\mathbb{R}^n}{\text{argmin}}\ L_t(\mathbf{x},\mathbf{z}^k,\boldsymbol{\mu}^k) \\ \mathbf{z}^{k+1} &= \underset{\mathbf{z}\in\mathbb{R}^m}{\text{argmin}}\ L_t(\mathbf{x}^{k+1},\mathbf{z},\boldsymbol{\mu}^k) \\ \boldsymbol{\mu}^{k+1} &= \boldsymbol{\mu}^k + (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}). \end{aligned} \tag{8}$$

ADMM has also been used for non-convex problems [9], [10], [11], and bi-convex problems in particular. For some non-convex problems, it can be shown that the algorithm converges to local optima [12].

III. Proposed Methods

We propose two alternate methods for solving the BQPs. The methods are discussed in the following two subsections.

A. ADMM-BQP

1) Formulation: We now discuss a method in which we introduce an auxiliary variable, which is constrained to be equal to the binary variable, but is not subject to that restriction. This is used with ADMM, which will result in the effect of the binary variables being decoupled when we consider their minimization. This was motivated largely by the success of ADMM in dealing with the binary constraints in [13], though the problem addressed in that paper did not require the introduction of a new variable which was explicitly designed to serve as a free version of the binary variable, as we do here. It has now also appeared in [14], where it was shown that this technique can be used for any constraints that result in feasible sets which are non-convex but easy to project onto.

$$\begin{aligned} \underset{\mathbf{x}\in\{-1,1\}^n;\mathbf{y}\in\mathbb{R}^n}{\text{minimize}} \quad & \frac{1}{2}\mathbf{y}^T\mathbf{A}\mathbf{y} \\ \text{subject to} \quad & \mathbf{x} = \mathbf{y}, \end{aligned} \tag{9}$$

where $\mathbf{A} \in S_n^+$ and the augmented Lagrangian function can be formed as

$$L(\mathbf{x},\mathbf{y},\boldsymbol{\mu}) = \frac{1}{2}\mathbf{y}^T\mathbf{A}\mathbf{y} + \frac{t}{2}\|\mathbf{x} - \mathbf{y} + \boldsymbol{\mu}\|_2^2, \tag{10}$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ is the scaled dual variable. The augmented Lagrangian function (10) can be minimized over $\mathbf{x}$ and $\mathbf{y}$ iteratively by fixing one variable at a time and updating the other. The entire algorithm is summarized in Algorithm 1.

The key observation of the Algorithm 1 is that while minimizing over $\mathbf{x}$, all the components x_i of $\mathbf{x}$ are decoupled and thus, can be optimized independently. Since each x_i can be either $+1$ or -1, we have to check only 2 values for each x_i; in fact, the solution can be taken as simply $\text{sign}(y^k - \mu^k)$. Intuitively, it can be thought of thresholding the real valued solution ($\mathbf{y}$) obtained after each ADMM iteration, with an adjustment for the dual variable or constraint mismatch.

2) Stopping Criteria: Following [15], a reasonable termination criterion is that the primal and dual residuals must be small, i.e.,

$$\|\mathbf{r}^k\|_2 \le \epsilon_{\text{pri}} \quad \text{and} \quad \|\mathbf{s}^k\|_2 \le \epsilon_{\text{dual}}, \tag{11}$$

where $\epsilon_{\text{pri}} = \sqrt{n}(\epsilon^{\text{abs}} + \epsilon^{\text{rel}})$ and $\epsilon_{\text{dual}} = \sqrt{n}(\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\|\boldsymbol{\mu}^k\|_2)$ are tolerances for the primal and dual, respectively. $\epsilon^{\text{abs}} > 0$ is an absolute tolerance and $\epsilon^{\text{rel}} > 0$ is a relative tolerance.

The primal and dual residuals $\mathbf{r}^k$ and $\mathbf{s}^k$ are given by

$$\begin{aligned} \mathbf{r}^k &= \mathbf{x}^k - \mathbf{y}^k \\ \mathbf{s}^k &= t(\mathbf{y}^k - \mathbf{y}^{k-1}). \end{aligned} \tag{12}$$

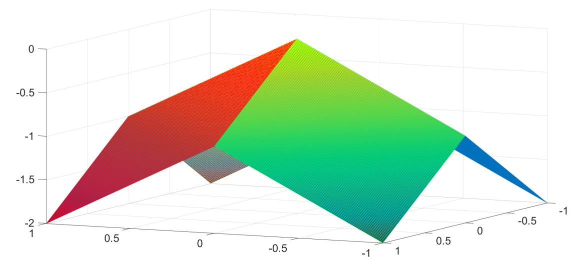

Fig. 1. Add negative L1 penalty to the objective.

Algorithm 1 Solving ADMM-BQP

```
Input  : Step size $t > 0$, Laplacian Matrix $\mathbf{A} \in S_n^+$
Output : The assignment vector $\mathbf{x} \in \{-1, 1\}^n$

Initialize $k = 0, \mathbf{y}^0 = \mathbf{0} \in \mathbb{R}^n, \boldsymbol{\mu}^0 = \mathbf{0} \in \mathbb{R}^n$
while not converged do
    Fix y and update x by:
    $\mathbf{x}^{k+1} = \underset{\mathbf{x} \in \{-1,1\}^n}{\operatorname{argmin}} \; L(\mathbf{x}, \mathbf{y}^k, \boldsymbol{\mu}^k)$   OR
    $x_i^{k+1} = \underset{x_i \in \{-1,1\}}{\operatorname{argmin}} \; (x_i - y_i^k + \mu_i^k)^2, \quad \forall i = 1, \ldots, n$
    Fix x and update y by:
    $\mathbf{y}^{k+1} = \underset{\mathbf{y} \in \mathbb{R}^n}{\operatorname{argmin}} \; L(\mathbf{x}^{k+1}, \mathbf{y}, \boldsymbol{\mu}^k)$   OR
    $\mathbf{y}^{k+1} = t(\mathbf{A} + t\mathbf{I})^{\dagger}(\mathbf{x}^{k+1} + \boldsymbol{\mu}^k)$

    Update dual variable:
    $\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}^k + \mathbf{x}^{k+1} - \mathbf{y}^{k+1}$

    Increase Iteration:
    $k = k + 1$
end
```

B. L1-norm regularized BQP (L1-BQP)

1) Formulation: The original binary problem can be reformulated as shown in (13). The L-infinity constraint ensures that the solution is bounded between -1 and +1, whereas the L1 constraint makes sure to push the solution out until it hits a corner of the hypercube and is binary. The method we introduce here can be arrived at by considering a perturbation function of this problem which only perturbs the last constraint, and then minimizes the resulting Lagrangian function.

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & \mathbf{x}^T \mathbf{A} \mathbf{x} \\ \text{subject to} \quad & \|\mathbf{x}\|_\infty \leq 1; \\ & \|\mathbf{x}\|_1 \geq n \end{aligned} \tag{13}$$

This results in adding a negative one-norm penalty to the objective function, while keeping the constraint on the infinity-norm, as shown in (14). Fig. 1 depicts the negative one-norm function for a two dimensional variable.

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & \mathbf{x}^T \mathbf{A} \mathbf{x} - \gamma \|\mathbf{x}\|_1 \\ \text{subject to} \quad & -\mathbf{1} \leq \mathbf{x} \leq \mathbf{1}; \end{aligned} \tag{14}$$

This problem is non-convex of course, but now the only term driving this is one which is piecewise linear. The problem is thus convex on regions in which the variable does not hit a critical point, and we now explore a few methods which exploit this to address this problem.

First, an additional constraint is added to restrict the solution $\mathbf{x}^k$ within a certain range of the previous solution $\mathbf{x}^{k-1}$. The proposed problem is shown in (15).

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & \mathbf{x}^T \mathbf{A} \mathbf{x} - \gamma \|\mathbf{x}\|_1 \\ \text{subject to} \quad & -\mathbf{1} \leq \mathbf{x} \leq \mathbf{1}; \\ & \|\mathbf{x} - \mathbf{x}^{k-1}\|_2 \leq \delta, \end{aligned} \tag{15}$$

where δ is a parameter. However, for varying the value of each component of $\mathbf{x}$ from -1 to $+1$ a total of 2^n convex problems are created, where n is the length of the $\mathbf{x}$ vector. For example, when $\mathbf{x}$ is a 2-dimensional vector, say, $\mathbf{x} = [x_1 \; x_2]^T$, there are 4-quadrants around the origin for varying both x_1 and x_2 from -1 to $+1$. Hence, in order to get the optimum solution for this problem directly, the problem needs to be reformulated in a case-by-case basis, i.e. 4 different problems need to be solved separately and the one that gives the minimum optimal value should be taken as the final solution. However, the number of such independent convex problems grow exponentially with the length of $\mathbf{x}$, making it an NP-hard problem. Also, from implementation point of view, the objective function is the difference of two convex functions and hence non-convex, therefore, the formulation in (15) is not suitable for straight-forward implementation.

In order to obtain a feasible solution of the L1-BQP problem, we utilize the sign information of the solution at previous instance to modify the problem into a quadratic convex problem. For a positive semi-definite A matrix, the overall L1-BQP algorithm for the k-th iteration has the form

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & \mathbf{x}^T \mathbf{A} \mathbf{x} - \gamma \mathbf{z}^T \mathbf{x} \\ \text{subject to} \quad & -1 \leq \mathbf{x} \leq \mathbf{1}; \\ & \|\mathbf{x} - \mathbf{x}^{k-1}\|_2 \leq \delta, \end{aligned} \tag{16}$$

where $z = \mathbf{sign}(\mathbf{x^{k-1}})$ is a constant at the k-th iteration, γ is the L1-norm multiplying factor and δ is the proximity constraint factor. Therefore, the second part of the objective function is basically a linear function of $\mathbf{x}$ making the overall quadratic problem convex. Here, $\mathbf{x}$ can take any real value between -1 and $+1$ and after obtaining the solution we perform functional thresholding to find the optimum threshold that minimizes the objective function while the components of $\mathbf{x}$ are pushed towards either -1 or $+1$.

2) Stopping Criteria: We terminate the algorithm when the optimal solution for the current iteration is less than some ϵ distance from the solution of the previous iteration, i.e.,

$$\|\mathbf{x} - \mathbf{x}^{k-1}\|_2 \leq \epsilon, \tag{17}$$

where ϵ was chosen to be $\sqrt{n} \times 10^{-3}$. The algorithm was also applied for image segmentation. The results are shown in Fig. 4. The complete algorithm for this method is shown in Algorithm 2.

Algorithm 2 Solving L1-BQP

```
Input  : L1-norm multiplying factor $\gamma^0$, Proximity constraint
         factor $\delta$, Laplacian Matrix $\mathbf{A} \in S_n^+$
Output : The assignment vector $\mathbf{x} \in \{-1, 1\}^n$

Initialize $k = 0, \mathbf{x}^0 \in \mathbb{R}^n, -\mathbf{1} \leq \mathbf{x}^0 \leq \mathbf{1}$
while not converged do
    Compute sign of $\mathbf{x}^k$ and update $\mathbf{z}$ by:
    $\mathbf{z} = \mathbf{sign}(\mathbf{x^k})$

    Solve the convex problem in (16) to obtain optimal $\mathbf{x}^*$

    Update primal variable:
    $\mathbf{x}^{k+1} = \mathbf{x}^*$
    Update L1-norm multiplying factor: $\gamma^{k+1} = 1.5 \times \gamma^k$

    Increase Iteration:
    $k = k + 1$
end
```

3) Alternative Approach: We also investigated an alternative L1-BQP approach which enforces the L1-norm function more strictly in the objective function. The alternative approach has the following differences:

- a constraint is imposed on the $\mathbf{x}^k$ variable which restraints its components to change their signs during the k-th iteration. The components that try to change signs can only go very close to zero during that iteration.
- before going into the next iteration, the signs of all those components of $\mathbf{x}$ which are within ϵ distance from 0 are toggled.

The intuition behind imposing these constraints stems from the fact that in the formulation of L1-BQP, the value of $\mathbf{z}$ is set to the sign information of the $\mathbf{x}$ values of previous iteration in order to replace the absolute value operator for calculating the L1-norm. This way the objective function is made convex. However, if $\mathbf{x}$ is permitted to change signs during an iteration then $\mathbf{z}^T\mathbf{x}$ will not be the L1-norm. Thus, this alternative approach is actually more in-line with the ideal formulation presented in (15). Formally, the alternate version of L1-BQP for the k-th iteration has the form (here $\mathbf{x} \equiv \mathbf{x}^k$ and $z \equiv z(k)$)

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & \mathbf{x}^T\mathbf{A}\mathbf{x} - \gamma\mathbf{z}^T\mathbf{x} \\ \text{subject to} \quad & -\mathbf{1} \leq \mathbf{x} \leq \mathbf{1}; \\ & \|\mathbf{x} - \mathbf{x}^{k-1}\|_2 \leq \delta; \\ & z_i x_i > 0 \quad \forall i = 1, 2, ..., n, \end{aligned} \tag{18}$$

where, $\mathbf{z}^0 = \text{sgn}(\mathbf{x}^0)$, $\mathbf{x}^k$ is the solution to the problem and

$$z_i^{k+1} = \begin{cases} -z_i^k & \text{if } x_i^k = 0; \\ z_i^k & \text{otherwise.} \end{cases} \tag{19}$$

After solving for optimal $\mathbf{x}^k$, we set $\mathbf{x}^k(|\mathbf{x}| \leq \epsilon) = -\mathbf{x}^k(|\mathbf{x}| \leq \epsilon)$ and then the new z for next iteration, $z(k+1) = \mathbf{sign}(\mathbf{x^k})$. We have chosen $\epsilon = 10^{-5}$ empirically while evaluating the method. Here, the additional constraint $z_i x_i > 0 \quad \forall i = 1, 2, ..., n$ denotes that the element-wise product between $\mathbf{z}$ and $\mathbf{x}$ must always be positive, which is possible only if the signs of z_i and x_i are same and therefore this condition imposes that the sign of the variables remain unchanged while solving the convex problem during the iteration. The complete algorithm for this method is shown in Algorithm 3.

Algorithm 3 Alternative L1-BQP Approach

```
Input  : L1-norm multiplying factor $\gamma$, Proximity constraint
         factor $\delta$, Zero Proximity Constraint Factor $\epsilon$, Lapla-
         cian Matrix $\mathbf{A} \in S_n^+$
Output : The assignment vector $\mathbf{x} \in \{-1, 1\}^n$

Initialize $k = 0, \mathbf{x}^0 \in \mathbb{R}^n, -\mathbf{1} \leq \mathbf{x}^0 \leq \mathbf{1}$
while not converged do
    Compute sign of $\mathbf{x}^k$ and update $\mathbf{z}$ by:
    $\mathbf{z} = \mathbf{sign}(\mathbf{x^k})$

    Solve the convex problem in (19) to obtain optimal $\mathbf{x}^*$

    Update primal variable:
    $\mathbf{x}^k(|\mathbf{x}| \leq \epsilon) = -\mathbf{x}^k(|\mathbf{x}| \leq \epsilon)$
    $\mathbf{x}^{k+1} = \mathbf{x}^*$

    Update L1-norm multiplying factor: $\gamma^{k+1} = 1.5 \times \gamma^k$

    Increase Iteration:
    $k = k + 1$
end
```

One major difference in performance between L1-BQP and its alternative is that the alternative needs more iterations to converge to the solution. This is obvious since in the alternative approach the variables are not allowed to change sign within an iteration. Whereas, in L1-BQP they can easily switch between the negative and positive values to minimize the objective function within the same iteration and therefore, can move faster to the optimum. Apparently, both methods work well and produce almost similar results on the datasets that are chosen for evaluating the performance, however, since the L1-BQP approach is faster, we will only discuss the results obtained for L1-BQP in the experimental results section.

IV. Experiments

The experiments were performed on the Atom dataset, Chainlink dataset [16] and on an artificial dataset. The Atom dataset is 3-dimensional with 800 total number of points, equally separated into two groups. The Chainlink dataset is also 3-dimensional with 1000 data points. A total of 400 points were generated for artificial dataset from two 2-dimensional gaussian distribution with different mean and standard deviation.

In Table I, comparisons on the synthetically generated dataset is in terms of mean objective value taken over 100

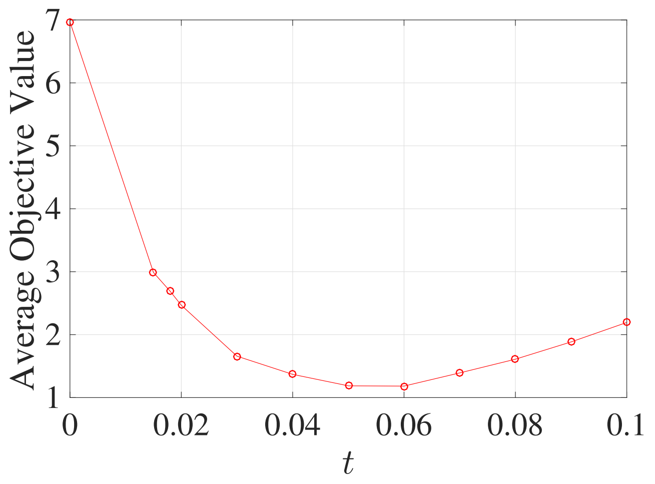

Fig. 2. Average Objective Value vs. t for ADMM-BQP

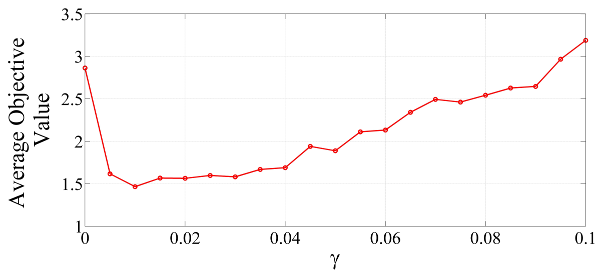

Fig. 3. Average Objective value vs. γ ($\delta = 7$) for L1-BQP

matrices. We analyzed two different methods for initialization: a) Random Initialization where all methods are initialized with same random value, and b) Spectral Initialization where all methods are initialized with the solution obtained from the spectral method. Here, lower numbers represent better performance and as can be seen from the table, both ADMM-BQP and L1-BQP methods perform reasonably compared to spectral method. The L1-BQP approach performs even better if γ is varied. In a similar manner, another set of performance comparison is shown in II in terms of mean and standard deviation of the objective value taken over 10 random initialization on the synthetically generated dataset. The advantage of varying γ is evident in this table as well.

The effect of varying the parameter t vs average objective value is shown in Figure 2 for ADMM-BQP approach. If t is too low then ADMM-BQP approach gives very low weightage to the penalty term and hence the solution x becomes all 1's or -1's leading to the maximum possible objective value. On the other hand, if we keep on increasing t then initially the objective value decreases and reaches to an optimum value and after that again objective increases as t increases which might be due to the fact that it gets stuck in local minima.

The performance of L1-BQP approach at various γ on the artificial dataset is shown in Figure 3. The mean objective value decreases upto $\gamma = 0.01$, and then increases steadily. When γ is very small, the quadratic term dominates resulting in a solution which is close to 0. On the other hand, when γ is large, it will push the initial solution to its nearest binary value culminating to high error. We choose the fixed γ of

Method	Mean Objective (Rand. init.)	Mean Objective (Spectral init.)
Spectral Method	2.6184	NA
ADMM-BQP	1.1827	1.0806
L1-BQP	1.6845	1.6544
L1-BQP (γ Varying)	1.3065	1.2913

TABLE I
PERFORMANCE COMPARISON ON THE SYNTHETICALLY GENERATED DATASET IN TERMS OF MEAN OBJECTIVE VALUE TAKEN OVER 100 MATRICES

Method	Mean Objective	Std. Deviation
Spectral Method	2.2383	NA
ADMM-BQP	1.1057	0.1001
L1-BQP	1.7369	0.4991
L1-BQP (γ Varying)	1.2547	0.1335

TABLE II
PERFORMANCE COMPARISON ON THE SYNTHETICALLY GENERATED DATASET IN TERMS OF MEAN AND STANDARD DEVIATION OF THE OBJECTIVE VALUE TAKEN OVER 10 RANDOM INITIALIZATION.

0.01 to compare with the varying γ approach . Also, the proximity constraint factor δ was chosen empirically to be $0.2\sqrt{n}$ which provides an average movement window of 0.2 for each component of $\mathbf{x}$. The solution obtained didn't require explicit thresholding because the solution values were either $+1$ or -1. There was also some dependency on the initialization of $\mathbf{x}$. Correct solution was obtained most of the times, but for few initializations the all the points were grouped to one cluster.

The experiments were also performed for image segmentation. Images are represented by weighted graphs $G(\mathbf{V};\mathbf{E})$ in graph based segmentation, where the vertices $\mathbf{V}$ correspond to pixels and edges $\mathbf{E}$ capture the pair-wise similarities between pixel. Each pixel can be labeled as $+1$ or -1 based on whether it is a part of foreground or background. A partition $\mathbf{x} \in \{-1,1\}^n$ is optimized to cut the minimal edge weights and results into two balanced disjoint groups. Performance can be improved by encoding labeled vertices of a graph, i.e., pixels/superpixels in an image [5]. The problem of image segmentation can also be formulated as a binary quadratic problem given by (1), where the graph Laplacian matrix $\mathbf{A}$ for images is computed as

$$\mathbf{A} = \mathbf{D} - \mathbf{W}. \tag{20}$$

Here, the elements of affinity matrix $\mathbf{W}$ is given by w_{ij}. $\mathbf{D}$ is a diagonal matrix where the i^{th} diagonal element is given by

$$d_{ii} = \sum_{j=1}^{n} w_{ij}. \tag{21}$$

The affinity matrix or the similarity graph is usually computed using one of the three ways: ϵ-neighborhood graph, k-nearest neighbors graph or fully-connected graph [1]. The

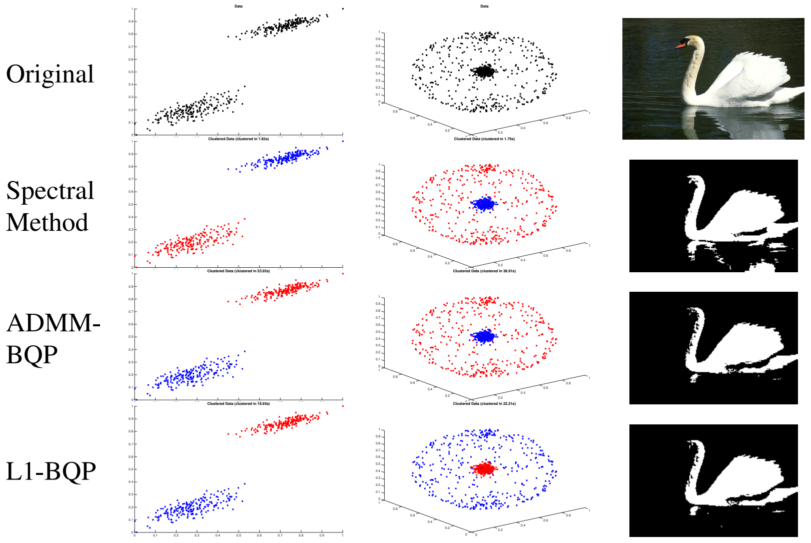

Fig. 4. Clustering and segmentation results (from left to right) on (a) artificial dataset. (b) Atom dataset. (c) Chainlink dataset. (d) RainbowDash, and (e) WhiteSwan

first one is an unweighted graph with the connection between nodes only if they are in an ϵ distance from one another. The second method connects vertices v_i to v_j if the latter is one of the k-nearest neighbors of the former. The third alternative is to connect all points in the graph with edge weights computed using Gaussian similarity function given by

$$w_{ij} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2}\right), \tag{22}$$

where the parameter σ controls the width of the neighborhood. For the image segmentation problem, the affinity matrix $\mathbf{W}$ is calculated as

$$w_{ij} = \begin{cases} \exp\left(-\frac{||\mathbf{f}_i - \mathbf{f}_j||_2^2}{\sigma_f^2} - \frac{\text{dist}(i,j)^2}{\sigma_d^2}\right), & \text{if dist}(i,j) < r \\ 0 & \text{otherwise} \end{cases}. \tag{23}$$

Here, $\mathbf{f}_i$ and $\mathbf{f}_j$ are color histograms of i, j and $\text{dist}(i,j)$ is the spatial distance between i,j. The values of σ_f, σ_d and r are given by the user. We perform the evaluation on two images: "RainbowDash" and "WhiteSwan".

The performance of the ADMM-BQP approach at various increasing threshold t is evaluated and the $\%$ Error (with respect to the ground truth GT) is plotted in Fig. 2. We can see that the error increases with higher values of t. Intuitively, it seems as t increases, the method gives less importance to the objective (clustering) as compared to the constraint ($\mathbf{x} = \mathbf{y}$) and thus resulting in poor clustering.

As the problem is non-convex, ADMM-BQP approach can stuck in local optima, thus we run the method with several random initialization and choose the best one [12]. We also verified that the initial value of $\boldsymbol{\mu}$ need not be exactly $\mathbf{0}$ but should be kept small (For example: $\mathbf{10^{-3}}$) for better convergence.

The segmentation result for different methods are shown in Figure 4. The results for SDP method are not presented here due to the fact that the SDP method went out-of-memory for large number of variables. We, however, tested it's performance for very small number of variables and seen that it is somewhat similar to that of the SM approach. It can be seen from the figure that the segmentation performance of all the three methods are similar on the first three point segmentation cases, namely, the artificial data, the Atom data and the Chainlink data. All three methods are capable of perfectly clustering the points into two segments. However, for the RainbowDash image the proposed approaches successfully segment the foreground from the background, whereas, the SM method fails to differentiate between them. On the other hand, for the WhiteSwan image, all three methods were able to segment the foreground object from the background. It is evident from these results that the two proposed methods demonstrate comparable or even superior segmentation performance over traditional BQP approaches.

V. Conclusion

We reviewed several methods for handling binary constraints, and binary quadratic problems in particular. We then discussed a method utilizing ADMM, along with the introduction of an additional variable which is identical to the binary variable but unconstrained, which is similar to some methods introduced in recent literature. We further introduced a method based on restricting the infinity-norm while penalizing the one-norm. This can be viewed as the minimization of a Lagrangian for an equivalent formulation of the binary problem, which will likely inform more systematic parameter updates and analysis. These last two methods are not restricted to BQPs, but can be applied to any problem which is convex except for binary constraints.

References

[1] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007. [Online]. Available: http://dx.doi.org/10.1007/s11222-007-9033-z

[2] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[3] S. Guattery and G. L. Miller, “On the quality of spectral separators,” *SIAM J. Matrix Anal. Appl.*, vol. 19, no. 3, pp. 701–719, Jul. 1998. [Online]. Available: http://dx.doi.org/10.1137/S0895479896312262

[4] K. Lang, “Fixing two weaknesses of the spectral method,” in *Advances in Neural Information Processing Systems*, 2005, pp. 715–722.

[5] P. Wang, C. Shen, and A. van den Hengel, “A fast semidefinite approach to solving binary quadratic problems,” *CoRR*, vol. abs/1304.0840, 2013. [Online]. Available: http://arxiv.org/abs/1304.0840

[6] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1109/34.868688

[7] T. Cour and J. Shi, “Solving Markov random fields with spectral relaxation,” in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, vol. 2, 2007, pp. 75–82.

[8] C. Olsson, A. P. Eriksson, and F. Kahl, “Improved spectral relaxation methods for binary quadratic optimization problems,” *Comput. Vis. Image Underst.*, vol. 112, no. 1, pp. 3–13, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2008.05.010

[9] S. You and Q. Peng, “A non-convex alternating direction method of multipliers heuristic for optimal power flow,” in *Proceedings of the IEEE International Conference on Smart Grid Communications*, 2014, pp. 788–793.

[10] X. Bai, K. Scheinberg, and R. Tutuncu, “Least-squares approach to risk parity in portfolio selection,” *Quantitative Finance*, vol. 16, no. 3, pp. 357–376, 2016.

[11] R. Chartrand and B. Wohlberg, “A nonconvex ADMM algorithm for group sparsity with sparse groups,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6009–6013.

[12] S. Magnússon, P. C. Weeraddana, M. G. Rabbat, and C. Fischione, “On the convergence of alternating direction Lagrangian methods for nonconvex structured optimization problems,” *arXiv preprint arXiv:1409.8033*, 2014.

[13] A. Alavian and M. Rotkowitz, “An optimization-based approach to decentralized assignability,” in *Proceedings of the 2016 American Control Conference*, July 2016, pp. 5199–5204.

[14] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, “A simple effective heuristic for embedded mixed-integer quadratic programming,” in *Proceedings of the 2016 American Control Conference*, July 2016, pp. 5620–5625.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[16] F. Moutarde and A. Ultsch, “U*F clustering: A new performant ”cluster-mining” method based on segmentation of self-organizing maps,” in *Proceedings of the 5th Workshop on Self-Organzing Maps*, 2005.